



Symantec™ Data Center Security: Server Advanced v6.10

Security Target

Evaluation Assurance Level (EAL): EAL2+

Document Version: 0.8

Copyright © 2025 Broadcom. All Rights Reserved. The term “Broadcom” refers to Broadcom Inc. and/or its subsidiaries. For more information, go to www.broadcom.com. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

Broadcom reserves the right to make changes without further notice to any products or data herein to improve reliability, function, or design. Information furnished by Broadcom is believed to be accurate and reliable. However, Broadcom does not assume any liability arising out of the application or use of this information, nor the application or use of any product or circuit described herein, neither does it convey any license under its patent rights nor the rights of others.

Table of Contents

TABLE OF CONTENTS	1
1. INTRODUCTION.....	3
1.1. SECURITY TARGET, TOE AND CC IDENTIFICATION	3
1.2. CONFORMANCE CLAIMS.....	3
1.3. CONVENTIONS	3
1.4. GLOSSARY	4
1.5. ABBREVIATIONS AND ACRONYMS.....	5
2. TOE OVERVIEW	6
2.1 TOE TYPE	6
2.2 TOE COMPONENTS	6
2.3 TOE DESCRIPTION	8
2.4 PHYSICAL BOUNDARIES.....	11
2.5 LOGICAL BOUNDARIES	13
2.6 CAPABILITIES PROVIDED BY THE OPERATIONAL ENVIRONMENT	14
2.7 TOE DOCUMENTATION	14
3. SECURITY PROBLEM DEFINITION	15
3.1 ASSUMPTIONS	15
3.2 THREATS.....	15
4. SECURITY OBJECTIVES	16
4.1 SECURITY OBJECTIVES FOR THE TOE.....	16
4.2 SECURITY OBJECTIVES FOR THE OPERATIONAL ENVIRONMENT	16
5. IT SECURITY REQUIREMENTS.....	18
5.1 EXTENDED COMPONENTS DEFINITION	18
5.2 TOE SECURITY FUNCTIONAL REQUIREMENTS	21
5.3 TOE SECURITY ASSURANCE REQUIREMENTS	26
6. TOE SUMMARY SPECIFICATION.....	35
6.1 SECURITY AUDIT.....	35
6.2 IDENTIFICATION AND AUTHENTICATION.....	36
6.3 SECURITY MANAGEMENT	37
6.4 PROTECTION OF THE TSF	38
6.5 TOE ACCESS.....	39
6.6 TRUSTED PATH/CHANNELS.....	39
6.7 INTRUSION PREVENTION AND DETECTION.....	39
7. RATIONALE	46
7.1 SECURITY OBJECTIVES RATIONALE	46
7.2 SECURITY FUNCTIONAL REQUIREMENTS RATIONALE	49
7.3 SECURITY ASSURANCE REQUIREMENTS RATIONALE	52
7.4 REQUIREMENT DEPENDENCY RATIONALE	52
7.5 TOE SUMMARY SPECIFICATION RATIONALE.....	53

List of Tables

Table 1 – Glossary.....	5
Table 2 – Minimum Hardware Requirements	13
Table 3 – Assumptions	15
Table 4 – Threats.....	15
Table 5 – Security Objectives.....	16
Table 6 – Security Functional Components.....	21
Table 7 – TOE Security Assurance Components	27
Table 8 – Security Problem Definition to Security Objective Correspondence	46
Table 9 – Objectives to Requirement Correspondence	49
Table 10 – Requirement Dependencies	53
Table 11 – Security Functions vs. Requirements Mapping	54

List of Figures

Figure 1 – TOE Components and Interactions	8
--	---

1. Introduction

This section introduces the Target of Evaluation (TOE) and provides the Security Target (ST) and TOE identification, ST and TOE conformance claims, ST conventions, glossary and list of abbreviations.

The ST contains the following additional sections:

- TOE Description (Section 2)—provides an overview of the TOE and describes the physical and logical boundaries of the TOE
- Security Problem Definition (Section 3)—describes the threats and assumptions that define the security problem to be addressed by the TOE and its environment
- Security Objectives (Section 4)—describes the security objectives for the TOE and its operational environment necessary to counter the threats and satisfy the assumptions that define the security problem
- IT Security Requirements (Section 5)—specifies the security functional requirements (SFRs) and security assurance requirements (SARs) to be met by the TOE
- TOE Summary Specification (Section 6)—describes the security functions of the TOE and how they satisfy the SFRs
- Rationale (Section 7)—provides mappings and rationale for the security problem definition, security objectives, security requirements, and security functions to justify their completeness, consistency, and suitability.

1.1. Security Target, TOE and CC Identification

ST Title – Symantec™ Data Center Security: Server Advanced v6.10 Security Target

ST Version – Version 0.8

ST Date – 8 May 2026

TOE Identification – Symantec™ Data Center Security: Server Advanced v6.10

TOE Developer – Symantec Corporation

Evaluation Sponsor – Symantec Corporation

CC Identification – Common Methodology for Information Technology Security Evaluation, CEM:2022, Revision 1, November 2022

1.2. Conformance Claims

This ST and the TOE it describes are conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Components, CC:2022 Revision 1, November 2022.
 - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security Assurance Components, CC:2022 Revision 1, November 2022.
 - Part 3 Conformant

This ST and the TOE it describes are conformant to the following package:

- EAL2 Augmented (ALC_FLR.1).

1.3. Conventions

The following conventions are used in this document:

- Security Functional Requirements—Part 1 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 this ST, iteration is identified with a number in parentheses following the base component identifier. For example, iterations of FCS_COP.1 are identified in a manner similar to FCS_COP.1(1) (for the component) and FCS_COP.1.1(1) (for the elements).
 - Assignment—allows the specification of an identified parameter. Assignments are indicated using bold text and are enclosed 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 enclosed 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 ...”).
- Other sections of the ST—other sections of the ST use bolding to highlight text of special interest, such as captions.

1.4. Glossary

This ST uses a number of terms that have a specific meaning within the context of the ST and the TOE. This glossary provides a list of those terms and how they are to be understood within this ST.

Term	Definition
agent	A TOE software component that an administrator installs on a computer (asset) to be protected.
alert	A notification generated by the TOE when it detects events matching configured filters within configured thresholds.
asset	A computer on which a TOE agent is installed and which is protected by the policies enforced by the installed agent.
authorized users	A user of the system with the System Administrator, Manager, or View role.
configurations	Sets of parameters that specify how agents operate.
event	A record of security-sensitive activity occurring on an asset protected by a TOE agent.
IDS	Intrusion Detection System—a device or software application that monitors a network or systems for malicious activity or policy violations.
IDS data	Refers to raw data (i.e., events) collected by the TOE from resources it is monitoring, based on configured detection rules.
IPS	Intrusion Prevention System—a device or software application that monitors network or system activities for malicious activity, logs information about this activity, reports it and attempts to block or stop it.
policy	A set of rules enforced by an agent to provide protection to an asset. The TOE supports prevention policies and detection policies.
security group	A logical grouping of assets that enables an administrator to apply configurations and policies to a set of assets or agents.
WTMP	wtmp and btmp and variants such as wtmpx and btmpx are files on Unix-like systems that keep track of all logins and logouts to the system.

Table 1 – Glossary

1.5. Abbreviations and Acronyms

The following abbreviations and acronyms are used throughout this ST:

API	Application Programming Interface
CC	Common Criteria
EAL	Evaluation Assurance Level
HTTPS	Hypertext Transfer Protocol Secure
IDS	Intrusion Detection System
IT	Information Technology
JDBC	Java Database Connectivity
JSON	JavaScript Object Notation
ODBC	Open Database Connectivity
SAR	Security Assurance Requirement
SDDC	Software-Defined Data Center
SFR	Security Functional Requirement
SIEM	Security Information and Event Management
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
ST	Security Target
TCP/IP	Transmission Control Protocol/Internet Protocol—communications protocols used on the Internet and similar computer networks.
TOE	Target of Evaluation
TLS	Transport Layer Security
TSF	TOE Security Function
UDP	User Datagram Protocol
USB	Universal Serial Bus

2. TOE Overview

The TOE is Data Center Security: Server Advanced v6.10. It's primary purpose is to provide Intrusion Prevention and Detection services to monitor, control, detect, and respond to endpoint and network threats. The TOE provides these service by monitoring and protecting physical and virtual network endpoints and data centers using a combination of host-based intrusion detection, intrusion prevention, least privilege access control. Additionally, the TOE provides the following features:

- **Security Audit** – The TOE generates and securely stores audit records of security-relevant events
- **Identification & Authentication** – The TOE manages authorized user accounts
- **Security Management** – Those responsible for administering the TOE use the management console to control security functions with role-based access restrictions
- **Protection of the TSF** – The TOE uses HTTPS to protect TSF data communicated between distributed components of the TOE.
- **TOE Access** – The TOE ends inactive interactive sessions after a configurable period
- **Trusted Path/Channels** – HTTPS-based trusted paths and channels for authentication and management.
- **Intrusion prevention and detection** - The TOE provides a policy-based approach to intrusion prevention and intrusion detection

2.1 TOE Type

The TOE Type is a Host-Based Intrusion Prevention System and also a Server Security Management solution. The TOE is software that provides a comprehensive, policy-based security layer for heterogeneous server and endpoint environments. Its primary function is to lock down and monitor servers and endpoints to prevent zero-day exploits and ensure continuous compliance, offering more granular controls than typical desktop endpoint security solutions

2.2 TOE Components

The TOE consists of the following components:

- **Management Server**—the Management Server provides the following capabilities:
 - Secure communications with other TOE components
 - Policy storage and coordination of policy distribution
 - Management of agent event logging and reporting
 - Bulk event file storage management for efficient archival storage of all logged events
 - Alert processing (SMTP, SNMP, file), data purging, and other management functions
 - REST API—a fully instrumented REST API that provides a corresponding API for all actions, enabling full internal and external cloud automation.
- **Communication Server**—The Communication Server provides the following capabilities:
 - Secure communications with Agents and the Management Server
 - Agent Registration and coordination of policy distribution to the Agents
 - Policy and Event processing

Agents report events to the Communication Server for storage and are viewed in the Management Console. TLS versions 1.2 and 1.3 secures communication between the Communication Server and the Agents.

