LogRhythm Integrated Solution 6.3.4 Security Target

Version 1.2
18 December 2015

Prepared for:

LogRhythm Inc.
4780 Pearl East Circle
Boulder, CO 80301

Prepared by:

Leidos Inc. (formerly Science Applications International Corporation)

Common Criteria Testing Laboratory
6841 Benjamin Franklin Drive, Columbia, Maryland 21046

LIST OF TABLES

1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is LogRhythm Integrated Solution 6.3.4 provided by LogRhythm Inc. The product is a network security appliance solution that provides log and event management, file integrity monitoring, and endpoint monitoring and control. The TOE consists of several components that coordinate with one another to collect and analyze information from multiple log sources including syslog, snmp, netflow and sflow devices, Windows events, flat file, databases or applications.

The focus of this evaluation is on the TOE functionality supporting the claims in the *Protection Profile for Network Devices* (See section 1.2 for specific version information) only, all other capabilities are not covered. The security functionality specified in [NDPP] includes protection of communications between TOE components and trusted IT entities, identification and authentication of administrators, auditing of security-relevant events, ability to verify the source and integrity of updates to the TOE, and specifies FIPS-validated cryptographic mechanisms.

The Security Target contains the following additional sections:

- TOE Description (Section 2)
- Security Problem Definition (Section 3)
- Security Objectives (Section 4)
- IT Security Requirements (Section 5)
- TOE Summary Specification (Section 6)
- Protection Profile Claims (Section 7)
- Rationale (Section 8)

1.1 Security Target, TOE and CC Identification

ST Title – LogRhythm Integrated Solution 6.3.4 Security Target

ST Version – Version 1.2

ST Date – 18 December 2015

TOE Identification –

The TOE consists of a deployment of the following LogRhythm appliances and applications which constitute the LogRhythm Integrated Solution 6.3.4. All appliances are pre-configured for Windows Server 2008 R2.

- 1 Event Manager (EM) with LogRhythm Event Manager Software v6.3.4
  - LR-EM3350 (dedicated EM server appliance)
  - LR-EM5350 (dedicated EM server appliance)
  - LR-EM7350 (dedicated EM server appliance)
- 1 or more Log Manager(s) (LM) with LogRhythm Log Manager Software v6.3.4
  - LR-LM3300 Series (dedicated LM server appliance)
  - LR-LM5300 Series (dedicated LM server appliance)
  - LR-LM7300 Series (dedicated LM server appliance)

OR

- 1 or more All-In-One LogRhythm XM appliances with LogRhythm Event Manager Software v6.3.4 and LogRhythm Log Manager Software v6.3.4
  - LR-XM4310 (combined EM/LM server appliance)

- LR-XM4350 (combined EM/LM server appliance)
- LR-XM6310 (combined EM/LM server appliance)
- LR-XM6350 (combined EM/LM server appliance)

Note: One or more All-In-One LogRhythm XM appliances with LogRhythm Event Manager Software v6.3.4 and LogRhythm Log Manager Software v6.3.4 can be used in a LogRhythm deployment instead of individual instances of the Event Manager and Log Managers.

AND

- 1 or more Advanced Intelligence Engine (AI Engine) Server(s) with LogRhythm Advanced Intelligence Engine Software v6.3.4
  - LR-AIE5310 (dedicated AI Engine appliance)
  - LR-AIE7310 (dedicated AI Engine appliance)
  - LR-AIE9310 (dedicated AI Engine appliance)
- 1 or more Site Log Forwarder (SLF) with LogRhythm Site Log Forwarder Software v6.3.4
  - LR-SLF3310 (Site Log Forwarder Appliance)
- 1 or more LogRhythm Web Services Appliances with LogRhythm Web Services Software v6.3.4
  - WS-3310 (dedicated Web Services appliance)

Applications

- 1 or more LogRhythm Client Console Software v6.3.4 (Windows application that runs on one of the appliances (either the EM or XM) in the evaluated configuration.)

TOE Developer – LogRhythm Inc.

Evaluation Sponsor – LogRhythm Inc.

CC Identification – *Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 4, September 2012*

1.2 Conformance Claims

This TOE is conformant to the following CC specifications:

- This ST is conformant to the *Protection Profile for Network Devices*, Version 1.1, 8 June 2012 (NDPP) as amended by Errata #3 dated 3 November 2014, and includes the additional optional SFRs: FCS_HTTPS_EXT.1, FCS_TLS_EXT.1, FCS_IPSEC_EXT.1, FIA_PSK_EXT.1, and FPT_ITT.1.
- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 4, September 2012.
  - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1 Revision 4, September 2012.
  - Part 3 Conformant

1.3 Conventions

The following conventions have been applied in this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.

- Iteration: allows a component to be used more than once with varying operations. In the ST, iteration is indicated by a number in parentheses placed at the end of the component. For example FDP_ACC.1(1) and FDP_ACC.1(2) indicate that the ST includes two iterations of the FDP_ACC.1 requirement, (1) and (2).
- Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]). Note that an assignment within a selection would be identified in italics and with embedded bold brackets (e.g., [***[selected-assignment]***]).
- Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., [***selection***]).
- Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …"). Note that 'cases' that are not applicable in a given SFR have simply been removed without any explicit identification.

- The NDPP uses an additional convention – the 'case' – which defines parts of an SFR that apply only when corresponding selections are made or some other identified conditions exist. Only the applicable cases are identified in this ST and they are identified using **bold** text.
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

1.3.1 Terminology

This section identifies product-specific terminology.

Term	Definition
Knowledge Base	A LogRhythm package that includes both required and optional content shared across a LogRhythm deployment. It consists of the core Knowledge Base as well as modules. The core Knowledge Base includes content applicable to all deployments, such as log processing rules, policies, and classifications.
Object	Resource such as a file, file path, or registry key that is referenced or impacted by log activity.
System Objects	A type of object that is created by LogRhythm Labs and imported with the Knowledge Base.
Custom Objects	A type of Object that is created by an end user.
Private	A permission value for an object that can only be managed by the owner.
Public	A permission value for an object that generally means all users have access to view the object. May have edit restrictions.
Global	A permission value that provides general global access for the object.

Table 1 TOE Specific Terminology

1.3.2 Abbreviations

This section identifies abbreviations and acronyms used in this ST.

AES	Advanced Encryption Standard
ASCII	American Standard Code for Information Interchange
CAVP	Cryptographic Algorithm Validation Program
CC	Common Criteria
CCTL	Common Criteria Testing Laboratory
CEM	Common Evaluation Methodology

CSTL	Cryptographic and Security Testing Laboratory
DRBG	Deterministic Random Bit Generator
FIPS	Federal Information Processing Standard
HMAC	Hash-based Message Authentication Code
HTTPS	Hyper-Text Transport Protocol Secure
IA	Initial Assessment
IAD	Information Assurance Directorate
IP	Internet Protocol
IPsec	Internet Protocol security
LR	LogRhythm
MMC	Microsoft Management Console
NDPP	Protection Profile for Network Devices
NIAP	National Information Assurance Partnership
NIST	National Institute of Standards and Technology
NSA	National Security Agency
NTP	Network Time Protocol
PP	Protection Profile
RSA	Rivest-Shamir-Adelman – a public-key cryptography algorithm
SAR	Security Assurance Requirement
SEIM	Security Information and Event Management
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SNMP	Simple Network Management Protocol
SSH	Secure Shell
SQL	Structured Query Language
ST	Security Target
TCP	Transport Control Protocol
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Function(s)
TSFI	TOE Security Function Interface(s)

2. TOE Description

The TOE is the LogRhythm Integrated Solution 6.3.4 consisting of the hardware and software components identified in Section 1.1.

The TOE is a network device consisting of several components that coordinate with one another to provide automated centralization of log collection and event management. The TOE collects information from multiple log sources (such as Windows events, syslog, flat file, NetFlow, sFlow, databases or applications). The product also provides File Integrity Monitoring and Machine Analytics, with Host and Network Forensics, in a unified Security Intelligence Platform. It analyzes collected data and provides tools to view and analyze IDS results and to issue alerts of significant events; however this functionality is outside the scope of the evaluation. The LogRhythm services support running on both IPv4 and IPv6 networks, however, IPv6 was not tested as part of this evaluation.

The only capabilities covered by the evaluation are those in the NDPP; all other capabilities are not covered.

The LogRhythm Console[1] provides the user interface into a LogRhythm deployment. The Console is a Windows .NET-based client application. Authenticated users can view logs, events, and reports. The Console also provides real-time monitoring and incident management. In addition, the Console provides interfaces for TOE configuration and user management. All communications between the Console and LogRhythm servers (EM and LMs) are via SQL Server protocols. The Web Console is a component on the LogRhythm Web Services appliance that provides accessibility to customized dashboards and analytical tools accessible by web browsers.

2.1 TOE Overview

The LogRhythm 6.3.4 TOE consists of the hardware and software components identified in Section 1.1.

All TOE appliances are pre-configured with Windows Server 2008 R2. A SQL Server instance resides on each Log Manager server and on the Event Manager server. The Event Manager (EM) and Log Manager (LM) can reside on the same server (XM) for low-volume deployments, or on dedicated servers for high volume deployments. Each AI Engine Server is always a standalone server. The Site Log Forwarder (SLF) is a standalone appliance. The Client Console is a Windows application that runs on either the EM appliance or the XM appliance in the evaluated configuration.

The Site Log Forwarder (SLF) is system monitor agent software which is delivered pre-installed on an appliance in the evaluated configuration. The SLF converts collected logs to ASCII text strings, which can be encrypted before forwarding across untrusted networks (e.g. internet). The logs are forwarded to a LM, where they are analyzed and written to a centralized database in the LM and also archived on a file system. LogRhythm also offers software only System Monitor Agents that can be deployed on remote Windows, Linux, Solaris, HP-UX or AIX systems to collect logs from most sources including Windows events, syslog, flat file, NetFlow, databases or applications. The software only System Monitor Agents are excluded from the evaluated configuration.

2.2 TOE Architecture

The LogRhythm Site Log Forwarder[2], Log Manager(s), AI Engine Server(s), Event Manager, Console(s) appliances, with the LogRhythm software, running Windows Server 2008 R2 and SQL Server 2008 software constitute the TOE. The Web Server appliance includes a Nginx web server. The TOE is delivered pre-configured on dedicated appliances. See Section 2.2.1 below for more detail.

The following paragraphs describe the basic functionality of the LogRhythm suite in the system monitoring and logging. Most of this functionality, with the exception of protection of inter-appliance communications and storage of the audit logs, is not covered by the evaluation. The specific areas covered by the evaluation are only those discussed in Section 2.2.2, and focus on protection, restriction, and monitoring of TOE administrator actions and protection of communications, both with the administrator and between components. Any other function described in this section to aid in the understanding of the product as a whole, not to imply its coverage by the evaluation.

[1] Also referred to as the "client console" to distinguish it from the "web console".
[2] The SLF contains a System Monitor Agent and is labeled "LogRhythm Agent" in Figure 1.

The following figure depicts the TOE components within their environment and shows communications among the components and operational environment devices. SQL Server is an internal component of Log Managers and Event Manager and is not shown in the figure. The figure depicts examples of collection sources including syslog, snmp, netflow and sflow devices but the TOE can also collect Windows events, flat file, databases or applications.

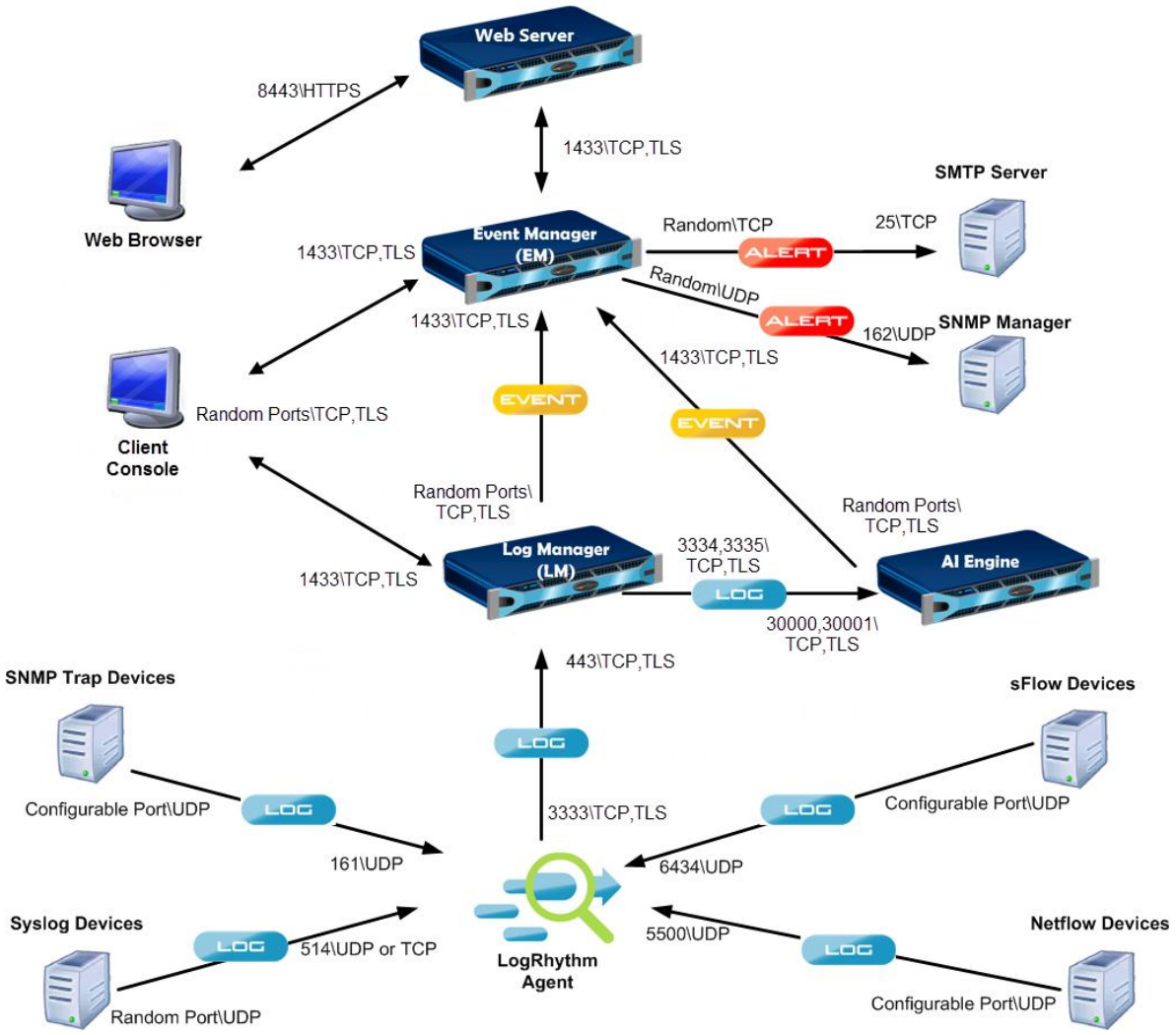

Figure 1 TOE Components

In general, remote log information flows to the SLF. The SLF in turn passes information to the Log Managers and AI Engine Servers to the Event Manager where a SQL Server is used internally to store log information. Log Managers analyze individual log messages and identify Events. The SLFs collect logs that were stored in various locations, e.g., Windows Event Log, SQL Server trace files and converts collected logs to ASCII text strings, which can be encrypted before forwarding across untrusted networks (e.g. Internet). An Event is a log message or collection of log messages that LogRhythm determines to be important or interesting. AI Engine Servers analyze log metadata gleaned from sets of log messages to identify more complex Events. The Event Manager processes Events and raises alarms as appropriate. Administrators use the Client Console to configure LogRhythm (for example, selecting rules that identify Events) and to view log reports and analyses. The alarms may be viewed by the Web Console or Client Console. Optionally the alarms may be sent to an external SMTP Server or SNMP Server in the operational environment. Alarms are not in the scope of evaluation and were not tested.

Each SLF forwards logs to the LM that is configured to receive them, where they are analyzed against defined Knowledge Base rules, written to a centralized database in the LM, and also archived on a file system. SLF communications with LM(s) are authenticated and encrypted via FIPS 140-2 certified TLS. Each LM consists of a SQL Server 2008 R2 instance and a LogRhythm Mediator Server. The Mediator Server takes in log messages (collected and forwarded by SLF and processes them against Knowledge Base rules that identify and categorize the log messages. The applied Knowledge Base rules determine whether the Mediator Server forwards log metadata to an AI Engine or forwards the log message to the EM as an Event or both. The Mediator Server is also responsible for writing incoming logs to an active archive, which is a file on the file system of the LM Host. Once that active archive file reaches a certain size or age (administrative configurable), the active archive is converted to an inactive archive file. During that conversion, the contents are SHA-1 hashed and then compressed. The SHA-1 hash value is stored in a database table within the LogRhythm Event Manager. If there is a restore request of the logs contained within the inactive archives, the SHA-1 hash is verified to ensure that the file has not been altered since being sealed. Communications between LM and AI Engine Server and between LM and EM are protected by FIPS 140-2 certified TLS. Updates to the Knowledge Base rules can be obtained by licensed customers at the LogRhythm's website. Though the Knowledge Base server and Knowledge Base Rules themselves have not been subject to evaluation; the communications with the Knowledge Base server is protected using TLS/HTTPS and this has been tested. The Knowledge Base Server communicates with the EM. The EM, via the Client Console or Job Manager, contacts the Knowledge Base server and the Knowledge base updates are downloaded directly to the EM database.

An AI Engine Server consists of two services: AI Engine Communication Manager service and AI Engine service. The AI Engine Communication Manager receives log metadata from one or more Log Managers. It marshals the data for the AI Engine to process. Also, it maintains TLS connections with Log Managers. An AI Engine processes the data by applying AI rules to the set of log metadata collected over time. An AI rule can correlate multiple log messages to identify an Event, which the AI Engine sends to the EM.

The EM consists of two services: the LogRhythm Alarming and Response Manager (ARM) service and the Job Manager service together with a SQL Server instance. There is only one EM per deployment. The EM receives and maintains log information from the LMs that have been analyzed against the Knowledge Base rules and have been identified as Events. The EM receives Events corresponding to complex conditions from the AI Engine Server. The ARM service evaluates Alarm Rules to determine if an Event (or series of Events) should be alarmed on and, if so, what the response should be (e.g., sending e-mails to people on a notification list, sending SNMP traps, or perform a remediation action).

The Web Console is a component that comprises a single service on the LogRhythm Web Services appliance. To support the most common end-user activities, the user interface provides easy access to analytical tools, alarms, and customized dashboards. The Web Console includes graphic visualizations and guided workflows for both trained security analysts and non-technical users. The Web Console communication is protected by FIPS 140-2 certified TLS/HTTPS.

The Client Console provides the user interface into a LogRhythm deployment. The Client Console is a Windows .NET-based client application that provides authenticated users with the ability to view logs, Events, alarms and reports. The Client Console also provides real-time monitoring, incident management, and interfaces for TOE configuration and user management. All communications between the Client Console and LogRhythm servers (EM and LMs) are protected by FIPS 140-2 certified TLS.

The product also provides a programmatic interface to maintain the integrity of shared information distributed between LogRhythm and other external data sources. This includes the ability to automate the exchange and synchronization of configuration data to enhance administrative functions as well as extend monitoring and analysis functions. The LogRhythm Web Services are intended to provide broad interoperability. They are SOAP based, WS-1 Basic Profile 1.1 compliant. A WSDL 1.1 compliant descriptor is provided which can be used to generate proxy classes in .Net, JAVA, or other languages. The Web Services API is not included in the evaluated configuration.

Site Log Forwarder (SLF) – LogRhythm's SLF appliances contain an Agent-Less System Monitor Agent that collects log, flow, and machine data for secure transport from remote locations to LogRhythm LMs. An Agent-less collector means that an agent is not required to be installed on the log sources being collected from. SLFs additionally manage bandwidth consumption via collection scheduling and/ or compression of transmitted data.

Every TOE deployment will have one EM component, at least one LM component, AI Engine Server and at least one Site Log Forwarder (SLF) component along with the consoles.

2.2.1 Physical Boundaries

The LogRhythm 6.3.4 TOE consists of the hardware and software components identified in Section 1.1.

The LogRhythm components operate within the context of a Windows Server 2008 R2 operating system and require a SQL server database. For the purposes of evaluation against the NDPP, the Windows operating system and SQL database are included with the LogRhythm components in the TOE boundary, which is bounded by the physical hardware appliances on which they are installed.

Other than the specific software component installed on a given appliance, the only differences among the appliance models relate to speed, performance and capacity (as described in the tables below) and do not affect any of the claimed security functions. Each appliance has Windows Server 2008 R2 with SQL Server 2008 installed on the Log Manager and the Event Manager.

All-in-one (XM) - LogRhythm XM appliances provide all the capabilities of the EM and LM appliance on the same platform. Many deployments begin with an XM configuration providing a high performance solution in a single appliance.

Appliance Series	Appliance Model	Description
LR-XM4300	LR-XM4310 LR-XM4330 LR-XM4350 (combined EM/LM server)	Max Processing – 1,000 MPS CPU – 6 core Memory - 64 GB Storage - (Useable -1TB) (Raw 2 TB) Ethernet – Broadcom 5720 (2 x 1 GB)
LR-XM6300	LR-XM6310 LR-XM6330 LR-XM6350 (combined EM/LM server)	Max Processing – 5,000 MPS CPU – 12 core Memory - 128 GB Storage - (Useable -2TB)) (Raw 4 TB) Ethernet – Broadcom 5720 (4 x 1 GB)

Event Manager - The LogRhythm Event Manager server is a Windows Server system. There is one Event Manager per deployment. The Event Manager provides centralized event management, incident management, analysis, reporting, and configuration across a LogRhythm deployment.

Appliance Line	Description
LR-EM3350	Max Processing – N/A CPU – 6 core Memory - 64 GB Storage - (Useable -1TB) (Raw 2 TB) Ethernet – Broadcom 5720 (2 x 1 GB)
LR-EM5350 (dedicated EM server)	Max Processing – N/A CPU – 12 core Memory - 128 GB Storage - (Useable -2TB)) (Raw 4 TB) Ethernet – Broadcom 5720 (4 x 1 GB)
LR-EM7350 (dedicated EM server)	Max Processing – N/A CPU – 16 core Memory - 128 GB (Expandable 256 GB) Storage - (Useable -2TB)) (Raw 4 TB) Ethernet – Broadcom 5720 (4 x 1 GB)

Log Manager (LM) - LogRhythm LM appliances provide high performance, distributed and redundant log collection and management.. Each LogRhythm deployment has at least one Log Manager.