- **Agents**—software components that enforce policy on the endpoint computers on which they are installed. Each agent enforces rules that are expressed in policies, thereby controlling and monitoring application and user behavior on the endpoint. Agents provide the following capabilities:
 - Download of policies and settings from the Communication Server and upload of events and status information to the Communication Server
 - Interception of system calls to enforce prevention policies

- Monitoring of system change events and log files in accordance with detection policies
- Agent configuration and diagnostic support
- Native support for Windows, UNIX and Linux servers and workstations
- Support on VMware guest systems for detection and prevention with any of the operating systems that are natively supported
- Remote monitoring of hosts without a native agent (only detection features are available in this mode)
- Management Console—interface for performing administrative tasks including policy management, configuration management and user management
- Database—the database stores policies, agent information, and real-time actionable events. It is accessible through JDBC/ODBC. The System Administrator can configure encrypted communications between the database and the Management Server and Communication Server.

The Management Server, Communication Server, and Management Console run on Windows operating systems. The agents run on Windows, UNIX and Linux operating systems.

Agents report events to the Communication Server for storage and are viewed in the Management Console. Agent log rules control the events that are logged for that agent. Logged data includes event date and time, event type, importance rating, and any prevention action performed. Dashboards in the Management Console provide charts and graphs displaying aggregated summary data about events, agents, and policies.

The TOE uses Transport Layer Security (TLS) using X.509 certificates with SHA-256 to secure communications between its various components.

The following figure depicts the TOE components and their interactions with each other and with the operational environment.

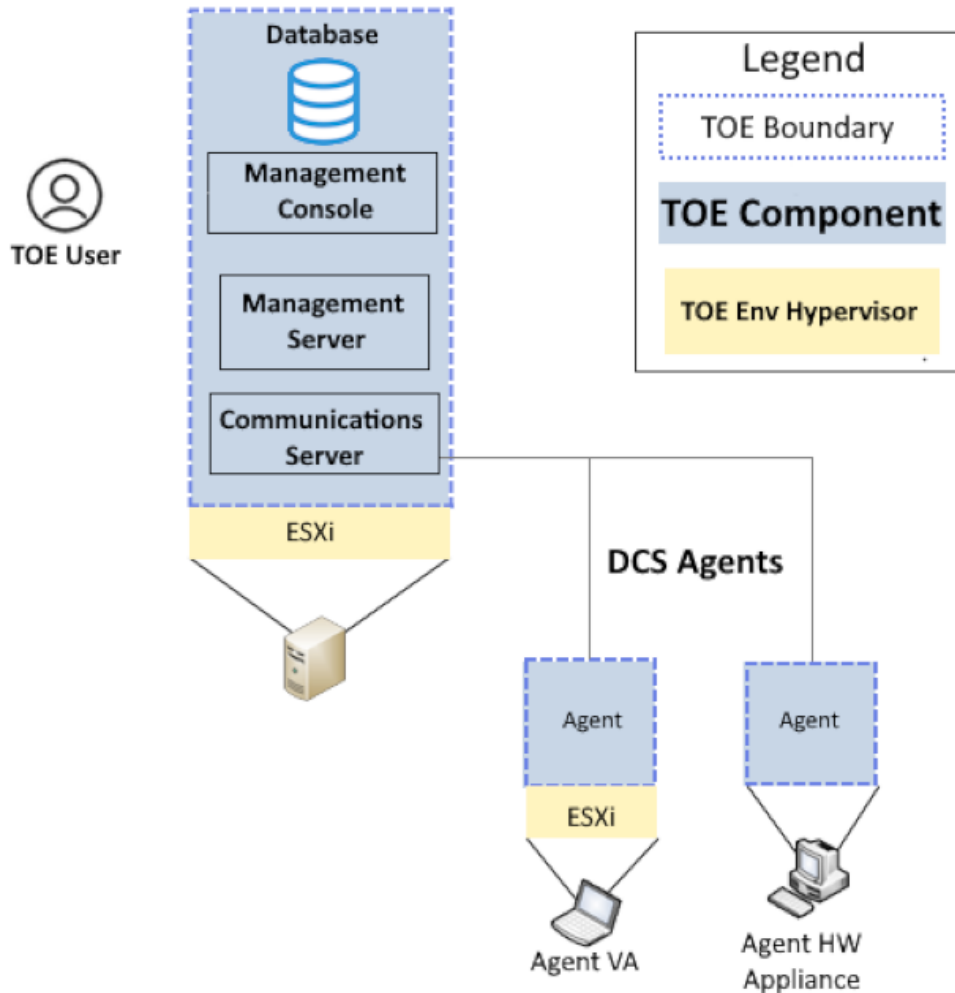


Figure 1 – TOE Components and Interactions

Note that agents continue to monitor and enforce security even if network outages occur between the agents and the server environment. Agents can be configured to operate in a standalone or an unmanaged mode.

The TOE components can be deployed on physical systems and in virtualized environments. A virtualized ecosystem such as the one supported by VMware has many parts. Its parts include management infrastructure, virtual guest machines, and hypervisors that span a variety of operating systems. To protect this heterogeneous environment, the TOE relies on specific policies and enforcement agents that are appropriate to each component to be secured. The components include ESX, ESXi, and vCenter.

2.3 TOE Description

The TOE provides its security functionality through implementation and management of the following objects:

- Assets
- Policies
- Security groups
- Configurations
- Events
- Alerts and notifications.

These objects are described in more detail in the following subsections.

2.3.1 Assets

An **asset** is a computer on which an agent is installed. An **agent** is the software that an administrator installs on a computer to be protected. Agents protect assets by enforcing rules that are expressed in **policies**. TOE System Administrators and Managers can view details of an asset, search for an asset, edit details of an asset, assign a security group (see Section 2.3.3 below) to an asset, and delete an asset. Agents support prevention and detection features. Agents that support prevention features control behavior by allowing and preventing specific actions that an application or user might take. For example, a prevention policy can specify that an email application may not spawn other processes, including dangerous processes such as viruses, worms, and Trojan horses. However, the email application can still read and write to the directories that it needs to access. Agents that support detection features control behavior by detecting suspicious activity and taking action. For example, a detection policy can take action when it detects an attempt by an unauthorized user to gain illegitimate access to a system. No action would be taken for failed attempts that resulted from normal behavior such as an expired password or a user forgetting a password.

2.3.2 Policies

Agents use the following types of policies:

- **Prevention policies**—these confine each process on a computer to its normal behavior. Programs that are identified as critical to system operation are given specific behavior controls, while generic behavior controls provide compatibility for other services and applications.
- **Detection policies**—these monitor events and syslogs, and report anomalous behavior. Features include: policy-based auditing and monitoring; log consolidation for easy search, archival, and retrieval; event analysis and response capabilities; and file and registry protection and monitoring.

Agent policies have options that allow the System Administrator or Manager to configure a policy for assignment to a target computer. Policy options comprise a simplified set of controls that the System Administrator or Manager can use to enable or disable features in a policy. Some options have parameters, which provides even further customization.

2.3.3 Security Groups

A **security group** is a logical grouping of assets that enables a System Administrator or Manager to apply configurations and policies to a set of assets or agents. System Administrators and Managers can create, edit, copy, and delete security groups, apply policy to a security group, and apply a workstation configuration to a security group. The TOE ships with a number of out-of-the-box (OOTB) security groups and categories. A category is a further logical grouping of a set of security groups. System Administrators or Managers can create new categories and assign security groups to the categories. When the last security group that is assigned to a category is deleted, the category is also deleted.

2.3.4 Configurations

Configurations specify how agents operate. The TOE uses **common**, **prevention**, and **detection** configurations.

Common configurations consist of the following groups of communication parameters and event logging parameters:

- Communication parameters control how agents communicate with the Management Server:
 - Polling interval
 - Enable real-time notification
 - Port
 - Connection timeout
- Event logging parameters control how agents log events:
 - Enable log consolidation
 - Enable log rotation
 - Enable bulk log transfer
 - Delete log files after processing
 - Stop and restart logging at disk usage (%)

- Reader and writer limits
- Event Management.

Prevention configurations comprise log rules. The System Administrator or Manager uses log rules to configure the transmission of events that agents send to the Management Server.

Log rules comprise the following:

- Filter rules
- Transmit action.

The System Administrator or Manager uses the log rule editor to specify filter rules and a transmit action.

Detection configurations comprise the following parameters:

- Parameters that control how the detection features of an agent operate. These parameters include the following:
 - File collector
 - Event log collector
 - Audit collector
 - Registry collector
 - Syslog collector
 - WTMP collector
 - BTMP collector
 - C2 collector
- Log rules.

2.3.5 Events

Events are records of informative, notable, and critical activities occurring on the assets protected by TOE agents. An agent logs events to the Management Server based on the agent's log rules. Additionally, operations on the Management Server can generate server-related events.

The TOE defines the following categories of events:

- Prevention—an agent's prevention policy generates prevention events when applications access computer and network resources that violate the policy's behavior control. Prevention events have two sub-groups: Tunable and Benign. System Administrators or Managers can add a strategy to the Tunable events. The Benign events are informational events.
- Detection—an agent's detection policy generates detection events when monitored files or registry keys change, or when system or application logs generate events that match the policy's criteria.
- Management—an agent records management events that are related to the agent's configuration and communication status
- Profile—an agent's prevention policy generates profile events when a process is profiled.
- File Catalog—an agent records file catalog events that acknowledge the following activities:
 - Successful event log rollover
 - Successful storage of log files in the agent repository in the Management Server during bulk log transfer
- Analysis—analysis events comprise the events that were transferred to the Management Server using bulk log transfer and then loaded into the database. Analysis events are of long-term interest, generally for audit or forensic analysis needs
- Audit—the Management Server records audit events whenever changes to the system configuration are made. Optionally, the Management Server can record audit events whenever searches, queries, or reports are executed.

2.3.6 Alerts and Notifications

Alerts are used to send events of interest to the following destinations:

- email messages
- SNMP traps
- text files.

The Alert function polls the database for events that match an alert filter. When a match is found, the Alert function generates and sends email messages, SNMP traps, and text files that are associated with the alert. The TOE supports email aggregation that combines multiple alerts occurring within a specified time interval into a single email message. Email aggregation prevents flooding email addresses with excessive emails or with messages that exceed size limitations.

System Administrators or Managers can: configure alert settings; add, edit, copy, delete, import and export alerts; enable and disable alerts. They can also view the notifications for the alerts that have been created and perform the following actions: view details of notifications; acknowledge a notification so the TOE will not generate further notifications for the alert; mark an acknowledged notification as not acknowledged.

2.4 Physical Boundaries

2.4.1 Physical Scope

2.4.1.1 Hardware

This is a software only TOE. No hardware is included in the TOE Boundary.

2.4.1.2 Firmware

This is a software only TOE. No hardware is included in the TOE Boundary.

2.4.1.3 Guidance

The TOE guidance documentation (AGD) is available as a **.pdfs**, dated February 19, 2026. The TOE guidance can be made available upon request.

2.4.1.4 Software

The TOE consists of two (2) software packages: one that includes Management Console, Management Server, and Communication Server, and one that includes the Agent. The software packages can be downloaded from the Broadcom Software downloads website: <https://support.broadcom.com/group/ecx/downloads>. Customers must have an active entitlement to access the TOE downloads. The files are downloaded as **.zip** files

The TOE is a software product provided in the following form:

- Server
 - Symantec_Data_Center_Security_Server_Advanced_6.10.1_Management_Components.zip
- Agent
 - Symantec_Data_Center_Security_Server_Advanced_6.10.1_Windows_Agent.zip

All TOE software downloads have a corresponding SHA-256 checksum that can be used for integrity verification.

The evaluated configuration of the TOE will have all Server components (Management Console, Management Server, Communication Server) installed on a single Windows 2019 server machine, and the TOE agent residing on a Windows 10 endpoint. The specifications of the 2 machines are as follows:

TOE Component	Operating System (non-TOE)	Hardware Machine Specifications (non-TOE)
---------------	----------------------------	---

Server <ul style="list-style-type: none"> • Management Console • Management Server • Communication Server 	Windows Server 2019 Standard Edition	64-bit (x86_64), four-core CPU, 16 GB RAM, 150 GB hard drive storage
Agent	Windows 10 Enterprise	64-bit (x86_64), two-core CPU, 4 GB RAM, 50 GB hard drive storage

2.4.2 Operational Environment Components

The Management Server (with the Communications Server, Management Console and database) is supported on the following 64-bit Windows operating systems (but not in Server Core configurations):

- Windows Server 2025, 64-bit
- Windows Server 2022, 64-bit
- Windows Server 2019, 64-bit
- Windows Server 2016, 64-bit

For the database, the TOE supports Microsoft SQL Server 2016 with Service Pack 1 or higher. This includes all production editions of SQL Server such as Enterprise and Standard, and all corresponding service packs. SQL Server Express is not supported in a production environment.

The management console is supported on the following Windows platforms (but not in Server Core configurations):

- Windows Server 2025, 64-bit
- Windows Server 2022, 64-bit
- Windows Server 2019, 64-bit
- Windows Server 2016, 64-bit
- Windows 11, 64-bit.

Agents are available for and supported on the following platforms:

- Alma Linux 8.5 to 8.10, x86_64
- Amazon Linux 2023, x86_64
- RHEL 10, x86_64
- RHEL /Rocky Linux /Oracle Linux 9.0 to 9.5, x86_64
- RHEL 9.0 to 9.3, aarch64
- RHEL /Rocky Linux /Oracle Linux 8.0 to 8.10
- RHEL 8.4 to 8.10, aarch64
- SUSE Linux Enterprise Server 15 SP0 to SP6, x86_64
- Ubuntu 24.04 LTS, x86_64
- Ubuntu 22.04 LTS, x86_64
- Ubuntu 20.04 LTS, x86_64
- Windows 2025, x86_64
- Windows 2022, x86_64
- Windows 2019, x86_64
- Windows 2016, x86_64
- Windows 11, x86_64

The following table lists the minimum hardware requirements for the TOE components.

Component	OS and Hardware
Management Server Communication Server Management Console	Windows Server 2019 Standard 100 GB free disk space (all platforms) 8 GB RAM 4 CPUs (or cores)
Agent	1 GB free disk space (all platforms) 256 MB of RAM 4 CPUs (or cores)

Table 2 – Minimum Hardware Requirements

In addition to the hardware and software platforms identified above, the TOE may require the following in its operational environment, depending on configuration:

- SMTP Server—supports e-mail alert notifications
- SNMP Server—supports SNMP trap alert notifications
- Active Directory—supports external user identification and authentication.

2.5 Logical Boundaries

This section summarizes the security functions provided by the TOE.

2.5.1 Security Audit

The TOE is able to generate audit records of security-relevant events, which it stores in the Management Server database. The database protects the stored audit records from unauthorized modification and deletion. The TOE provides System Administrators with capabilities to review the generated audit records, including capabilities for searching audit records based on the values in specified audit record fields and to filter records based on audit event type and period of time.

2.5.2 Identification & Authentication

The TOE maintains accounts of the authorized users of the system. The user account includes the following attributes associated with the user: username; password; roles; and e-mail address information. This information is stored in the Management Server database. The TOE supports user authentication with local-defined passwords and remote authentication using Active Directory. The TOE enforces password complexity requirements that dictate that the password is at least eight characters in length, contains at least one upper case character, contains at least one lower case character, contains at least 1 number, and contains at least 1 special character (! @ # \$ % ^ & - + = () { } _).. The TOE enforces an account lockout mechanism triggered by failed authentication attempts.

2.5.3 Security Management

TOE users with the System Administrators or Managers role manage the TOE and its security functions using the Management Console, which provides access to all the TOE's security management functions. The TOE provides the following default security management roles for the Management Console: System Administrators; Managers; and Viewers (which does not provide any security management capability). The TOE enforces restrictions on which management capabilities are available to each role.

2.5.4 Protection of the TSF

The TOE uses HTTPS to protect TSF data communicated between distributed components of the TOE.

2.5.5 TOE Access

The TOE will terminate interactive sessions after a period of inactivity configurable by a System Administrator. By default, interactive sessions are terminated after 60 minutes of inactivity. The TOE also allows user-initiated termination of the user's own interactive session by explicitly logging off.

2.5.6 Trusted Path/Channels

The TOE provides a trusted path for TOE System Administrator or Managers to communicate with the TOE via its REST API. Users of the TOE with any role initiate the trusted path by establishing an HTTPS connection. The trusted path is used for initial authentication and all subsequent administrative actions. The use of HTTPS ensures all communication over the trusted path is protected from disclosure and modification.

2.5.7 Intrusion Prevention and Detection

The TOE provides a policy-based approach to intrusion prevention and intrusion detection. The TOE is able to control and monitor what programs and users can do to computers. Agent software at the endpoints controls and monitors behavior based on policy. The TOE policy library contains prevention and detection policies that a System Administrator or Manager can deploy and customize to protect the network and endpoints. Agents enforce rules specified in prevention policies to control how processes running on the protected asset access resources, such as other processes, memory, files, registry keys (on Windows-based assets) and network connections. Agents apply rules configured in detection policies to collect IDS data from monitored resources such as Windows event logs, text logs, registry keys, files, syslog daemons and UNIX wtmp files.

In response to identified violations of prevention and detection policies, agents can generate events that are stored in the Management Server database. The database protects the stored events from unauthorized modification and deletion. The TOE can monitor the events stored in the database against configured filter rules and generate alerts if a configured minimum number of events occur in a configured time window. When an alert is generated, a notification can be sent to configured alert destinations, including email addresses, an SNMP server or a text file.

The TOE also provides capabilities for TOE System Administrators, Managers, and Viewers to review generated events, including capabilities for searching events based on the values in specified event fields and to filter events based on event type and period of time.

2.6 Capabilities Provided by the Operational Environment

The TOE relies on the operational environment for the following components and capabilities:

- The underlying operating system of each TOE component is relied on to protect the component and its configuration from unauthorized access.
- The underlying operating system of each TOE component is relied on to provide a reliable date and time stamp for use by the TOE.

2.7 TOE Documentation

This section identifies the guidance documentation included in the TOE. The documentation comprises:

- Symantec™ Data Center Security: Server Advanced v6.10, Common Criteria Guidance Supplement, Evaluation Assurance Level (EAL): EAL2+, Document Version: 0.1

3. Security Problem Definition

This section defines the security problem to be addressed by the TOE, in terms of threats to be countered by the TOE or its operational environment, and assumptions about the intended operational environment of the TOE.

3.1 Assumptions

This section contains assumptions regarding the operational environment and the intended usage of the TOE.