Appliance Series	Appliance Models	Description
LR-LM3300	LR-LM3310 LR-LM3350	Max Processing – 2,500 MPS CPU – 6 core Memory - 64 GB Storage - (Useable -1TB) (Raw 2 TB) Ethernet – Broadcom 5720 (2 x 1 GB)
LR-LM5300	LR-LM5310 LR-LM5350	Max Processing – 5,000 MPS CPU – 12 core Memory - 128 GB Storage - (Useable -2TB)) (Raw 4 TB) Ethernet – Broadcom 5720 (4 x 1 GB)
LR-LM7300	LR-LM7310 LR-LM7311 LR-LM7312 LR-LM7313 LR-LM7350 LR-LM7351 LR-LM7352 LR-LM7353 (dedicated LM servers)	Max Processing – 15,000 MPS CPU – 16 core Memory - 128 GB (Expandable 256 GB) Storage - (Useable -2TB)) (Raw 4 TB) Ethernet – Broadcom 5720 (4 x 1 GB) Note: The models are all the same system, but with different numbers of DAS attached and different LR licensing levels. The number of DAS for each model is the last number of model number. For example model number LR-LM7353 has 3 DAS.

Advanced Intelligence (AI) Engine - The AI Engine is a Windows Server system. It is LogRhythm's advanced analysis platform that performs correlation, pattern recognition, and behavioral analysis. It receives logs from the Log Manager Mediator's AI Engine Data Provider and sends events to the Event Manager. There are no databases for the AI Engine. It communicates with the Log Manager. It consists of the following services: AI Engine Communication Manager and AI Engine Server.

Appliance Line	Description
LR-AIE5310	Max Processing – 15,000 MPS CPU – 6 core Memory - 64 GB / (Expandable 128 GB) Storage - 550 GB Ethernet – Broadcom 5720 (4 x 1 GB)
LR-AIE7310	Max Processing – 30,000 MPS CPU – 16 core Memory - 128 GB / (Expandable 256 GB) Storage - 1 TB Ethernet – Broadcom 5720 (4 x 1 GB)
LR-AIE9310	Max Processing – 75,000 MPS CPU – 32 core Memory - 256 GB / (Expandable 512 GB) Storage - 1 TB Ethernet – Broadcom 5720 (4 x 1 GB)
MPS = Messages Per Second	

Site Log Forwarder (SLF) - The Site Log Forwarder is an appliance containing an Agent-Less System Monitor Agent. An Agent-less collector means that an agent is not required to be installed on the log sources being collected from. The System Monitor Agent, also just called Agent, provides local and remote log data collection across various operating systems including Windows, Linux, AIX, HPUX, and Solaris. It serves as a central log data collector, collecting logs from many devices, servers, databases, and applications, performing host activity monitoring and forwarding logs, via authenticated TCP connections, to the Log Manager. The communications channel with the various log sources are protected using IPsec, TLS or HTTPS. The communication channel from the SLF to the LM component is protected using TLS.

Appliance	Description
SLF3310	A System Monitor Agent: Max Collection Rate: 10,000 CPU – 6 core Memory - 16 GB RAM Ethernet – (2 x 1 GB) Physical Disk: 2 x 300 GB 10K RPM SAS RAID 1 278 GB usable Logical Volume: C Drive (200 GB) D Drive (78 GB)

The SLF collects audit data from the monitored sources. As such the operating environment includes devices being monitored by the TOE.

LogRhythm Console – In the evaluated configuration the Client Console is installed on one of the appliances (either the EM or XM) and communicates with both the LM and the EM and provides local administration. The Console provides interfaces for TOE configuration and user management. Identification, Authentication and administrative activity are audited to hold users accountable for their actions. The audit records are protected from unauthorized modification and deletion. All administrators must be identified and authenticated (authentication is performed either by the local Windows OS or by Active Directory in the operational environment).

Web Services Host[3] **–** The Web Services appliance is a component that hosts the LogRhythm Web Console that supports most common end-user activities, and provides a user interface for easy access to analytical tools, alarms, and customized dashboards.

Appliance	Description
WS3310	Max Processing – 1,000 MPS CPU – 6 core Memory - 32 GB Storage - (Useable -1TB) (Raw 2 TB) Ethernet – Broadcom 5720 (2 x 1 GB)

2.2.1.1 Additional Software Requirements

The TOE requires an NTP Server in the operational environment to ensure time is synched among the distributed components. The product provides time stamps for its own use derived from the system clock managed by the underlying operating system.

[3] The Web Console/ Web Services Host component is referred to as the Web Server in Figure 1.

2.2.1.2 Additional Hardware Requirements

The following hardware is used by the TOE and is considered part of the operational environment:

- NTP Server
- Knowledge Base Server
- Windows Active Directory

2.2.2 Logical Boundaries

This section identifies the security functions that the TOE provides, consistent with the functional requirements specified in the NDPP. These comprise the following:

- Security Audit
- Cryptographic Support
- Identification and Authentication
- Security Management
- Protection of the TOE Security Functions
- TOE Access
- Trusted Path/Channel.

2.2.2.1 Security audit

The TOE is designed to be able to generate logs for a wide range of security relevant events including the events specified in the NDPP. The TOE can be configured to store the logs locally so they can be accessed by an administrator.

The TOE operates as a log collector, receiving log events transmitted to it from a range of log sources. The TOE receives the logs over a trusted channel using IPsec or TLS.

2.2.2.2 Cryptographic support

The TOE is operated in FIPS mode and includes NIST-validated cryptographic algorithm implementations for asymmetric key generation, symmetric encryption and decryption, cryptographic signature services, cryptographic hashing services, keyed-hash message authentication services, deterministic random bit generation seeded from a suitable entropy source and key zeroization. The cryptographic algorithm implementations support cryptographic protocols used for secure communication—IPsec, TLS and HTTPS.

2.2.2.3 Identification and authentication

The TOE requires users (i.e., administrators) to be successfully identified and authenticated before they can access any security management functions available in the TOE. The TOE provides both local and remote administrative access. Local access is via a direct console connection. Remote access is via a thick client (the LogRhythm Client Console) secured by TLS or web-based graphical user interface (the Web Console) secured by TLS/HTTPS.

The TOE supports the local (i.e., on device) definition of administrators with usernames and passwords. Additionally, the TOE can be configured to use Active Directory to support, for example, centralized user administration.

2.2.2.4 Security management

The TOE provides a thick client (Client Console) and a web-based client (Web Console) as mechanisms to access its security management functions. Security management commands are limited to administrators and are available only after they have provided acceptable user identification and authentication data to the TOE.

2.2.2.5 Protection of the TSF

The TOE implements a number of features designed to protect itself to ensure the reliability and integrity of its security features.

It protects particularly sensitive data such as stored passwords and cryptographic keys so that they are not accessible even by an administrator. The TOE ensures that reliable time information is available (e.g., for log accountability) provided through the use of a NTP Server. The NTP server is considered part of the operational environment.

The TOE uses FIPS 140-2 certified cryptographic algorithms to protect communications between distributed TOE components.

The TOE includes functions to perform self-tests so that it might detect when it is failing. It also includes mechanisms so that the TOE itself can be updated while ensuring that the updates will not introduce malicious or other unexpected changes in the TOE.

2.2.2.6 TOE access

The TOE can be configured to display an informative banner that will appear prior to an administrator being permitted to establish an interactive session. The TOE subsequently will enforce an administrator-defined inactivity timeout value after which the inactive session will be terminated.

The TOE provides users (local and remote) the ability to terminate their own interactive sessions, by logging off of the TOE.

2.2.2.7 Trusted path/channels

The TOE protects interactive communication with remote administrators using HTTP over TLS or TLS. TLS ensures both integrity and disclosure protection. The TOE is configured by an administrator to receive external log records over a secure (IPsec or TLS protected) trusted channel.

The TOE supports external user authentication via Active Directory, over a secure IPsec communication between the TOE and Active Directory.

The TOE provides an interface to the Knowledge Base server and can be configured to automatically check for Knowledge Base updates. Communications with the Knowledge Base server are protected using TLS v1.0.

2.3 TOE Documentation

Log Rhythm offers a series of documents that describe the installation process for the TOE, as well as guidance for subsequent use and administration of the system security features.

- LogRhythm Help
- LogRhythm Web Console Installation Guide Version 6.3.4
- LogRhythm Web Console User Guide Version 6.3.4
- LogRhythm Compliance Overview AGD Supplement Guide
- LogRhythm Solution Software (LRSS) Installation Guide v6.3

.

3. Security Problem Definition

This security target includes by reference the Security Problem Definition (composed of organizational policies, threat statements, and assumptions) from the NDPP.

In general, the NDPP has presented a Security Problem Definition appropriate for network infrastructure devices, and as such is applicable to the LogRhythm TOE.

4. Security Objectives

Like the Security Problem Definition, this security target includes by reference the Security Objectives from the NDPP. The NDPP security objectives for the operational environment are reproduced below, since these objectives characterize technical and procedural measures each consumer must implement in their operational environment.

In general, the NDPP has presented a Security Objectives statement appropriate for network infrastructure devices, and as such is applicable to the LogRhythm TOE.

4.1 Security Objectives for the Operational Environment

OE.NO_GENERAL_PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.
OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
OE.TRUSTED_ADMIN	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.

5. IT Security Requirements

This section defines the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) that serve to represent the security functional claims for the Target of Evaluation (TOE) and to scope the evaluation effort.

The SFRs have all been drawn from the Protection Profile (PP): *Protection Profile for Network Devices*, Version 1.1, 8 June 2012 (NDPP), as amended by Errata #3. As a result, refinements and operations already performed in that PP are not identified (e.g., highlighted) here, rather the requirements have been copied from that PP and any residual operations have been completed herein. Of particular note, the NDPP made a number of refinements and completed some of the SFR operations defined in the CC and that PP should be consulted to identify those changes if necessary.

The SARs are the set of SARs specified in NDPP.

5.1 Extended Requirements

All of the extended requirements in this ST have been drawn from the NDPP. The NDPP defines the following extended SFRs and since they are not redefined in this ST, the NDPP should be consulted for more information in regard to those CC extensions.

- FAU_STG_EXT.1: External Audit Trail Storage
- FCS_CKM_EXT.4: Cryptographic Key Zeroization
- FCS_HTTPS_EXT.1: Extended: HTTP Security (HTTPS)
- FCS_IPSEC_EXT.1: Explicit IPSEC
- FCS_RBG_EXT.1: Extended: Cryptographic Operation (Random Bit Generation)
- FCS_TLS_EXT.1: Extended: Transport Layer Security (TLS)
- FIA_PMG_EXT.1: Password Management
- FIA_PSK_EXT.1: Extended: Pre-Shared Key Composition
- FIA_UAU_EXT.2: Extended: Password-based Authentication Mechanism
- FIA_UIA_EXT.1: User Identification and Authentication
- FPT_APW_EXT.1: Extended: Protection of Administrator Passwords
- FPT_SKP_EXT.1: Extended: Protection of TSF Data (for reading of all symmetric keys)
- FPT_TST_EXT.1: TSF Testing
- FPT_TUD_EXT.1: Extended: Trusted Update
- FTA_SSL_EXT.1: TSF-initiated Session Locking

5.2 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by the LogRhythm TOE.

Requirement Class	Requirement Component
FAU: Security audit	FAU_GEN.1: Audit Data Generation
	FAU_GEN.2: User identity association
	FAU_STG_EXT.1: External Audit Trail Storage
FCS: Cryptographic support	FCS_CKM.1: Cryptographic Key Generation (for asymmetric keys)
	FCS_CKM_EXT.4: Cryptographic Key Zeroization
	FCS_COP.1(1): Cryptographic Operation (for data encryption/decryption)
	FCS_COP.1(2): Cryptographic Operation (for cryptographic signature)
	FCS_COP.1(3): Cryptographic Operation (for cryptographic hashing)
	FCS_COP.1(4): Cryptographic Operation (for keyed-hash message authentication)
	FCS_IPSEC_EXT.1: Explicit: IPSEC
	FCS_HTTPS_EXT.1: Extended: HTTP Security (HTTPS)
	FCS_RBG_EXT.1: Extended: Cryptographic Operation (Random Bit Generation)
	FCS_TLS_EXT.1: Extended: Transport Layer Security (TLS)
FDP: User data protection	FDP_RIP.2: Full Residual Information Protection
FIA: Identification and authentication	FIA_PMG_EXT.1: Password Management
	FIA_PSK_EXT.1: Extended: Pre-Shared Key Composition
	FIA_UAU.7: Protected Authentication Feedback
	FIA_UAU_EXT.2: Extended: Password-based Authentication Mechanism
	FIA_UIA_EXT.1: User Identification and Authentication
FMT: Security management	FMT_MTD.1: Management of TSF Data (for general TSF data)
	FMT_SMF.1: Specification of Management Functions
	FMT_SMR.2: Restrictions on Security Roles
FPT: Protection of the TSF	FPT_APW_EXT.1: Extended: Protection of Administrator Passwords
	FPT_ITT.1: Basic Internal TSF Data Transfer Protection
	FPT_SKP_EXT.1: Extended: Protection of TSF Data (for reading of all symmetric keys)
	FPT_STM.1: Reliable Time Stamps
	FPT_TST_EXT.1: TSF Testing
	FPT_TUD_EXT.1: Extended: Trusted Update
FTA: TOE access	FTA_SSL.3: TSF-initiated Termination
	FTA_SSL.4: User-initiated Termination
	FTA_SSL_EXT.1: TSF-initiated Session Locking
	FTA_TAB.1: Default TOE Access Banners
FTP: Trusted path/channels	FTP_ITC.1: Trusted Channel
	FTP_TRP.1: Trusted Path

Table 2 TOE Security Functional Components

5.2.1 Security audit (FAU)

5.2.1.1 Audit Data Generation (FAU_GEN.1)

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shut-down of the audit functions;
b) All auditable events for the not specified level of audit; and
c) All administrative actions;
d) Specifically defined auditable events listed in **Table 3Table 2**.

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, information specified in column three **Table 3**.

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None.	
FAU_GEN.2	None.	
FAU_STG_EXT.1	None.	
FCS_CKM.1	None.	
FCS_CKM_EXT.4	None.	
FCS_COP.1(1)	None.	
FCS_COP.1(2)	None.	
FCS_COP.1(3)	None.	
FCS_COP.1(4)	None.	
FCS_HTTPS_EXT.1	Failure to establish a HTTPS session. Establishment/Termination of a HTTPS session.	Reason for failure. Non-TOE endpoint of connection (IP address) for both successes and failures
FCS_IPSEC_EXT.1	Failure to establish an IPsec SA. Establishment/Termination of an IPsec SA.	Reason for failure. Non-TOE endpoint of connection (IP address) for both successes and failures.
FCS_RBG_EXT.1	None.	
FCS_TLS_EXT.1	Failure to establish a TLS session. Establishment/Termination of a TLS session.	Reason for failure. Non-TOE endpoint of connection (IP address) for both successes and failures
FDP_RIP.2	None.	
FIA_PMG_EXT.1	None.	
FIA_UAU_EXT.2	All use of the authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UIA_EXT.1	All use of the identification and authentication mechanism.	Provided user identity, origin of the attempt (e.g., IP address).
FIA_UAU.7	None.	
FMT_MTD.1	None.	
FMT_SMF.1	None.	
FMT_SMR.2	None.	
FPT_APW_EXT.1	None.	
FPT_ITT.1	None.	
FPT_SKP_EXT.1	None.	
FPT_STM.1	Changes to the time.	The old and new values for the time. Origin of the attempt (e.g., IP address).
FPT_TUD_EXT.1	Initiation of update.	No additional information.
FPT_TST_EXT.1	None.	
FTA_SSL_EXT.1	Any attempts at unlocking of an interactive session.	No additional information.
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	No additional information.

Requirement	Auditable Events	Additional Audit Record Contents
FTA_SSL.4	The termination of an interactive session.	No additional information.
FTA_TAB.1	None.	
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1	Initiation of the trusted channel. Termination of the trusted channel. Failures of the trusted path functions.	Identification of the claimed user identity.

Table 3 Audit Requirements

5.2.1.2 User identity association (FAU_GEN.2)

FAU_GEN.2.1 For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

5.2.1.3 External Audit Trail Storage (FAU_STG_EXT.1)

FAU_STG_EXT.1.1 The TSF shall be able to **[*receive and store audit data from an external IT entity*]** using a trusted channel implementing the **[*IPsec, TLS, TLS/HTTPS*]** protocol.

5.2.2 Cryptographic support (FCS)

5.2.2.1 Cryptographic Key Generation (for asymmetric keys) (FCS_CKM.1)

FCS_CKM.1.1 **Refinement:** The TSF shall generate **asymmetric** cryptographic keys **used for key establishment** in accordance with

[

- ***NIST Special Publication 800-56B, "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography" for RSA-based key establishment schemes*]**

and specified cryptographic key sizes *equivalent to, or greater than, a symmetric key strength of 112 bits.*

5.2.2.2 Cryptographic Key Zeroization (FCS_CKM_EXT.4)

FCS_CKM_EXT.4.1 The TSF shall zeroize all plaintext secret and private cryptographic keys and CSPs when no longer required.

5.2.2.3 Cryptographic Operation (for data encryption/decryption) (FCS_COP.1(1))

FCS_COP.1(1).1 **Refinement:** The TSF shall perform encryption and decryption in accordance with a specified cryptographic algorithm [AES operating in [***CBC***]] and cryptographic key sizes 128-bits and 256-bits that meets the following:

- FIPS PUB 197, 'Advanced Encryption Standard (AES)'
- [***NIST SP 800-38A].***

5.2.2.4 Cryptographic Operation (for cryptographic signature) (FCS_COP.1(2))

FCS_COP.1.1(2) **Refinement:** The TSF shall perform **cryptographic signature services** in accordance with a [***(2) RSA Digital Signature Algorithm (rDSA) with a key size (modulus) of 2048 bits or greater,*** ~~***or***~~

]

that meets the following:

Case: RSA Digital Signature Algorithm

- **FIPS PUB 186-2 or FIPS PUB 186-3, "Digital Signature Standard"**

5.2.2.5 Cryptographic Operation (for cryptographic hashing) (FCS_COP.1(3))

FCS_COP.1.1(3) **Refinement:** The TSF shall perform [*cryptographic hashing services*] in accordance with a specified cryptographic algorithm [***SHA-1*] and message digest sizes [*160*] bits** that meet the following: *FIPS Pub 180-3 "Secure Hash Standard."*

Cryptographic Operation (for keyed-hash message authentication) (FCS_COP.1(4))

FCS_COP.1.1(4) **Refinement:** The TSF shall perform [*keyed-hash message authentication*] in accordance with a specified cryptographic algorithm HMAC-[***SHA-1*], key size [160], and message digest sizes [*160*] bits** that meet the following: *FIPS Pub 198-1 "The Keyed-Hash Message Authentication Code", and FIPS PUB 180-3, "Secure Hash Standard."*

5.2.2.6 Extended: HTTP Security (HTTPS) (FCS_HTTPS_EXT.1)

FCS_HTTPS_EXT.1.1 The TSF shall implement the HTTPS protocol that complies with RFC 2818.

FCS_HTTPS_EXT.1.2 The TSF shall implement HTTPS using TLS as specified in FCS_TLS_EXT.1.

5.2.2.7 Explicit: IPSEC (FCS_IPSEC_EXT.1)

FCS_IPSEC_EXT.1.1 The TSF shall implement the IPsec architecture as specified in RFC 4301.

FCS_IPSEC_EXT.1.2 The TSF shall implement [***transport mode***].

FCS_IPSEC_EXT.1.3 The TSF shall have a nominal, final entry in the SPD that matches anything that is otherwise unmatched, and discards it.

FCS_IPSEC_EXT.1.4 The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using **[*the cryptographic algorithms AES-CBC-128 (as specified by RFC 3602) together with a Secure Hash Algorithm (SHA)-based HMAC, AES-CBC-256 (as specified by RFC 3602) together with a Secure Hash Algorithm (SHA)-based HMAC, [no other algorithms]*]**.

FCS_IPSEC_EXT.1.5 The TSF shall implement the protocol: **[*IKEv1 as defined in RFCs 2407, 2408, 2409, RFC 4109, [no other RFCs for extended sequence numbers], and [no other RFCs for hash functions]*]**.

FCS_IPSEC_EXT.1.6 The TSF shall ensure the encrypted payload in the **[*IKEv1*]** protocol uses the cryptographic algorithms AES-CBC-128, AES-CBC-256 as specified in RFC 6379 and **[*no other algorithm*]**.

FCS_IPSEC_EXT.1.7 The TSF shall ensure that IKEv1 Phase 1 exchanges use only main mode.

FCS_IPSEC_EXT.1.8 The TSF shall ensure that **[*IKEv1 SA lifetimes can be established based on [number of packets/number of bytes ; length of time, where the time values can be limited to: 24 hours for Phase 1 SAs and 8 hours for Phase 2 SAs]*]**.

FCS_IPSEC_EXT.1.9 The TSF shall ensure that all IKE protocols implement DH Groups 14 (2048-bit MODP), and **[*no other DH groups*]**.

FCS_IPSEC_EXT.1.10 The TSF shall ensure that all IKE protocols perform Peer Authentication using the **[*RSA*]** algorithm and Pre-shared Keys.

5.2.2.8 Extended: Cryptographic Operation (Random Bit Generation) (FCS_RBG_EXT.1)

FCS_RBG_EXT.1.1 The TSF shall perform all random bit generation (RBG) services in accordance with [***NIST Special Publication 800-90 using* [*CTR_DRBG (AES)*]]** seeded by an entropy source that accumulated entropy from [***a software-based noise source, a TSF-hardware-based noise source***].

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded with a minimum of [***256 bits***] of entropy at least equal to the greatest security strength of the keys and hashes that it will generate.

5.2.2.9 Extended: Transport Layer Security (TLS) (FCS_TLS_EXT.1)

FCS_TLS_EXT.1.1 The TSF shall implement one or more of the following protocols [***TLS 1.0 (RFC 2246), TLS 1.2 (RFC 5246)***] supporting the following ciphersuites:
Mandatory Ciphersuites:
TLS_RSA_WITH_AES_128_CBC_SHA

Optional Ciphersuites:
[*None*].

5.2.3 User data protection (FDP)

5.2.3.1 Full Residual Information Protection (FDP_RIP.2)

FDP_RIP.2.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the [selection: *allocation of the resource to, deallocation of the resource from*] all objects.

***Application Note**: There are no packets that flow "through" the TOE, so the TOE does not handle any resources that would be subject to this requirement. In this case, the requirement is implicitly met and the selection does not need to be made.*

5.2.4 Identification and authentication (FIA)

5.2.4.1 Password Management (FIA_PMG_EXT.1)

FIA_PMG_EXT.1.1 The TSF shall provide the following password management capabilities for administrative passwords:

1. Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [***"!", "@", "#", "$", "%", "^", "&", "*", "(", ")", [no other characters]***];
2. Minimum password length shall be settable by the Security Administrator, and support passwords of 15 characters or greater;

5.2.4.2 Extended: Pre-Shared Key Composition (FIA_PSK_EXT.1)

FIA_PSK_EXT.1.1 The TSF shall be able to use pre-shared keys for IPsec.