Assumption	Description
A.MANAGE	There will be one or more competent individuals assigned to manage the TOE and the security of the information it contains.
A.PLATFORM	The underlying operating system of each TOE component will protect the component and its configuration from unauthorized access.
A.PROTECT	The hardware hosting the TOE software and the TOE software critical to security policy enforcement <u>will be located within controlled access facilities which will prevent unauthorized physical access.</u>

Table 3 – Assumptions

3.2 Threats

This section identifies and describes the threats to be countered by the TOE and its operational environment.

Threat	Description
T.BRUTE_FORCE	An unauthorized user may gain access to the TOE through repeated password-guessing attempts.
T.INTEGRITY_COMPROMISE	An unauthorized user may attempt to modify or destroy audit or IDS data, thus removing evidence of unauthorized or malicious activity.
T.INTRUSION_ATTEMPT	An unauthorized user or process may attempt to perform actions on a host system that could compromise the security of the host system or its resources, or make improper use of system resources.
T.NETWORK_COMPROMISE	TSF data communicated between components of the TOE, or between the TOE and external entities, is disclosed or modified.
T.NO_ACCOUNTABILITY	Authorized users of the TOE perform adverse actions on the TOE, or attempt to perform unauthorized actions, which go undetected.
T.NO_ACCOUNTABILITY	Authorized users of the TOE perform adverse actions on the TOE, or attempt to perform unauthorized actions, which go undetected.
T.UNATTENDED_SESSION	An unauthorized user gains access to the TOE via an unattended authorized user session.
T.UNAUTHORIZED_ACCESS	An unauthorized user may gain access to the TOE security functions and data.
T.UNAUTHORIZED_ACTIVITY	Authorized users perform unauthorized actions on the TOE.
T.UNDETECTED_THREATS	Events generated by entities in the IT system indicative of misuse or unauthorized or malicious activity go undetected

Table 4 – Threats

4. Security Objectives

This section identifies the security objectives for the TOE and its operational environment. The security objectives identify the responsibilities of the TOE and its environment in addressing the security problem defined in Section 3.

4.1 Security Objectives for the TOE

The following are the TOE security objectives:

Security Objective	Description
O.AUDIT	The TOE shall be able to generate audit records of security-relevant events.
O.AUDIT_REVIEW	The TOE shall provide a means for the System Administrator to review the audit records generated by the TOE.
O.I_AND_A	The TOE shall provide a means for users to be identified and authenticated before gaining access to TOE services.
O.INTRUSION	The TOE shall provide capabilities to prevent and detect intrusion attempts on monitored assets, based on configured prevention and detection policies.
O.PASSWORD_CONTROLS	The TOE must ensure that passwords used to protect TOE access are sufficiently strong, and that a limit is placed on failed authentication attempts.
O.PROTECTED_COMMS	The TOE shall protect communications between distributed parts of the TOE, and between the TOE and external entities, from disclosure and modification.
O.RESPONSE	The TOE shall respond to misuse and unauthorized or malicious activity it identifies based on its configuration.
O.REVIEW	The TOE shall provide capabilities for effective review of stored IDS data.
O.SECURITY_MANAGEMENT	The TOE shall restrict the ability to perform security management functions on the TOE to authorized users having appropriate privileges.
O.SESSION_TERMINATION	The TOE shall provide mechanisms to terminate a user session after a period of inactivity or at the request of the user.
O.STORAGE	The TOE shall protect stored audit records and IDS data from unauthorized modification or deletion.

Table 5 – Security Objectives

4.2 Security Objectives for the Operational Environment

The following are the security objectives for the operational environment of the TOE.

Security Objective for the OE	Description
OE.PERSONNEL	Those responsible for the TOE must ensure that personnel working as authorized users have been carefully selected and trained for proper operation of the TOE.
OE.PHYSICAL	Those responsible for the TOE must ensure that those parts of the TOE critical to security policy are protected from any physical attack.
OE.PLATFORM	The underlying operating system of each TOE component will protect the component and its configuration from unauthorized access.
OE.TIME	The underlying operating system of the TOE provides a reliable time source for use by the TOE.

5. IT Security Requirements

5.1 Extended Components Definition

5.1.1 Intrusion Prevention and Detection (IPD)

This ST defines a new functional class for use within this ST: Intrusion Prevention and Detection (IPD). This family of requirements was created to specifically address the capabilities of intrusion prevention and intrusion detection solutions. The user data protection family (FDP) of the CC was used as a model for creating requirements for intrusion prevention capabilities, while the audit family (FAU) was used as a model for creating requirements for intrusion detection capabilities. Intrusion prevention depends on the ability of a TSF to identify anomalous behaviour exhibited by maleficent software and to prevent damage to the protected resource (e.g., an endpoint computer, operating system, or application) by blocking the potential attack. Intrusion detection depends on the ability of a TSF to detect activities on a protected resource that indicate the potential to inappropriately affect the protected resource.

5.1.1.1 IDS Data Collection (IPD_IDC)

This family defines requirements for a capability to collect IDS data from a monitored asset based on an administrator-configurable policy.

Management: IPD_IDC.1

The following actions could be considered for the management functions in FMT:

- a) management of detection policies.

Audit: IPD_IDC.1

There are no auditable events foreseen.

IPD_IDC.1 – IDS data collection

Hierarchical to: No other components.

Dependencies: None

IPD_IDC.1.1 The TSF shall be able to collect IDS data from the following monitored resources, based on configured detection rules: [**selection: *Windows event logs, text logs, registry keys, files, syslog daemon, C2 audit log, WTMP file, [assignment: other specifically defined resources]***].

5.1.1.2 IPD Event Review (IPD_IER)

This family defines requirements for reviewing event data.

Management: IPD_IER.1

The following actions could be considered for the management functions in FMT:

- a) maintenance of the group of users with read access rights to the event data.

Management: IPD_IER.2

There are no management actions foreseen.

Audit: IPD_IER.1, IPD_IER.2

There are no auditable events foreseen.

IPD_IER.1 – Controlled event review

Hierarchical to: No other components.

Dependencies: IPD_RCT.1

- IPD_IER.1.1** The TSF shall provide [**assignment: *authorized users***] with the capability to read [**assignment: *list of event data***] from the generated events.
- IPD_IER.1.2** The TSF shall provide the event data in a manner suitable for the user to interpret the information.
- IPD_IER.1.3** The TSF shall prohibit all users read access to the event data, except those users that have been granted explicit read access.

IPD_IER.2 – Selectable event review

Hierarchical to: No other components.

Dependencies: IPD_IER.1

- IPD_IER.2.1** The TSF shall provide the ability to apply [**assignment: *methods of selection and/or ordering***] of event data based on [**assignment: *criteria with logical relations***].

5.1.1.3 Policy-Based Prevention (IPD_PBP)

This family defines requirements for a capability to prevent attempts to damage a protected asset based on an administrator-configurable policy.

Management: IPD_PBP.1

The following actions could be considered for the management functions in FMT:

- a) management of prevention policies.

Audit: IPD_PBP.1

There are no auditable events foreseen.

IPD_PBP.1 – Simple policy-based prevention

Hierarchical to: No other components.

Dependencies: None

- IPD_PBP.1.1** The TSF shall enforce an intrusion prevention policy to assets based on the following: [**assignment: *list of subjects and resources controlled under the intrusion prevention policy, and for each, the relevant security attributes, or named groups of relevant security attributes***].
- IPD_PBP.1.2** The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled resources is allowed: [**assignment: *rules governing access among controlled subjects and controlled resources using controlled operations on controlled resources***].
- IPD_PBP.1.3** The TSF shall explicitly authorize access of subjects to resources based on the following additional rules: [**assignment: *rules based on security attributes that explicitly authorize access of subjects to resources***].
- IPD_PBP.1.4** The TSF shall explicitly deny access of subjects to resources based on the following additional rules: [**assignment: *rules based on security attributes that explicitly deny access of subjects to resources***].

5.1.1.4 Intrusion Reaction (IPD_RCT)

This family defines requirements for capabilities to generate events in response to triggered intrusion prevention or intrusion detection rules and to respond to events matching specified criteria.

Management: IPD_RCT.1

There are no management actions foreseen.

Management: IPD_RCT.2

The following actions could be considered for the management functions in FMT:

- b) maintenance of the rules that control alert generation.

Audit: IPD_RCT.1, IPD_RCT.2
There are no auditable events foreseen.

IPD_RCT.1 – Event generation

Hierarchical to: No other components.

Dependencies: IPD_PBP.1 or IPD_IDC.1 or IPD_SBD.1

IPD_RCT.1.1 The TSF shall be able to generate an event in response to identified violations of [**selection: *prevention, detection***] policies.

IPD_RCT.1.2 The TSF shall record within each event at least the following information: source of the event; date and time the event occurred; type of event; severity of event; description of event.

IPD_RCT.2 – Alert definition and reaction

Hierarchical to: No other components.

Dependencies: IDS_RCT.1

IPD_RCT.2.1 The TSF shall be able to trigger an alert when [**assignment: *set of conditions***] are met.

IPD_RCT.2.2 The TSF shall send a notification to [**assignment: *alert destination***] when an alert is triggered.

5.1.1.5 Signature-based Detection (IPD_SBD)

This family defines requirements for capabilities to detect threats and intrusions in network traffic and on network endpoints using signatures.

Management: IPD_SBD.1

There are no management actions foreseen.

Audit: IPD_SBD.1

There are no auditable events foreseen.

IPD_SBD.1 – Signature-based detection

Hierarchical to: No other components.

Dependencies: None

IPD_SBD.1.1 The TSF shall be able to detect malware using signatures on Linux Systems.