FIA_PSK_EXT.1.2 The TSF shall be able to accept text-based pre-shared keys that:

- are 22 characters and **[*[from 1 to 100 characters long]*]**;
- composed of any combination of upper and lower case letters, numbers, and special characters (that include: "!", "@", "#", "$", "%", "^", "&", "*", "(", and ")").

FIA_PSK_EXT.1.3 The TSF shall condition the text-based pre-shared keys by using **[*[the bit values of the ASCII characters directly as the key]*]** and be able to **[*use no other pre-shared keys*]**.

Protected Authentication Feedback (FIA_UAU.7)

FIA_UAU.7.1 The TSF shall provide only obscured feedback to the administrative user while the authentication is in progress at the local console.

5.2.4.3 Extended: Password-based Authentication Mechanism (FIA_UAU_EXT.2)

FIA_UAU_EXT.2.1 The TSF shall provide a local password-based authentication mechanism, [***[remote authentication via Active Directory]***] to perform administrative user authentication.

5.2.4.4 User Identification and Authentication (FIA_UIA_EXT.1)

FIA_UIA_EXT.1.1 The TSF shall allow responses to the following actions prior to requiring the non-TOE entity to initiate the identification and authentication process:

- Display the warning banner in accordance with FTA_TAB.1;
- **[*[no other actions]*]**.

FIA_UIA_EXT.1.2 The TSF shall require each administrative user to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that administrative user.

5.2.5 Security management (FMT)

5.2.5.1 Management of TSF Data (for general TSF data) (FMT_MTD.1)

FMT_MTD.1.1 The TSF shall restrict the ability to manage the TSF data to the Security Administrators.

5.2.5.2 Specification of Management Functions (FMT_SMF.1)

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions:

- Ability to administer the TOE locally and remotely;
- Ability to update the TOE, and to verify the updates using the [***digital signature***] capability prior to installing those updates;
- **[*Ability to configure the cryptographic functionality*]**

5.2.5.3 Restrictions on Security Roles (FMT_SMR.2)

FMT_SMR.2.1 The TSF shall maintain the roles:

- Authorized Administrator.

FMT_SMR.2.2 The TSF shall be able to associate users with roles.

FMT_SMR.2.3 The TSF shall ensure that the conditions

- Authorized Administrator role shall be able to administer the TOE locally;
- Authorized Administrator role shall be able to administer the TOE remotely;

are satisfied.

5.2.6 Protection of the TSF (FPT)

5.2.6.1 Extended: Protection of Administrator Passwords (FPT_APW_EXT.1)

FPT_APW_EXT.1.1 The TSF shall store passwords in non-plaintext form.

FPT_APW_EXT.1.2 The TSF shall prevent the reading of plaintext passwords.

5.2.6.2 Extended: Protection of TSF Data (for reading of all symmetric keys) (FPT_SKP_EXT.1)

FPT_SKP_EXT.1.1 The TSF shall prevent reading of all pre-shared keys, symmetric key, and private keys.

5.2.6.3 Basic Internal TSF Data Transfer Protection

FPT_ITT.1.1 **Refinement:** The TSF shall protect TSF data from disclosure and detect its modification when it is transmitted between separate parts of the TOE through the use **of** [***TLS***].

5.2.6.4 Reliable Time Stamps (FPT_STM.1)

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps for its own use.

5.2.6.5 TSF Testing (FPT_TST_EXT.1)

FPT_TST_EXT.1.1 The TSF shall run a suite of self tests during initial start-up (on power on) to demonstrate the correct operation of the TSF.

5.2.6.6 Extended: Trusted Update (FPT_TUD_EXT.1)

FPT_TUD_EXT.1.1 The TSF shall provide security administrators the ability to query the current version of the TOE firmware/software.

FPT_TUD_EXT.1.2 The TSF shall provide security administrators the ability to initiate updates to TOE firmware/software.

FPT_TUD_EXT.1.3 The TSF shall provide a means to verify firmware/software updates to the TOE using a [***digital signature mechanism***] prior to installing those updates.

5.2.7 TOE access (FTA)

5.2.7.1 TSF-initiated Termination (FTA_SSL.3)

FTA_SSL.3.1 **Refinement:** The TSF shall terminate a remote interactive session after a Security Administrator-configurable time interval of session inactivity.

5.2.7.2 User-initiated Termination (FTA_SSL.4)

FTA_SSL.4.1 The TSF shall allow Administrator-initiated termination of the Administrator's own interactive session.

5.2.7.3 TSF-initiated Session Locking (FTA_SSL_EXT.1)

FTA_SSL_EXT.1.1 The TSF shall, for local interactive sessions, [***terminate the session***] after a Security Administrator-specified time period of inactivity.

5.2.7.4 Default TOE Access Banners (FTA_TAB.1)

FTA_TAB.1.1 **Refinement:** Before establishing an administrative user session the TSF shall display a Security Administrator-specified advisory notice and consent warning message regarding use of the TOE.

5.2.8 Trusted path/channels (FTP)

5.2.8.1 Trusted Channel (FTP_ITC.1)

FTP_ITC.1.1 **Refinement:** The TSF shall use [***IPsec***, ***TLS***] to provide a **trusted** communication channel between itself and **authorized IT entities supporting the following capabilities: ~~audit server~~,** [***remote authentication via Active Directory*, [updates from the Knowledge Base server, audit sources,]**] that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data **from disclosure and detection of modification of the channel data**.

FTP_ITC.1.2 The TSF shall permit the TSF, or the authorized IT entities to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for [**remote authentication via Active Directory, updates from the Knowledge Base server, audit sources,**].

Application Note: *The TOE is an audit server where non-TOE entities send audit data to the TOE. The TOE does not send any audit records to an audit server. Therefore, the audit server requirement is not applicable in FTP_ITC.1.1and is identified with a strike-through.*

5.2.8.2 Trusted Path (FTP_TRP.1)

FTP_TRP.1.1 Refinement: The TSF shall use [***TLS/HTTPS***] **to** provide a communication path between itself and remote administrators that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from disclosure and detection of modification of the communicated data.

FTP_TRP.1.2	Refinement: The TSF shall permit remote administrators to initiate communication via the trusted path.
FTP_TRP.1.3	The TSF shall require the use of the trusted path for initial administrator authentication and all remote administrative actions.

5.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are included by reference from the NDPP.

Requirement Class	Requirement Component
ADV: Development	ADV_FSP.1 Basic functional specification
AGD: Guidance documents	AGD_OPE.1: Operational user guidance
	AGD_PRE.1: Preparative procedures
ALC: Life-cycle support	ALC_CMC.1 Labelling of the TOE
	ALC_CMS.1 TOE CM coverage
ATE: Tests	ATE_IND.1 Independent testing - conformance
AVA: Vulnerability assessment	AVA_VAN.1 Vulnerability survey

Table 4 Assurance Components

Consequently, the assurance activities specified in NDPP apply to the TOE evaluation.

6. TOE Summary Specification

This chapter describes the security functions:

- Security audit
- Cryptographic support
- User data protection
- Identification and authentication
- Security management
- Protection of the TSF
- TOE access
- Trusted path/channels

6.1 Security audit

The TOE functions as an audit server — while it provides significant capabilities for log analysis, correlation and alerting, all these capabilities rest on the ability of the TOE to receive logs from myriad external sources using a trusted channel that implements the IPsec or TLS protocol. The TOE generates audit records as described above in **Table 3** above.

The TOE is designed to generate the audit records as identified in the NDPP. The events that can cause an audit record to be logged include starting and stopping the audit function, any use of an administrator command via the Web Console and the LogRhythm Client Console, as well as all of the events identified in **Table 3.** The only protocol (i.e., IPsec, TLS/HTTPS, TLS) failures auditable by the TOE are authentication failures for user-level connections.

The logged audit records identify the date and time, the nature or type of the triggering event, an indication of whether the event succeeded or failed, and the identity of the user responsible for the event. The logged audit records also include event-specific content that includes at least all of the content required in **Table 3**.

The LogRhythm Audit Generation and storage make use of a SQL Server audit trace that is output by SQL Server in the form of trace files. These trace files contain the audit data. The SQL Server trace will capture all required audit events and will produce trace files (.trc) at a configurable location. Filters can be configured to exclude LogRhythm service activity from being included in the LogRhythm Audit Generated data.

Once the audit trace is defined and started, SQL Server will begin writing audit events to the trace at a location configured within the LogRhythm_Audit stored procedure. The LogRhythm_Audit stored procedure also configures the maximum size an audit trace is allowed to become before a new file is started (i.e. log rotation). An LR deployment requires a minimum of 5% of the available storage for system audit data. This value is 100MB by default and includes LR diagnostics and other events generated by the system.

The LogRhythm_Audit stored procedure defines the SQL Audit trace in such a way that if the audit trail becomes full (i.e. exhaustion of hard drive space) the following events occur:

1. No more audit data will be written to the audit trace file
2. 100% of the existing audit data will be retained
3. The SQL Server instance will shut down preventing any user from performing auditable actions
4. The SQL Server Agent will shut down preventing any database jobs from running
5. SQL Server will write the notice of the audit write error to the Application Event Log on the host system

The TOE stores the audit records locally and protects them from unauthorized deletion by allowing only authorized administrators to gain access to the audit trail. The audit trace files will typically reside on an NTFS file system at a location determined by the LogRhythm_Audit configuration. Because the trace files reside on a file system in order to restrict access to them, proper discretionary access controls must be implemented at the file system level. In general, permissions are granted to these trace files for read access and/or maintenance and will vary from environment to environment. The file system folder that the LogRhythm audit traces are written to (the trace folder) must be locked down with appropriate discretionary access controls to prevent access, modification, or deletion at the file system level.

The TOE must be configured to record the audit logs involved in the IPsec communication. The Windows MMC Console is used to configure the Local Computer Policy to record the success and failures of the IPsec connections in the Windows Security Event log. Administrative changes to the system time and initiation of an update are also recorded in the Window Security Event Log. Windows protects against the loss of events through a combination of controls associated with audit queuing and event logging. As configured in the TOE, audit data is appended to the audit log until it is full. The TOE protects against lost audit data by allowing the authorized administrator to configure the system to generate an audit event when the security log reaches a specified capacity percentage (e.g., 90%). Additionally, the authorized administrator can configure the system not to overwrite events and to shutdown when the security log is full. When so configured, after the system has shutdown due to audit overflow, only the authorized administrator can log on. When the security log is full, a message is written to the terminal display of the authorized administrator indicating the audit log has overflowed.

The Security audit function is designed to satisfy the following security functional requirements:

- FAU_GEN.1: The TOE can generate audit records for events including starting and stopping the audit function, administrator commands, and all other events identified in **Table 3**. Furthermore, each audit record identifies the date/time, event type, outcome of the event, responsible subject/user, as well as the additional event-specific content indicated in **Table 3**.
- FAU_GEN.2: The TOE associates each auditable event with the identity of the user that caused the event.
- FAU_STG_EXT.1: The TOE receives and stores audit data from external IT entities using a trusted channel that implements the IPsec and TLS protocol. TLS 1.0 is used between the TOE and the external IT audit source entities that it communicates with via TLS.

6.2 Cryptographic support

The TOE provides a FIPS mode of operation, which must be enabled in the evaluated configuration. The TOE includes NIST-validated cryptographic algorithms providing supporting cryptographic functions. The TOE relies on the FIPS 140-2 validated cryptographic module, certificate #1336 for the Microsoft Windows Server 2008 R2 Cryptographic Primitives Library (bcryptprimitives.dll). The evaluation leverages the Cryptographic Algorithm Validation Program (CAVP) certifications obtained as part of that validation.

The TOE provides NIST-validated cryptographic algorithm implementations for asymmetric key generation, symmetric encryption and decryption, cryptographic signature services, cryptographic hashing services, keyed-hash message authentication services, deterministic random bit generation seeded from a suitable entropy source and key zeroization. The cryptographic algorithm implementations support cryptographic protocols used for secure communication—IPsec, TLS and HTTPS.

The following functions have been certified in accordance with the identified standards.

Functions	Standards	Certificates

Asymmetric key generation		
• Domain parameter generation (key size 2048 bits)	NIST Special Publication 800-56B	RSA # 559
Encryption/Decryption		
• AES CBC (128 and 256 bits)	FIPS PUB 197 NIST SP 800-38A	AES # 1168
Cryptographic signature services		
• RSA Digital Signature Algorithm (rDSA) (modulus 2048)	FIPS PUB 186-2 FIPS PUB 186-3	RSA # 559 RSA # 567
Cryptographic hashing		
• SHA-1 (digest sizes 160 bits))	FIPS PUB 180-3	SHS # 1081
Keyed-hash message authentication		
• HMAC-SHA-1	FIPS PUB198-1 FIPS PUB 180-3	HMAC # 686
Random bit generation		
• CTR-DRBG(AES) with an independent hardware-based noise source and software-based noise source that provide 256 bits of non-determinism	NIST Special Publication 800-90A	DRBG # 23

Table 5 Cryptographic Functions

The TSF includes a key isolation service designed specifically to host secret and private keys in a protected process to mitigate tampering or access to sensitive key materials. The TSF performs key entry and output in accordance with FIPS 140-2. The TSF performs a key error detection check on each transfer of key (internal, intermediate transfers). The TSF prevents archiving of expired (private) signature keys. The TSF destroys non-persistent cryptographic keys – note that all keys subject to destruction are stored within the cryptomodule that was subject to FIPS 140-2 certification - after a cryptographic administrator-defined period of time of inactivity.
The TSF overwrites each intermediate storage area for plaintext key/critical cryptographic security parameter (i.e., any storage, such as memory buffers, that is included in the path of such data). This overwriting is performed as follows:

- For non-volatile memories other than EEPROM and Flash, the overwrite is executed three or more times using a different alternating data pattern each time upon the transfer of the key/critical cryptographic security parameter to another location.
- For volatile memory and non-volatile EEPROM and Flash memories, the overwrite is a single direct overwrite consisting of a pseudo random pattern, followed by a read-verify upon the transfer of the key/critical cryptographic security parameter to another location.

The TSF implements the HTTPS protocol that complies with RFC 2818 and TLS 1.2 that complies with RFC 5246. The TOE protects the interactive communication with administrators using HTTPS over TLS 1.2, which provides confidentiality of transmitted information and detects any loss of integrity. The TOE generates asymmetric cryptographic keys for RSA-based key establishment schemes in accordance with Sections 5 through 8 of SP 800-56B. The TOE is a distributed TOE in which secure communications between the components are provided by the use of TLS 1.0 with the data encrypted with the TLS_RSA_WITH_AES_128_CBC_SHA ciphersuite. TLS 1.0 is also used between the TOE and the external IT entities that it communicates with via TLS which are the Knowledge Base Server and audit sources.

The TOE uses a hardware-based and software-based deterministic random bit generator that complies with NIST SP 800-90, using CTR_DRBG (AES). The TOE uses the BCRYPTPRIMITIVES.DLL which implements the CTR_DRBG (AES) (CAVP DRBG #23). The DRBG is seeded from the output of an in-kernel random number generator. The in-kernel random number generator generates random bytes by taking the output of a cascade of two SP800-90 AES-256 counter mode based PRNGs seeded from the Windows entropy pool.

The Windows entropy pool is populated by periodically gathering random bits from the Trusted Platform Module (TPM), as well as by periodically querying the values of various OS Variables, including:

- The process ID of the currently running process
- The thread ID of the currently running thread
- A 32-bit tick count since the system boot
- The current local date and time
- The current system time of day information consisting of the boot time, current time, time zone bias, time zone ID, boot time bias, and sleep time bias
- The current hardware-platform-dependent high-resolution performance-counter value
- Information about the system's current usage of physical and virtual memory, and the page file
- System device information
- Local disk information
- A hash of the environment block for the current process
- Some hardware CPU-specific cycle counters
- System file cache information
- System processor power information
- System page file information
- System processor idle information
- System processor performance information
- System exception information
- System look-aside information
- System processor performance information
- System interrupt information
- System process information.

From the perspective of the LogRhythm application, the entropy source is considered a third-party entropy source as it is embedded within the proprietary Windows Server 2008 R2 operating system on which the LogRhythm software runs. It is assumed the entropy source provides a full 256 bits of entropy when seeding the in-kernel PRNG that ultimately provides the seed for the CTR_DRBG(AES) implemented in BCRYPTPRIMITIVES.DLL.

ID	Key Type	Size	Description	Origin	Storage	Zeroization Method
Secret and Private Keys						
TLS private key	RSA	2048 - bits	Used for TLS session establishment	N/A (entered through Windows operating system)	Volatile memory and the operating system	Zeroization is an overwrite consisting of a single direct overwrite consisting of a pseudo random pattern

ID	Key Type	Size	Description	Origin	Storage	Zeroization Method
TLS session encryption keys	AES	128 bits	Used for TLS communication	Generated through TLS handshake	Plaintext in volatile memory	Zeroization is an overwrite consisting of a single direct overwrite consisting of a pseudo random pattern
TLS session integrity keys	HMAC-SHA1	160 bits	Used for TLS communication	Generated through TLS handshake	Plaintext in volatile memory	Zeroization is an overwrite consisting of a single direct overwrite consisting of a pseudo random pattern
Public Keys						
TLS public key	RSA	2048 - bits	Used for TLS communication with Log Managers and Event Manager SQL Server, AI Engine Server, Windows System Monitor Agents, and SQL Servers	N/A (entered through Windows operating system)	Volatile memory and the operating system	Zeroization is an overwrite consisting of a single direct overwrite consisting of a pseudo random pattern
Log Manager public key	RSA	2048 - bits	Used for TLS communication with Log Managers	N/A (entered through TLS handshake)	Volatile memory	Zeroization is an overwrite consisting of a single direct overwrite consisting of a pseudo random pattern
Windows System Monitor Agent public keys	RSA	2048 - bits	Used for TLS communication with Windows System Monitor Agents	N/A (entered through TLS handshake)	Volatile memory	Zeroization is an overwrite consisting of a single direct overwrite consisting of a pseudo random pattern

ID	Key Type	Size	Description	Origin	Storage	Zeroization Method
AI Engine Server public key	RSA	2048 - bits	Used for TLS communication with AI Engine Server	N/A (entered through TLS handshake)	Volatile memory	Zeroization is an overwrite consisting of a single direct overwrite consisting of a pseudo random pattern
CA public Key	RSA	2048 - bits	Used for TLS communication with Log Managers and Event Manager SQL Server AI Engine Server, Windows System Monitor Agents, and SQL Servers	N/A (entered through Windows operating system)	Volatile memory and the operating system	Zeroization is an overwrite consisting of a single direct overwrite consisting of a pseudo random pattern
SQL Server public key	RSA	2048 - bits	Used for TLS communication with Event Manager SQL Server	N/A (entered through TLS handshake)	Volatile memory	Zeroization is an overwrite consisting of a single direct overwrite consisting of a pseudo random pattern
IPsec Critical Security Parameters						
IKE pre-shared keys	Shared Secret	1 – 100 Bytes	Pre-Shared Key	Entered by the administrator in plain text form and used for authentication during IKE	FLASH(plain text) and RAM (plain)	Zeroization is performed by a single direct overwrite consisting of a pseudo random pattern.

ID	Key Type	Size	Description	Origin	Storage	Zeroization Method
IKE RSA Authentication private Key	RSA - (derived as part of Main Mode exchange)	RSA: 2048 bits	private key used for IKE protocol during the handshake	(derived as part of Main Mode exchange)	RAM (plain text)	Automatically zeroized upon handshake finishing. Overwrite is executed three or more times using a different alternating data pattern each time upon the transfer of the key/critical cryptographic security parameter to another location
IKE Diffie-Hellman Key Pairs	RSA (AES) / DH	2048 bits	Key agreement for IKE (for DH Group 14)	N/A (entered through Windows operating system)	RAM (plain text)	Automatically zeroized upon handshake finishing. Overwrite is executed three or more times using a different alternating data pattern each time upon the transfer of the key/critical cryptographic security parameter to another location
IKE Encryption Key	AES	128 bits, 256 bits,	Used for encrypting IKE negotiations	N/A (entered through Windows operating system)	RAM (plain text)	Zeroized upon deleting the IKE session. Overwrite is executed three or more times using a different alternating data pattern each time upon the transfer of the key/critical cryptographic security parameter to another location

ID	Key Type	Size	Description	Origin	Storage	Zeroization Method
IPsec authentication keys	HMAC AES	160 bits AES: 128, 256 bits	Used for authenticating the IPsec traffic	Generated using IKE protocol (CTR_DRBG (AES)+HMAC-SHA1 AES	RAM (plain text)	Zeroized upon deleting the IPsec session. Overwrite is executed three or more times using a different alternating data pattern each time upon the transfer of the key/critical cryptographic security parameter to another location
IPsec encryption keys	AES-CBC	128 bits 256 bits	Used for encrypting the IPsec traffic		RAM (plain text)	Zeroized upon deleting the IPsec session. Overwrite is executed three or more times using a different alternating data pattern each time upon the transfer of the key/critical cryptographic security parameter to another location
Other Keys / Critical Security Parameters						
Power-up integrity test key	HMAC-SHA1	160 bits	Used to verify integrity of cryptographic module image on power up	Preplaced in module by LogRhythm	Obscured in volatile memory	Re‐initialize Module

Table 6 Critical Security Parameters

Table 6 identifies the Critical Security Parameters for the TOE. The key type and size are identified. The table identifies how the key is created, where it is stored and the method used to destroy the key when it is no longer needed.

The TOE includes an implementation of IPsec in accordance with RFC 4301. The TOE's implementation supports connections using transport mode. The TOE implements the Encapsulating Security Payload (ESP) as defined by RFC 4303 and supports AES-CBC-128 and AES-CBC-256 (as specified by RFC 3602) for data confidentiality, along with HMAC-SHA-1 for authenticating the IPsec traffic. The TOE uses SHA-1 (digest sizes 160 bits) for hash functions in both TLS and IPsec. The TOE implements IKEv1 as defined in RFCs 2407, 2408, 2409, and RFC 4109, supporting AES-CBC-128 and AES-CBC-256 for data confidentiality as specified in RFC 6379.

IKEv1 SA lifetime and volume limits are configured by an authorized administrator and can be limited by the number of packets or number of bytes. The IKEv1 SA lifetime can be configured for length of time, where the time values can be limited to: 24 hours for Phase 1 SAs and 8 hours for Phase 2 SAs.