5.1.1.6 IPD Event Storage (IPD_STG)

This family defines requirements for securely storing event data.

Management: IPD_STG.1

There are no management actions foreseen.

Audit: IPD_STG.1

There are no auditable events foreseen.

IPD_STG.1 – Protected event storage

Hierarchical to: No other components.

Dependencies: IPD_RCT.1

IPD_STG.1.1 The TSF shall protect the stored event data from unauthorized deletion.

IPD_STG.1.2 The TSF shall be able to [**selection, choose one of: *prevent, detect***] unauthorized modifications to stored event data.

5.2 TOE Security Functional Requirements

This section specifies the security functional requirements (SFRs) for the TOE. SFRs were drawn from the Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Components, CC:2022 Revision 1, November 2022, and from the extended components defined in Section 5.1 above.

Requirement Class	Requirement Component
FAU: Security Audit	FAU_GEN.1 – Audit data generation
	FAU_SAR.1 – Audit review
	FAU_SAR.2 – Restricted audit review
	FAU_SAR.3 – Selectable audit review
	FAU_STG.2 – Protected audit data storage
FIA: Identification and Authentication	FIA_AFL.1 – Authentication failure handling
	FIA_ATD.1 – User attribute definition
	FIA_SOS.1 – Verification of secrets
	FIA_UAU.2 – User authentication before any action
	FIA_UAU.5 – Multiple authentication mechanisms
	FIA_UID.2 – User identification before any action
FMT: Security Management	FMT_MOF.1 – Management of security functions behaviour
	FMT_MTD.1(*) – Management of TSF data
	FMT_SMF.1 – Specification of Management Functions
	FMT_SMR.1 – Security roles
FPT: Protection of the TSF	FPT_ITT.1 – Basic internal TSF data transfer protection
FTA: TOE Access	FTA_SSL.3 – TSF-initiated termination
	FTA_SSL.4 – User-initiated termination
FTP: Trusted Path/Channels	FTP_ITC.1 – Inter-TSF trusted channel
	FTP_TRP.1 – Trusted path
IPD: Intrusion Prevention and Detection	IPD_IDC.1 – IDS data collection
	IPD_IER.1 – Controlled event review
	IPD_IER.2 – Selectable event review
	IPD_PBP.1 – Simple policy-based prevention
	IPD_RCT.1 – Event generation
	IPD_RCT.2 – Alert definition and reaction
	IPD_SBD.1 – Signature-based detection
	IPD_STG.1 – Protected event storage

Table 6 – Security Functional Components

5.2.1 Security Audit (FAU)

FAU_GEN.1 – Audit data generation

- FAU_GEN.1.1** The TSF shall be able to generate audit data of the following auditable events:
- a) Start-up and shutdown of the audit functions;
 - b) All auditable events for the [*not specified*] level of audit; and
 - c) [the following auditable events:
 - Reading of information from the audit records
 - All use of the authentication mechanism
 - All use of the user identification mechanism
 - All modifications in the behavior of the functions of the TSF
 - All modifications to the values of TSF data
 - Use of the management functions
 - Modifications to the group of users that are part of a role].
- FAU_GEN.1.2** The TSF shall record within the audit data at least the following information:
- a) Date and time of the auditable event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event;
 - b) For each auditable event type, based on the auditable event definitions of the functional components included in the PP, PP-Module, functional package or ST, [none].

FAU_SAR.1 – Audit review

- FAU_SAR.1.1** The TSF shall provide [System Administrators, Managers, and Viewers] with the capability to read [all audit information] from the audit data.
- FAU_SAR.1.2** The TSF shall provide the audit data in a manner suitable for the user to interpret the information.

FAU_SAR.2 – Restricted audit review

- FAU_SAR.2.1** The TSF shall prohibit all users read access to the audit data, except those users that have been granted explicit read-access.

FAU_SAR.3 – Selectable audit review

- FAU_SAR.3.1** The TSF shall provide the ability to apply [searches and filtering] of audit data based on [the following criteria:
- Searches based on values of specified audit record fields and combinations of logical and conditional operators
 - Filtering based on audit event type and period of time].

FAU_STG.2 – Protected audit data storage

- FAU_STG.2.1** The TSF shall protect the stored audit data in the audit trail from unauthorized deletion.
- FAU_STG.2.2** The TSF shall be able to [*prevent*] unauthorized modifications to the stored audit data in the audit trail.

5.2.2 Identification and Authentication (FIA)

FIA_AFL.1 – Authentication failure handling

- FIA_AFL.1.1** The TSF shall detect when [an System Administrator configurable positive integer within [two-ten]] unsuccessful authentication attempts occur related to [*user logon*].
- FIA_AFL.1.2** When the defined number of unsuccessful authentication attempts has been [*met*], the TSF shall [*lock the user out until the lockout duration passes or the account is reset by a System Administrator*].

FIA_ATD.1 – User attribute definition

- FIA_ATD.1.1** The TSF shall maintain the following list of security attributes belonging to individual users: [
- **User Identity**
 - **Password**
 - **Roles**
 - **E-mail address**].

FIA_SOS.1 – Verification of secrets

- FIA_SOS.1.1** The TSF shall provide a mechanism to verify that secrets meet [the following constraints for all user accounts:
- **Minimum length of eight characters**
 - **Maximum length of twenty characters**
 - **At least one upper case**
 - **At least one lower case**
 - **At least one number**
 - **At least one of the following: ! @ # \$ % ^ & - + = () { } _]**.

FIA_UAU.2 – User authentication before any action

- FIA_UAU.2.1** The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

FIA_UAU.5 – Multiple authentication mechanisms

- FIA_UAU.5.1** The TSF shall provide [local passwords, support for remote authentication using Active Directory] to support user authentication.
- FIA_UAU.5.2** The TSF shall authenticate any user's claimed identity according to the [authentication method configured for the user account, either:
- **local password-based authentication of user identities, or**
 - **remote authentication using Active Directory username and password**].

FIA_UID.2 – User identification before any action

- FIA_UID.2.1** The TSF shall require each user to be successfully identified before allowing any TSF-mediated actions on behalf of that user.

5.2.3 Security Management (FMT)**FMT_MOF.1 – Management of security functions behaviour**

- FMT_MOF.1.1** The TSF shall restrict the ability to [*modify the behavior of*] the functions [security audit] to [System Administrators and Managers].

FMT_MTD.1 – Management of TSF data

- FMT_MTD.1.1(1)** The TSF shall restrict the ability to [*modify, delete*] the [assets] to [System Administrators and Managers].
- FMT_MTD.1.1(2)** The TSF shall restrict the ability to [*modify, delete, create*] the [security groups, alerts and notifications] to [System Administrators and Managers].
- FMT_MTD.1.1(3)** The TSF shall restrict the ability to [*create*] the [policies] to [System Administrators, Managers].
- FMT_MTD.1.1(4)** The TSF shall restrict the ability to [*modify, delete*] the [policies] to [System Administrators, Managers].

- FMT_MTD.1.1(5)** The TSF shall restrict the ability to [*modify, delete, [create]*] the [**configurations**] to [**System Administrators, Managers**].
- FMT_MTD.1.1(6)** The TSF shall restrict the ability to [*modify, delete, [create]*] the [**Management Console user accounts**] to [**System Administrators**].
- FMT_MTD.1.1(7)** The TSF shall restrict the ability to [*modify*] the [**System user accounts**] to [**System Administrators**].
- FMT_MTD.1.1(8)** The TSF shall restrict the ability to [*modify*] the [**password of another user**] to [**System Administrators**].

FMT_SMF.1 – Specification of Management Functions

- FMT_SMF.1.1** The TSF shall be capable of performing the following security management functions: [
- **Manage assets**
 - **Manage security groups**
 - **Manage policies**
 - **Manage configurations**
 - **Manage Management Console user accounts and roles**
 -
 - **Manage alerts and notifications**
 - **View events**
 - **Manage audit function**
 - **Manage users and groups**
 - **Modify user passwords**].

FMT_SMR.1 – Security roles

- FMT_SMR.1.1** The TSF shall maintain the roles: [
- **System Administrators**
 - **Managers**
 - **Viewers**].
- FMT_SMR.1.2** The TSF shall be able to associate users with roles.

5.2.4 Protection of the TSF (FPT)

FPT_ITT.1 – Basic internal TSF data transfer protection

- FPT_ITT.1.1** The TSF shall protect TSF data from [*disclosure, modification*] when it is transmitted between separate parts of the TOE.

5.2.5 TOE Access (FTA)

FTA_SSL.3 –TSF-initiated Termination

- FTA_SSL.3.1** The TSF shall terminate a **Management Console** an interactive session after a [**time period of 60 minutes has elapsed**].

FTA_SSL.4 –User-initiated Termination

- FTA_SSL.4.1** The TSF shall allow user-initiated termination of the user’s own interactive session.

5.2.6 Trusted Path/Channels (FTP)

FTP_ITC.1 – Inter-TSF trusted channel

- FTP_ITC.1.1** The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.
- FTP_ITC.1.2** The TSF shall permit [*the TSF*] to initiate communication via the trusted channel.
- FTP_ITC.1.3** The TSF shall initiate communication via the trusted channel for [**remote authentication to a LDAP server**].

FTP_TRP.1 – Trusted path

- FTP_TRP.1.1** The TSF shall provide a communication path between itself and [*remote*] users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from [*disclosure, [undetected modification]*].
- FTP_TRP.1.2** The TSF shall permit [*remote users*] to initiate communication via the trusted path.
- FTP_TRP.1.3** The TSF shall require the use of the trusted path for [*initial user authentication, [all remote administrative actions]*].