The IKEv1 protocols implemented by the TOE include the DH Group 14 2 (2048-bit MODP) and utilize RSA (aka rDSA) peer authentication. The TOE can support other DH Groups. The selection of the DH Groups is provided through the Windows Server 2008 R2 Firewall configuration using the following steps:

1. Open the Start menu and search for "Windows Firewall with Advanced Security". Open it.
2. In the left pane of Windows Firewall, right click on "Windows Firewall with Advanced Security". Select "Properties" and open the IPsec Settings tab. In the "IPsec defaults" section, click on "Customize…"
3. In the "Key exchange (Main Mode)" section, select "Advanced" and click on "Customize…"
4. Click "Add"
5. Drop down menus are provided to select the following:
    a. Integrity Algorithm
    b. Encryption Algorithm
    c. Key Exchange Algorithm
        i. From this drop down menu select "Diffie-Hellman Group 14"

The TOE can be configured to use pre-shared keys with a given peer. When a pre-shared key is configured, the IPsec SA will be established using the configured pre-shared key, provided that the peer also has the pre-shared key. The TOE conditions the text-based pre-shared keys by using the bit values of the ASCII characters directly as the key. The pre-shared keys used for IPsec can be constructed of essentially any alphabetic character (upper and lower case), numerals, and special characters (e.g., "!", "@", "#", "$", "%", "^", "&", "*", "(", and ")") and can be anywhere from 1 to 100 characters in length (e.g., 22 characters).

The TOE does not implement the IKEv1 aggressive mode option during a Phase 1 key exchange.

The TOE IPsec implementation includes a security policy database (SPD), which states how to process network packets. The SPD uses the traffic source, destination and transport protocol to determine if a packet should be transmitted or received, blocked, or protected with IPsec, based on firewall and IPsec processing rules. The rules can be created for either inbound traffic or outbound traffic.

The network packet flow rules are defined below:

1. Block connection. BLOCK corresponds to the [RFC4301] DISCARD action. In Windows, BLOCK is considered a firewall policy, rather than an IPsec policy. These rules block all matching inbound network traffic.
2. Allow connection. ALLOW corresponds to the [RFC4301] PROTECT action. These rules allow matching inbound network traffic. Because the default behavior is to block unsolicited inbound network traffic, you must create an allow rule to support any network program or service that must be able to accept inbound connections. The Allow connection rule includes the ability to define how and whether packets should be protected using IPsec.
3. Bypass connection. BYPASS corresponds to the [RFC4301] BYPASS action. These rules allow traffic to transit the IPsec boundary without cryptographic protection.
4. The default behavior is to block unsolicited inbound network traffic, but to allow all outbound network traffic.

As soon as a network packet matches a rule, that rule is applied, and processing stops. In order to prevent unsolicited inbound traffic, an authorized administrator does not need to define a final catch-all rule which will discard a network packet when no other rules in the SPD apply because the TOE will discard the packet.

The Cryptographic support function is designed to satisfy the following security functional requirements:

FCS_CKM.1: See **Table 5 Cryptographic Functions**

- above.

FCS_CKM_EXT.4: See **Table 6 Critical Security Parameters**

- above.

FCS_COP.1(1): See **Table 5 Cryptographic Functions**

- above.

FCS_COP.1(2): See **Table 5 Cryptographic Functions**

- above.

FCS_COP.1(3): See **Table 5 Cryptographic Functions**

- above.

FCS_COP.1(4): See **Table 5 Cryptographic Functions**

- above.
- FCS_IPSEC_EXT.1: The TOE implements IPsec in accordance with RFC 4301. The TOE implements the Encapsulating Security Payload (ESP) as defined by RFC 4303 for the protection of the communication channel between the TOE and the AD server and the remote audit sources which protects from disclosure and modification.
- FCS_HTTPS_EXT.1: The TOE supports HTTPS web-based secure administrator sessions.
- FCS_RBG_EXT.1: See table above.
- FCS_TLS_EXT.1: All communication between the TOE components are secured using TLS 1.0. TLS 1.0 is also used between the TOE and the external IT entities that it communicates with via TLS which are the Knowledge Base Server and audit sources. The TOE protects the interactive communication with administrators using HTTPS over TLS 1.2.

6.3 User data protection

The TOE does not have any capability to process and forward network packets. There are no packets that flow "through the TOE", therefore the TOE does not handle any resources that would be subject to the FDP_RIP.2 requirement. This requirement is implicitly met.

The User data protection function is designed to satisfy the following security functional requirements:

- FDP_RIP.2: The TOE ensures that any previous content of a resource is made unavailable to all objects.

6.4 Identification and authentication

The TOE is designed to require users to be identified and authenticated before they can access any of the TOE functions. The TOE offers both a locally connected console and a network accessible interface (HTTPS) for interactive administrator sessions. The TOE provides Windows interfaces; a thick client (LogRhythm Client Console); and a web-based Web Console as mechanisms to access its security management functions. All communications are encrypted using TLS or TLS/HTTPS.

The TOE supports the local (i.e., on device) definition of administrators with usernames and passwords. Additionally, the TOE can be configured to use Active Directory to support, for example, centralized user administration.

When a user logs in to the TOE, Windows Active Directory or the local Windows operating system authenticates the claimed user identity. Windows Active Directory and the local Windows operating system support password credentials for user authentication. Windows protects user authentication data, storing passwords in non-plaintext form. Windows stores the user password in hashed form and is encrypted when not in use using a Rivest's Cipher (RC)4 algorithm and a RC4 system generated key) as identified in the Microsoft Windows Server 2008 R2 security target: https://www.commoncriteriaportal.org/files/epfiles/st_vid10390-st.pdf, ST Section 6.1.4.3.

The communication path to the authentication server is protected from disclosure and modification by implementing IPsec.

The TOE authenticates peers using pre-shared keys. When a pre-shared key is configured, the IPsec SA is established using the configured pre-shared key, provided that the peer also has the pre-shared key. The TOE conditions the text-based pre-shared keys by using the bit values of the ASCII characters directly as the key. The pre-shared keys used for IPsec can be constructed using any alphabetic character (upper and lower case), numerals,

and special characters (e.g., "!", "@", "#", "$", "%", "^", "&", "*", "(", and ")") and can be anywhere from 1 to 100 characters in length (e.g., 22 characters).

The TOE requires users (i.e., administrators) to be successfully identified and authenticated before they can access any security management functions available in the TOE. The only capabilities allowed prior to users authenticating are the display of the warning banner before authentication. The banner is displayed on every login attempt.

When logging in, the TOE will not echo passwords so that passwords are not inadvertently displayed to the user and any other users that might be able to view the login display.

Note also that should a user have their session terminated (e.g., due to inactivity), they are required to successfully authenticate, by reentering their identity and authentication data, in order to establish a new session.

Password requirements can be configured for local user accounts. Passwords can be composed of upper and lower case letters, numbers and special characters, including ["!", "@", "#", "$", "%", "^", "&", "*", "(", ")". The ability to set a minimum password length of 15 characters or greater is available via the underlying Windows operating system that is included as part of the TOE.

The Identification and authentication function is designed to satisfy the following security functional requirements:

- FIA_PMG_EXT.1: The TOE implements a rich set of password composition constraints as described above.
- FIA_PSK_EXT.1: The IPsec transport can be established using the configured pre-shared key (see section 6.2 above).
- FIA_UAU.7: The TOE does not echo passwords as they are entered.
- FIA_UAU_EXT.2: The TOE provides a local password-based authentication mechanism and can be configured to use external Active Directory authentication servers.
- FIA_UIA_EXT.1: The TOE only displays the warning banner prior to a user being identified and authenticated.

6.5 Security management

The product provides both local and remote administrative access. Local access is via a direct console connection. Remote access is via a thick client (the LogRhythm Client Console) or web-based graphical user interface (the Web Console) as mechanisms to access its security management functions. Security management commands are limited to administrators and are available only after they have provided acceptable user identification and authentication data to the TOE.

LogRhythm users access the LogRhythm administrative interface via the LogRhythm Client Console or Web Console by entering a user ID and password. LogRhythm users log into Windows as Windows Administrators on all the component appliances. On the XM and EM, they log in as a Windows Administrator and then login to the client console via their windows auth. The Windows Administrator role and the Global Administrator role are Security Administrators as defined in the NDPP.

As part of the login, all communications are encrypted. Users can log in with a Windows Active Directory Account or a specific user account as created via the Client Console. FIPS mode must be enabled in the local security policy in the LogRhythm application by the administrator; this restricts user authentication to Windows or Active Directory Authentication only.

Each LogRhythm user account must be is assigned a User Profile. A User Profile is assigned one of the following Security Roles: Global Administrator, Restricted Administrator, Global Analyst, Restricted Analyst, or Web Service Administrator. Only the Global Administrator is considered a 'Security Administrator' as defined in the NDPP. The product provides a means for the administrator to determine the currently running version of the product and to initiate updates to the product. Updates are provided from the LogRhythm Support site at *support.logrhythm.com* in the form of installation packages.

The administrator may configure the TOE to display an informative banner that will appear prior to authentication when accessing the TOE.

The Security management function is designed to satisfy the following security functional requirements:

- FMT_MTD.1: The TOE restricts the access to manage TSF data that can affect the security functions of the TOE to Security Administrators.
- FMT_SMF.1: The TOE includes the functions necessary to remotely and locally manage the cryptomodule and associated functions, and to manage and verify updates of the TOE software and firmware.
- FMT_SMR.2: The TOE includes predefined roles of which only the Global Administrator role has access to all security management functions of the TOE, which corresponds to the required 'Authorized Administrator'.

6.6 Protection of the TSF

The TOE meets FIPS 140-2 requirements and therefore provides self-tests at start-up (which are also on-demand tests available to administrators) on all cryptographic functions. Conditional self-tests are also run during the course of normal operation.

Each LogRhythm appliance performs a (start-up) power-on software integrity self-test to verify the integrity of the component software. If the system fails a software integrity test, it reports status indicating which failure occurred and transitions to an error state. The component does not contain any user data before or during the software integrity test so it is impossible for the component to output user data in this state or a subsequent error state that halts component operation. The component does output its status in the event of a failed software integrity test. Each LogRhythm component verifies the integrity of the software by generating an HMAC-SHA-1 signature for the LogRhythm components' files and comparing the codes against the expected values stored outside the module in HMAC-SHA-1 protected files generated at software build time. Operators can run the software integrity test on demand by stopping and restarting the LogRhythm component.

TLS 1.0 is used to protect communications between all the distributed TOE components. The data is encrypted with the TLS_RSA_WITH_AES_128_CBC_SHA ciphersuite.

Each LogRhythm component runs in a FIPS compliant mode of operation. When communicating components are running in FIPS mode, the communication is encrypted including:

- Site Log Forwarder to the Log Manager Mediator Server
- Log Manager Mediator Server to the EM SQL Server
- Log Manager Mediator Server to the AI Engine Communication Manager
- Client Console to EM SQL Server and the LM SQL Server
- AI Engine to Event Manager SQL Server
- Web Console to the EM SQL Server

Windows provides the system time and requires an NTP Server in the operational environment to ensure time is synched among the distributed components. The TOE uses the time stamps for log accountability and terminating user sessions that have been inactive for an administrator-configured period of time.

TOE updates are available from the LogRhythm secure support site in the form of installation packages along with its digital signature created by LogRhythm using a 2048 bit RSA secret key. LogRhythm will download both a package and its signature via HTTPS and verify the signature using the on-board public key (corresponding to the RSA key used to create the signature). After signature verification, the installation package is used to do a software upgrade. If the verification fails, it is assumed that the download was corrupted and the software upgrade or installation is not performed. The TOE provides functions to query and upgrade the TOE versions.

The underlying Windows operating system protects user authentication data, storing passwords in non-plaintext form and preventing reading of plaintext passwords so that they are not accessible even by an administrator. Windows stores the user password in hashed form and is encrypted when not in use using a Rivest's Cipher (RC)4 algorithm and a RC4 system generated key) as identified in the Microsoft Windows Server 2008 R2 security target: https://www.commoncriteriaportal.org/files/epfiles/st_vid10390-st.pdf, ST Section 6.1.4.3.

The task of protecting (or encrypting) the keys prior to storage in the file system is delegated to the Data Protection API (DPAPI) of Microsoft Windows Server 2008 R2. The DPAPI is a separate component of the operating system that is outside the boundaries of the cryptographic module but relies upon the Windows Server 2008 R2 Enhanced

Cryptographic Provider (RSAENH) for all cryptographic functionality. When a key container is deleted, the file is zeroized before being deleted. RSAENH offloads the key storage operations to the Microsoft Windows Server 2008 R2 operating system, which is outside the cryptographic boundary. Keys are not persistently stored inside the cryptographic module with the exception of the key used for power up self-testing. The TLS private keys are encrypted by the Microsoft Data Protection API (DPAPI) service and stored in the Microsoft Windows Server 2008 R2 file system. Keys are zeroized from memory after use.

The Protection of the TSF function is designed to satisfy the following security functional requirements:

- FPT_APW_EXT.1: The TOE does not offer any functions that will disclose to any user a plain text password.
- FPT_ITT.1: The TOE protects TSF data from disclosure when it is transmitted between separate parts of the TOE through the use of TLS.
- FPT_SKP_EXT.1: The TOE does not offer any functions that will disclose to any users a stored cryptographic key.
- FPT_STM.1: The TOE includes its own hardware clock.
- FPT_TST_EXT.1: The TOE includes a power-on software integrity self-test to verify the integrity of the component software and ensure the correct operation of the cryptographic functions.
- FPT_TUD_EXT.1: The TOE provides functions to query and upgrade the TOE versions. Digital signatures are used to ensure the integrity of each upgrade prior to performing the upgrade.

6.7 TOE access

The TOE is configured by an administrator to display advisory banners prior to allowing an administrator to establish an administrative user session. The banner will be displayed at the Windows login and when accessing the TOE via a direct console connection for local access via a thick client (the LogRhythm Client Console) or the Web Console graphical user interface for remote access.

The TOE provides both local and remote users the ability to logout (or terminate) their sessions as directed by the user.

The TOE is configured to set an interactive session timeout value to terminate an administrative session. Session timeout in the Web Console is defined in minutes. The default value 0 (disabled) is not permitted in the evaluated configuration. The range of valid values for the "minutesBeforeAutomaticLogout" setting are: 0 = disabled, 1 = 1 minute (the minimum timeout you may configure), 2 = 2 minutes up to 9007199254740992 = 9007199254740992 minutes = 150119987579016 hours = 6254999482459 days (the maximum value you can set, as determined by Javascript). These values are is configurable in config\default.json. A remote session that is inactive (i.e., no commands issuing from the remote client) for the defined timeout value will be terminated. A local session that is similarly inactive for the defined timeout period will be terminated. Session timeout in Client Console is defined in seconds with default value equal to 3600 seconds to timeout. The range of values that can be configured is Minimum Value: 1 Seconds to Maximum Value: 2,147,483,647 seconds. These values can be configured in the security.xml file in C:\Program Files\LogRhythm\LogRhythm Console\config. The user will be required to re-enter their user id and their password so they can establish a new session once a session is terminated. If the user id and password match those of the user that was locked, the session is reconnected and normal input/output can again occur for that user.

Note that all of these requirements, banner, session timeout etc. apply and were tested on just the Windows login and session.

The TOE access function is designed to satisfy the following security functional requirements:

- FTA_SSL.3: The TOE terminates remote sessions that have been inactive for an administrator-configured period of time.
- FTA_SSL.4: The TOE provides the function to logout (or terminate) both local and remote user sessions as directed by the user.

- FTA_SSL_EXT.1: The TOE terminates local sessions that have been inactive for an administrator-configured period of time.
- FTA_TAB.1: The TOE can be configured to display administrator-defined advisory banners before establishing an administrative user session.

6.8 Trusted path/channels

To support secure remote administration, an authorized administrator can establish secure remote connections using a web browser with the TOE via TLS/HTTPS. To successfully establish an interactive administrative session, the administrator must be able to provide acceptable user credentials (e.g., user id and password), after which they will be able to access the TOE security functionality. Note that the TOEs Client Console software is installed on one of the appliances (either the EM or XM) and connects to the TOEs EM SQL Server and the LM SQL Server. Therefore these connections are internal TOE and described in Section 6.6.

The TOE is configured to receive external log records over a secure (IPsec or TLS protected) trusted channel. Additionally, an IPsec tunnel in transport mode is used by the TOE to protect the communication with the Active Directory server,) which is used for external authentication with the TOE. This ensures that the credentials passed through the tunnel to authenticate using the external server are not disclosed.

The TOE can initiate communication with the LogRhythm Knowledge Base Server, to download updated processing rules, built-in reports (for compliance), and other processing-related information. The communication channel with the Knowledge Base Server is accessed by the TOE using TLS v1.0.

TLS 1.0 is used between the TOE and the external IT entities that it communicates with via TLS which include the Knowledge Base Server and audit sources. The secure protocols are supported by NIST-validated cryptographic algorithm implementations included in the TOE implementation.

The Trusted path/channels function is designed to satisfy the following security functional requirements:

- FTP_ITC.1: The TOE can be configured to use TLS and IPsec to ensure that any authentication operations are protected from disclosure or modification. Updates from the LogRhythm Knowledge Base Server are protected using TLS for the secure channel. The TOE can be configured to securely accept logs from audit sources received at the SLF using IPsec or TLS.
- FTP_TRP.1: The TOE provides TLS/HTTPS to support secure remote administration. Administrators can initiate a remote session that is secured (from disclosure and modification) using NIST-validated cryptographic operations, and all remote security management functions requires the use of this secure channel.

7. Protection Profile Claims

The ST conforms to the *Protection Profile for Network Devices,* Version 1.1, 8 June 2012 (NDPP), as amended by Errata #3– with the optional IPsec, HTTPS, ITT and TLS requirements.

As explained in Section 3, Security Problem Definition, the Security Problem Definition of the NDPP has been included by reference into this ST.

As explained in Section 4, Security Objectives, the Security Objectives of the NDPP have been included by reference into this ST.

The following table identifies all the Security Functional Requirements (SFRs) in this ST. Each SFR is reproduced from the NDPP and operations completed as appropriate.

Requirement Class	Requirement Component	Source
FAU: Security audit	FAU_GEN.1: Audit Data Generation	NDPP
	FAU_GEN.2: User identity association	NDPP
	FAU_STG_EXT.1: External Audit Trail Storage	NDPP
FCS: Cryptographic support	FCS_CKM.1: Cryptographic Key Generation (for asymmetric keys)	NDPP
	FCS_CKM_EXT.4: Cryptographic Key Zeroization	NDPP
	FCS_COP.1(1): Cryptographic Operation (for data encryption/decryption)	NDPP
	FCS_COP.1(2): Cryptographic Operation (for cryptographic signature)	NDPP
	FCS_COP.1(3): Cryptographic Operation (for cryptographic hashing)	NDPP
	FCS_COP.1(4): Cryptographic Operation (for keyed-hash message authentication)	NDPP
	FCS_IPSEC_EXT.1: Explicit: IPSEC	NDPP
	FCS_HTTPS_EXT.1: Extended: HTTP Security (HTTPS)	NDPP
	FCS_RBG_EXT.1: Extended: Cryptographic Operation (Random Bit Generation)	NDPP
	FCS_TLS_EXT.1: Extended: Transport Layer Security (TLS)	NDPP
FDP: User data protection	FDP_RIP.2: Full Residual Information Protection	NDPP
FIA: Identification and authentication	FIA_PMG_EXT.1: Password Management	NDPP
	FIA_PSK_EXT.1: Extended: Pre-Shared Key Composition	NDPP
	FIA_UAU.7: Protected Authentication Feedback	NDPP
	FIA_UAU_EXT.2: Extended: Password-based Authentication Mechanism	NDPP
	FIA_UIA_EXT.1: User Identification and Authentication	NDPP
	FMT_MTD.1: Management of TSF Data (for general TSF data)	NDPP
	FMT_SMF.1: Specification of Management Functions	NDPP
	FMT_SMR.2: Restrictions on Security Roles	NDPP
FPT: Protection of the TSF	FPT_SKP_EXT.1: Extended: Protection of TSF Data (for reading of all symmetric keys)	NDPP
	FPT_APW_EXT.1: Extended: Protection of Administrator Passwords	NDPP
	FPT_ITT.1: Basic Internal TSF Data Transfer Protection	NDPP
	FPT_STM.1: Reliable Time Stamps	NDPP
	FPT_TST_EXT.1: TSF Testing	NDPP
	FPT_TUD_EXT.1: Extended: Trusted Update	NDPP
FTA: TOE access	FTA_SSL.3: TSF-initiated Termination	NDPP
	FTA_SSL.4: User-initiated Termination	NDPP
	FTA_SSL_EXT.1: TSF-initiated Session Locking	NDPP
	FTA_TAB.1: Default TOE Access Banners	NDPP
FTP: Trusted path/channels	FTP_ITC.1: Trusted Channel	NDPP
	FTP_TRP.1: Trusted Path	NDPP

Table 7 SFR Protection Profile Sources

8. Rationale

This security target includes by reference the NDPP Security Problem Definition, Security Objectives, and Security Assurance Requirements. The security target makes no additions to the NDPP assumptions. NDPP security functional requirements have been reproduced with the Protection Profile operations completed. Operations on the security requirements follow NDPP application notes and assurance activities. Consequently, NDPP rationale applies but is incomplete. The TOE Summary Specification rationale below serves to complete the rationale required for the security target.

8.1 TOE Summary Specification Rationale

Each subsection in Section 6, the TOE Summary Specification, describes a security function of the TOE. Each description is followed with rationale that indicates which requirements are satisfied by aspects of the corresponding security function. The security functions work together to satisfy all of the security functional requirements. Furthermore, all of the security functions are necessary in order for the TSF to provide the required security functionality.

This Section in conjunction with Section 6, the TOE Summary Specification, provides evidence functions are suitable to meet the TOE security requirements. The collection of security functions provide all of the security requirements. The security functions described in the TOE summary necessary for the required security functionality in the TSF. **Table 8 Security Functions vs. Requirements Mapping**

demonstrates the relationship between security requirements and security functions.

	Security audit	Cryptographic support	User data protection	Identification and authentication	Security management	Protection of the TSF	TOE access	Trusted path/channels
FAU_GEN.1	X							
FAU_GEN.2	X							
FAU_STG_EXT.1	X							
FCS_CKM.1		X						
FCS_CKM_EXT.4		X						
FCS_COP.1(1)		X						
FCS_COP.1(2)		X						
FCS_COP.1(3)		X						
FCS_COP.1(4)		X						
FCS_IPSEC_EXT.1		X						
FCS_HTTPS_EXT.1		X						
FCS_RBG_EXT.1		X						
FCS_TLS_EXT.1		X						
FDP_RIP.2			X					
FIA_PMG_EXT.1				X				
FIA_PSK_EXT.1				X				
FIA_UAU.7				X				
FIA_UAU_EXT.2				X				
FIA_UIA_EXT.1				X				
FMT_MTD.1					X			
FMT_SMF.1					X			
FMT_SMR.2					X			
FPT_APW_EXT.1						X		
FPT_ITT.1						X		
FPT_SKP_EXT.1						X		
FPT_STM.1						X		
FPT_TST_EXT.1						X		
FPT_TUD_EXT.1						X		
FTA_SSL.3							X	
FTA_SSL.4							X	
FTA_SSL_EXT.1							X	
FTA_TAB.1							X	
FTP_ITC.1								X
FTP_TRP.1								X

Table 8 Security Functions vs. Requirements Mapping