5.2.7 Intrusion Prevention and Detection System (IPD)

IPD_IDC.1 – IDS data collection

- IPB_IDC.1.1** The TSF shall be able to collect IDS data from the following monitored resources, based on configured detection rules: [*Windows event logs, text logs, registry keys, files, syslog daemon, C2 audit log, WTMP file, [TOE agent error messages, TOE agent status messages]*].

IPD_IER.1 – Controlled event review

- IPD_IER.1.1** The TSF shall provide [**authorized users with the System Administrator**] with the capability to read [**all event data**] from the generated events.
- IPD_IER.1.2** The TSF shall provide the event data in a manner suitable for the user to interpret the information.
- IPD_IER.1.3** The TSF shall prohibit all users read access to the event data, except those users that have been granted explicit read access.

IPD_IER.2 – Selectable event review

- IPD_IER.2.1** The TSF shall provide the ability to apply [**searches and filtering**] of event data based on [**the following criteria**]:
- **Searches based on values of specified event fields and combinations of logical and conditional operators**
 - **Filtering based on event type and period of time**].

IPD_PBP.1 – Simple policy-based prevention

- IPD_PBP.1.1** The TSF shall enforce an intrusion prevention policy to assets based on the following: [
- **Subjects: processes—name, user, group, command line arguments, signature flag, publisher name, file hash**
 - **Resources:**
 - **processes—name, user, group, command line arguments, signature flag, publisher name, file hash, permissions**
 - **memory—address, access permissions**
 - **files—name, access permissions**
 - **registry keys—name, access permissions**
 - **network connections—IP address, TCP port, UDP port**

].

- IPD_PBP.1.2** The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled resources is allowed: **[a process can perform a requested operation on a resource if the requested operation is not explicitly blocked by a policy rule]**.
- IPD_PBP.1.3** The TSF shall explicitly authorize access of subjects to resources based on the following additional rules: **[If the policy is set to “Disable Prevention”, then all processes are allowed to access all resources]**.
- IPD_PBP.1.4** The TSF shall explicitly deny access of subjects to resources based on the following additional rules: **[If the policy is set to “Application Control”, then all processes that are not part of the Windows operating system are prohibited from executing unless they are explicitly listed in the policy configuration]**.

IPD_RCT.1 – Event generation

- IPD_RCT.1.1** The TSF shall be able to generate an event in response to identified violations of **[prevention, detection]** policies.
- IPD_RCT.1.2** The TSF shall record within each event at least the following information: source of the event; date and time the event occurred; type of event; severity of event; description of event.

IPD_RCT.2 – Alert definition and reaction

- IPD_RCT.2.1** The TSF shall be able to trigger an alert when **[the conditions**
- **An event matches a configured filter rule, and**
 - **A configured minimum number of events that should trigger an alert notification occur within a configured time window**
-] are met.**
- IPD_RCT.2.2** The TSF shall send a notification to **[configured notification destinations, which can be:**
- **E-mail address**
 - **SNMP server**
 - **text file**
-] when an alert is triggered.**

IPD_SBD.1 – Signature-based detection

- IPD_SBD.1.1** The TSF shall be able to detect malware using signatures on Linux Systems.

IPD_STG.1 – Protected event storage

- IPD_STG.1.1** The TSF shall protect the stored event data from unauthorized deletion.
- IPD_STG.1.2** The TSF shall be able to **[prevent]** unauthorized modifications to stored event data.

5.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are the EAL 2 components as specified in Part 3 of the Common Criteria. No operations are applied to the assurance components.

Requirement Class	Requirement Component
ADV: Development	ADV_ARC.1 – Security architecture description
	ADV_FSP.2 – Security-enforcing functional specification
	ADV_TDS.1 – Basic design
AGD: Guidance documents	AGD_OPE.1 – Operational user guidance
	AGD_PRE.1 – Preparative procedures

Requirement Class	Requirement Component
ALC: Life-cycle support	ALC_CMC.2 – Use of the CM system
	ALC_CMS.2 – Parts of the TOE CM coverage
	ALC_DEL.1 – Delivery procedures
	ALC_FLR.1 – Basic flaw remediation
ASE: Security Target evaluation	ASE_CCL.1 – Conformance claims
	ASE_ECD.1 – Extended components definition
	ASE_INT.1 – ST introduction
	ASE_OBJ.2 – Security objectives
	ASE_REQ.2 – Derived security requirements
	ASE_SPD.1 – Security problem definition
	ASE_TSS.1 – TOE summary specification
ATE: Tests	ATE_COV.1 – Evidence of coverage
	ATE_FUN.1 – Functional testing
	ATE_IND.2 – Independent testing – sample
AVA: Vulnerability assessment	AVA_VAN.2 – Vulnerability analysis

Table 7 – TOE Security Assurance Components

5.3.1 Development (ADV)

ADV_ARC.1 – Security architecture description

- ADV_ARC.1.1D** The developer shall design and implement the TOE so that the security features of the TSF cannot be bypassed.
- ADV_ARC.1.2D** The developer shall design and implement the TSF so that it is able to protect itself from tampering by untrusted active entities.
- ADV_ARC.1.3D** The developer shall provide a security architecture description of the TSF.
- ADV_ARC.1.1C** The security architecture description shall be at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE design document.
- ADV_ARC.1.2C** The security architecture description shall describe the security domains maintained by the TSF consistently with the SFRs.
- ADV_ARC.1.3C** The security architecture description shall describe how the TSF initialization process is secure.
- ADV_ARC.1.4C** The security architecture description shall demonstrate that the TSF protects itself from tampering.
- ADV_ARC.1.5C** The security architecture description shall demonstrate that the TSF prevents bypass of the SFR-enforcing functionality.
- ADV_ARC.1.1E** The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.2 – Security-enforcing functional specification

- ADV_FSP.2.1D** The developer shall provide a functional specification.
- ADV_FSP.2.2D** The developer shall provide a tracing from the functional specification to the SFRs.

ADV_FSP.2.1C	The functional specification shall completely represent the TSF.
ADV_FSP.2.2C	The functional specification shall describe the purpose and method of use for all TSFI.
ADV_FSP.2.3C	The functional specification shall identify and describe all parameters associated with each TSFI.
ADV_FSP.2.4C	For each SFR-enforcing TSFI, the functional specification shall describe the SFR-enforcing actions associated with the TSFI.
ADV_FSP.2.5C	For each SFR-enforcing TSFI, the functional specification shall describe direct error messages resulting from processing associated with the SFR-enforcing actions.
ADV_FSP.2.6C	The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.
ADV_FSP.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_FSP.2.2E	The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

ADV_TDS.1 – Basic design

ADV_TDS.1.1D	The developer shall provide the design of the TOE.
ADV_TDS.1.2D	The developer shall provide a mapping from the TSFI of the functional specification to the lowest level of decomposition available in the TOE design.
ADV_TDS.1.1C	The design shall describe the structure of the TOE in terms of subsystems.
ADV_TDS.1.2C	The design shall identify all subsystems of the TSF.
ADV_TDS.1.3C	The design shall provide the behavior summary of each SFR-supporting or SFR non-interfering TSF subsystem.
ADV_TDS.1.4C	The design shall summarise the SFR-enforcing behavior of the SFR-enforcing subsystems.
ADV_TDS.1.5C	The design shall provide a description of the interactions among SFR-enforcing subsystems of the TSF, and between the SFR-enforcing subsystems of the TSF and other subsystems of the TSF.
ADV_TDS.1.6C	The mapping shall demonstrate that all TSFIs trace to the behavior described in the TOE design that they invoke.
ADV_TDS.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_TDS.1.2E	The evaluator shall determine that the design is an accurate and complete instantiation of all security functional requirements.

5.3.2 Guidance Documents (AGD)

AGD_OPE.1 – Operational user guidance

AGD_OPE.1.1D	The developer shall provide operational user guidance.
AGD_OPE.1.1C	The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.
AGD_OPE.1.2C	The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.
AGD_OPE.1.3C	The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.
AGD_OPE.1.4C	The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5C	The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.
AGD_OPE.1.6C	The operational user guidance shall, for each user role, describe the security controls to be followed in order to fulfil the security objectives for the operational environment as described in the ST.
AGD_OPE.1.7C	The operational user guidance shall be clear and reasonable.
AGD_OPE.1.1E	The evaluator <i>shall confirm</i> that the information provided meets all requirements for content and presentation of evidence.

AGD_PRE.1 – Preparative procedures

AGD_PRE.1.1D	The developer shall provide the TOE including its preparative procedures.
AGD_PRE.1.1C	The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer’s delivery procedures.
AGD_PRE.1.2C	The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.
AGD_PRE.1.1E	The evaluator <i>shall confirm</i> that the information provided meets all requirements for content and presentation of evidence.
AGD_PRE.1.2E	The evaluator <i>shall apply</i> the preparative procedures to confirm that the TOE can be prepared securely for operation.

5.3.3 Life-cycle Support (ALC)

ALC_CMC.2 – Use of the CM system

ALC_CMC.2.1D	The developer shall provide the TOE and a unique reference for the TOE.
ALC_CMC.2.2D	The developer shall provide the CM documentation.
ALC_CMC.2.3D	The developer shall use a CM system.
ALC_CMC.2.1C	The TOE shall be labelled with its unique reference.
ALC_CMC.2.2C	The CM documentation shall describe the method used to uniquely identify the configuration items.
ALC_CMC.2.3C	The CM system shall uniquely identify all configuration items.
ALC_CMC.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_CMS.2 – Parts of the TOE CM coverage

ALC_CMS.2.1D	The developer shall provide a configuration list for the TOE.
ALC_CMS.2.1C	The configuration list shall include the following: the TOE itself; the evaluation evidence required by the SARs; and the parts that comprise the TOE.
ALC_CMS.2.2C	The configuration list shall uniquely identify the configuration items.
ALC_CMS.2.3C	For each TSF relevant configuration item, the configuration list shall indicate the developer of the item.
ALC_CMS.2.1E	The evaluator <i>shall confirm</i> that the information provided meets all requirements for content and presentation of evidence.

ALC_DEL.1 – Delivery procedures

ALC_DEL.1.1D	The developer shall document and provide procedures for delivery of the TOE or parts of it to the consumer.
---------------------	---

ALC_DEL.1.2D	The developer shall use the delivery procedures.
ALC_DEL.1.1C	The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to the consumer.
ALC_DEL.1.1E	The evaluator <i>shall confirm</i> that the information provided meets all requirements for content and presentation of evidence.

ALC_FLR.1 – Basic flaw remediation

ALC_FLR.1.1D	The developer shall document and provide flaw remediation procedures addressed to TOE developers.
ALC_FLR.1.1C	The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.
ALC_FLR.1.2C	The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.
ALC_FLR.1.3C	The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.
ALC_FLR.1.4C	The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.
ALC_FLR.1.1E	The evaluator <i>shall confirm</i> that the information provided meets all requirements for content and presentation of evidence.

5.3.4 Security Target Evaluation (ASE)

ASE_CCL.1 – Conformance claims

ASE_CCL.1.1D	The developer shall provide a conformance claim.
ASE_CCL.1.2D	The developer shall provide a conformance claim rationale.
ASE_CCL.1.1C	The conformance claim shall identify the edition of the CC to which the ST and the TOE claim conformance.
ASE_CCL.1.2C	The CC conformance claim shall describe the conformance of the ST to CC Part 2 as either CC Part 2 conformant or CC Part 2 extended.
ASE_CCL.1.3C	The CC conformance claim shall describe the conformance of the ST to CC Part 3 as either CC Part 3 conformant or CC Part 3 extended.
ASE_CCL.1.4C	The CC conformance claim shall be consistent with the extended components definition.
ASE_CCL.1.5C	The conformance claim shall identify a PP-configuration, or all PPs and security requirement packages to which the ST claims conformance.
ASE_CCL.1.6C	The conformance claim shall describe any conformance of the ST to a package as either package-conformant or package-augmented.
ASE_CCL.1.7C	The conformance claim shall describe any conformance of the ST to a PP as PP-conformant.
ASE_CCL.1.8C	The conformance claim rationale shall demonstrate that the TOE type is consistent with the TOE type in the PP-Configuration or PPs for which conformance is being claimed.
ASE_CCL.1.9C	The conformance claim rationale shall demonstrate that the statement of the security problem definition is consistent with the statement of the security problem definition in the PP-Configuration ¹ , PPs and any functional packages for which conformance is being claimed.
ASE_CCL.1.10C	The conformance claim rationale shall demonstrate that the statement of security objectives is consistent with the statement of security objectives in the PP-Configuration ² , PPs, and any functional package for which conformance is being claimed.

¹ In practice, this refers to the union of SPDs defined in the PP-Configuration components.

² In practice, this refers to the union of security objectives defined in the PP-Configuration components.

ASE_CCL.1.11C	The conformance claim rationale shall demonstrate that the statement of security requirements is consistent with the statement of security requirements in the PP-Configuration ³ , PPs, and any functional packages for which conformance is being claimed.
ASE_CCL.1.12C	The conformance claim for PP(s) or a PP-Configuration shall be exact, strict, or demonstrable or a list of conformance types.
ASE_CCL.1.13C	If the conformance claim identifies a set of Evaluation methods and Evaluation activities derived from CEM work units that shall be used to evaluate the TOE then this set shall include all those that are included in any package, PP, or PP-Module in a PP-Configuration to which the ST claims conformance, and no others.
ASE_CCL.1.1E	The evaluator <i>shall confirm</i> that the information provided meets all requirements for content and presentation of evidence.

ASE_ECD.1 – Extended components definition

ASE_ECD.1.1D	The developer shall provide a statement of security requirements.
ASE_ECD.1.2D	The developer shall provide an extended components definition.
ASE_ECD.1.1C	The statement of security requirements shall identify all extended security requirements.
ASE_ECD.1.2C	The extended components definition shall define an extended component for each extended security requirement.
ASE_ECD.1.3C	The extended components definition shall describe how each extended component is related to the existing CC components, families, and classes.
ASE_ECD.1.4C	The extended components definition shall use the existing CC components, families, classes, and methodology as a model for presentation.
ASE_ECD.1.5C	The extended components shall consist of measurable and objective elements such that conformance or nonconformance to these elements can be demonstrated.
ASE_ECD.1.1E	The evaluator <i>shall confirm</i> that the information provided meets all requirements for content and presentation of evidence.
ASE_ECD.1.2E	The evaluator <i>shall confirm</i> that no extended component may be clearly expressed using existing components.

ASE_INT.1 – ST introduction

ASE_INT.1.1D	The developer shall provide an ST introduction.
ASE_INT.1.1C	The ST introduction shall contain an ST reference, a TOE reference, a TOE overview and a TOE description.
ASE_INT.1.2C	The ST reference shall uniquely identify the ST.
ASE_INT.1.3C	The TOE reference shall identify the TOE.
ASE_INT.1.4C	The TOE overview shall summarize the usage and major security features of the TOE.
ASE_INT.1.5C	The TOE overview shall identify the TOE type.
ASE_INT.1.6C	The TOE overview shall identify any non-TOE hardware/software/firmware required by the TOE.
ASE_INT.1.7C	For a multi-assurance ST, the TOE overview shall describe the TSF organization in terms of the sub-TSFs defined in the PP-Configuration the ST claims conformance to.
ASE_INT.1.8C	The TOE description shall describe the physical scope of the TOE.
ASE_INT.1.9C	The TOE description shall describe the logical scope of the TOE.
ASE_INT.1.1E	The evaluator <i>shall confirm</i> that the information provided meets all requirements for content and presentation of evidence.

³ In practice, this refers to the union of SFRs defined in the PP-Configuration components

ASE_INT.1.2E The evaluator *shall confirm* that the TOE reference, the TOE overview, and the TOE description are consistent with each other.

ASE_OBJ.2 – Security objectives

ASE_OBJ.2.1D The developer shall provide a statement of security objectives.

ASE_OBJ.2.2D The developer shall provide a security objectives rationale.

ASE_OBJ.2.1C The statement of security objectives shall describe the security objectives for the TOE and the security objectives for the operational environment.

ASE_OBJ.2.2C The security objectives rationale shall trace each security objective for the TOE back to threats countered by that security objective and OSPs enforced by that security objective.

ASE_OBJ.2.3C The security objectives rationale shall trace each security objective for the operational environment back to threats countered by that security objective, OSPs enforced by that security objective, and assumptions upheld by that security objective.

ASE_OBJ.2.4C The security objectives rationale shall demonstrate that the security objectives counter all threats.

ASE_OBJ.2.5C The security objectives rationale shall demonstrate that the security objectives enforce all OSPs.

ASE_OBJ.2.6C The security objectives rationale shall demonstrate that the security objectives for the operational environment uphold all assumptions.

ASE_OBJ.2.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

ASE_REQ.2 – Derived security requirements

ASE_REQ.2.1D The developer shall provide a statement of security requirements.

ASE_REQ.2.2D The developer shall provide a security requirements rationale.

ASE_REQ.2.1C The statement of security requirements shall describe the SFRs and the SARs.

ASE_REQ.2.2C For a single-assurance ST, the statement of security requirements shall define the global set of SARs that apply to the entire TOE. The sets of SARs shall be consistent with the PPs or PP Configuration to which the ST claims conformance.

ASE_REQ.2.3C For a multi-assurance ST, the statement of security requirements shall define the global set of SARs that apply to the entire TOE and the sets of SARs that apply to each sub-TSF. The sets of SARs shall be consistent with the multi-assurance PP-Configuration to which the ST claims conformance.

ASE_REQ.2.4C All subjects, objects, operations, security attributes, external entities and other terms that are used in the SFRs and the SARs shall be defined.

ASE_REQ.2.5C The statement of security requirements shall identify all operations on the security requirements.

ASE_REQ.2.6C All operations shall be performed correctly.

ASE_REQ.2.7C Each dependency of the security requirements shall either be satisfied, or the security requirements rationale shall justify the dependency not being satisfied.

ASE_REQ.2.8C The security requirements rationale shall demonstrate that the SFRs meet all security objectives for the TOE.

ASE_REQ.2.9C The security requirements rationale shall explain why the SARs were chosen.

ASE_REQ.2.10C The statement of security requirements shall be internally consistent.

ASE_REQ.2.11C If the ST defines sets of SARs that expand the sets of SARs of the PPs or PP-Configuration it claims conformance to, the security requirements rationale shall include an assurance rationale that justifies the consistency of the extension and provides a rationale for the disposition of any

Evaluation methods and Evaluation activities identified in the conformance statement that are affected by the extension of the sets of SARs.

ASE_REQ.2.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

ASE_SPD.1 – Security problem definition

ASE_SPD.1.1D The developer shall provide a security problem definition.

ASE_SPD.1.1C The security problem definition shall describe the threats.

ASE_SPD.1.2C All threats shall be described in terms of a threat agent, an asset, and an adverse action.

ASE_SPD.1.3C The security problem definition shall describe the OSPs.

ASE_SPD.1.4C The security problem definition shall describe the assumptions about the operational environment of the TOE.

ASE_SPD.1.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

ASE_TSS.1 – TOE summary specification

ASE_TSS.1.1D The developer shall provide a TOE summary specification.

ASE_TSS.1.1C The TOE summary specification shall describe how the TOE meets each SFR.

ASE_TSS.1.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

ASE_TSS.1.2E The evaluator *shall confirm* that the TOE summary specification is consistent with the TOE overview and the TOE description.

5.3.5 Tests (ATE)

ATE_COV.1 – Evidence of coverage

ATE_COV.1.1D The developer shall provide evidence of the test coverage.

ATE_COV.1.1C The evidence of the test coverage shall show the correspondence between the tests in the test documentation and the TSFIs in the functional specification.

ATE_COV.1.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

ATE_FUN.1 – Functional testing

ATE_FUN.1.1D The developer shall test the TSF and document the results.

ATE_FUN.1.2D The developer shall provide test documentation.

ATE_FUN.1.1C The test documentation shall consist of test plans, expected test results and actual test results.

ATE_FUN.1.2C The test plans shall identify the tests to be performed and describe the scenarios for performing each test. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.3C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.4C The actual test results shall be consistent with the expected test results.

ATE_FUN.1.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2 – Independent testing – sample

ATE_IND.2.1D The developer shall provide the TOE for testing.

ATE_IND.2.1C The TOE shall be suitable for testing.

- ATE_IND.2.2C** The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.
- ATE_IND.2.1E** The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.
- ATE_IND.2.2E** The evaluator *shall execute* a sample of tests in the test documentation to verify the developer test results.
- ATE_IND.2.3E** The evaluator *shall test* a subset of the TSF to confirm that the TSF operates as specified.

5.3.6 Vulnerability Assessment (AVA)

AVA_VAN.2 – Vulnerability analysis

- AVA_VAN.2.1D** The developer shall provide the TOE for testing.
- AVA_VAN.2.2D** The developer shall provide a list of third party components included in the TOE and the TOE delivery.
- AVA_VAN.2.1C** The TOE shall be suitable for testing.
- AVA_VAN.2.2C** The list of third party components shall include components provided by third parties, and that are part of the TOE or otherwise part of the TOE delivery.
- AVA_VAN.2.1E** The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.
- AVA_VAN.2.2E** The evaluator *shall perform* a search of public domain sources to identify potential vulnerabilities in the TOE the components in the list of third party components, and specific IT products in the environment that the TOE depends on.
- AVA_VAN.2.3E** The evaluator *shall perform* an independent vulnerability analysis of the TOE using the guidance documentation, functional specification, TOE design and security architecture description to identify potential vulnerabilities in the TOE.
- AVA_VAN.2.4E** The evaluator *shall conduct* penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

6. TOE Summary Specification

This section describes the following security functions implemented by the TOE to satisfy the SFRs claimed in Section 5.2:

- Security audit
- Identification and authentication
- Security management
- Protection of the TSF
- TOE Access
- Trusted path
- Intrusion prevention and detection.

6.1 Security Audit

The TOE records events and messages related to agent and Management Server activity in a number of log files. Agents support the following three log files:

- Agent service log (SISIPSService.log)—contains logs related to agent service operation, application of policies and configuration settings, and communication with the Management Server
- Event log (SISIDSEvents*.csv)—contains all events recorded by the agent. The asterisk in the file name represents a version number.
- Real-time event log (SISRTEvents*.csv)—contains real-time events processed by the agent. This is a temporary file that is used to speed processing of real-time events. Events are also forwarded to the Management Server based on the agent's configured log rules. The asterisk in the file name represents a version number.

If bulk logging is enabled for the agent, the Event log file is uploaded to the Management Server, where it is stored. Bulk logging captures events to compressed log files instead of transmitting all events in real-time to the database for storage. The TOE includes the Bulk Loader Utility for loading the contents of a compressed bulk log file into the database.

Agent-transmitted events and events generated by the Management Server are loaded into the SDCSSEVENT table in the database. The Bulk Loader Utility loads events into the ANALYSIS_EVENT table by default, but can be directed to load them into the SDCSSEVENT table instead. The database protects the stored audit records from unauthorized deletion or modification.

The TOE can generate audit records for the following auditable events:

- The start-up and shutdown of audit functions (the audit function automatically starts at system start-up and can only be shutdown at system shutdown. In both instances, a record of the event is recorded.)
- Reading of information from the audit records
- All use of the authentication mechanism
- All use of the user identification mechanism
- All modifications in the behavior of the functions of the TSF
- All modifications to the values of TSF data
- Use of the management functions
- Modifications to the group of users that are part of a role.

All audit events include the date and time of the event, the type of event, the subject identity, and the outcome of the event, such as whether it was a success or failure.

By default, the TOE does not generate audit records of searches or queries of the audit records, but this can be enabled by an authorized user with the System Administrator role via the **Settings** page.

The TOE provides various capabilities for System Administrators to view audit records via the **Investigate->Events** page of the Management Console, and selecting the type 'Audit' for the Event Classification. The **Investigate** page provides authorized users with the System Administrator role with capabilities to view all audit records, search for audit records, and filter audit records based on audit event type and period of time. Searches can be simple or advanced. To perform a simple search, a System Administrator selects one of the audit record fields and specifies a value to search for in that field. When performing an advanced search, a System Administrator can select multiple audit record fields, specify a search value for each selected field, and conditions (e.g., equals, does not equal, less than, greater than).

All data is presented in such a manner that it can be read and the contents of the data can be interpreted; thus the reader can understand the content of the information presented.

The Security Audit security function satisfies the following security functional requirements:

- FAU_GEN.1—audit records are generated for security relevant events and include the date and time of the event, type of event, subject identity, and outcome of the event.
- FAU_SAR.1—the TOE provides authorized users (System Administrators, Managers, and Viewers) with the capability to read all audit information from the audit records. The audit records are displayed in a manner suitable for the authorized user to interpret the information.
- FAU_SAR.2—the TOE prohibits all users read access to the audit records, except those users that have been granted explicit read-access.
- FAU_SAR.3— the TOE provides capabilities to search for audit records and to filter displayed audit records based on audit event type and time span.
- FAU_STG.2—the TOE protects stored audit records from unauthorized modification and deletion.

6.2 Identification and Authentication

The TOE maintains user accounts that provide secure access to the management console. The user account includes the following attributes associated with the user: username; password; roles; and e-mail address information.

In order to access the functions provided by the TOE via the management console, the user must first be identified and authenticated. The TOE supports the following user authentication methods:

- Local password-based authentication— as part of the login process, the user submits a password that must match the password associated with the user account
- Remote password-based authentication using Active Directory—as part of the login process, the user enters their username in **domain\username** format and the password associated with the specified domain account.

The TOE enforces the following restrictions on passwords of local (i.e., not Active Directory) accounts:

- Minimum length of eight characters
- Maximum length of twenty characters
- At least one upper case
- At least one lower case
- At least one number
- At least one of the following: ! @ # \$ % ^ & - + = () { } _] .

To protect the passwords, the TOE only stores salted hashes (Base64-encoded SHA-256 with 128-bit random salt value) of the passwords in the Management Server database. When a password is submitted for authentication during login, the TOE hashes the submitted password and compares the resultant value with the hash value stored with the applicable user account. If either the login name or the password is incorrect, the login request fails and no

administrative functions are made available. As a result of a successful login, the interactive session is established and the administrative functions appropriate to the user's assigned roles are made available.

To protect against brute force attacks against local TOE accounts, the TOE implements an account lockout mechanism. This mechanism is triggered when a configurable number of failed authentication attempts has been reached. Once the configured number has been reached, the offending account is locked from accessing the TOE. The lock is released automatically after a configurable period of time has elapsed, or manually by a TOE System Administrator that unlocks the account.

A further means of interacting with the TOE is via REST APIs, which provide support for additional platforms and integration. In order to make use of the services provided by the REST APIs, the client must first generate a token. The client submits a user identity and associated password as part of an HTTP POST request (submitted over HTTPS) that includes a JSON request. If the TOE successfully authenticates the user identity, it returns a token to the client that the client includes with every subsequent request during the session. The session ends when the client logs out or the TOE invalidates the session (e.g., due to session inactivity).

The Identification and Authentication function satisfies the following security functional requirements:

- FIA_AFL.1 – The TOE enforces an account lockout mechanism that can be configured by an System Administrator.
- FIA_ATD.1—the TOE maintains the following security attributes associated with each user: username; password; roles; and e-mail address.
- FIA_SOS.1—the TOE enforces a password policy that ensures all secrets (i.e., passwords) associated with user accounts meet policy requirements.
- FIA_UAU.2—the TOE requires each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.
- FIA_UAU.5—the TOE supports local password-based authentication of user identities and remote authentication using Active Directory username and password.
- FIA_UID.2—the TOE requires each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

6.3 Security Management

6.3.1 Security Management Roles

When an System Administrator creates a Management Console user account, the account is associated with one or more user roles. The user roles determine what functions the user can perform in the Management Console.

The TOE supports the following built-in security management roles for management console users:

- System Administrators — Has full access on the TOE including User Management, RBAC, and AD Integration configurations. The default account ("symadmin") that was created during TOE installation is assigned the System Administrator role.
- Viewers — Has read-only access to most areas of the TOE Management Console except for the following:
 - It does not have access to view the Active Directory integration.
 - It does not have access to even view the existence of, nor view the details of, the Settings tabs: User Management, Certificates, Server Registry, and License tabs
- Managers — users with the Managers role can log on to the Management Console and make changes to agents and policies, such as modifying agent and group policy and configuration settings, and creating and modifying policies. Managers may optionally access queries and reports on the Reports page. Managers can access all agent groups on the Assets page. Unlike the System Administrator role, they do not have access to User Management, RBAC, or AD Integration settings.

6.3.2 Security Management Functions

TOE System administrators manage the TOE and its security functions using the Management Console. The Management Console provides the following capabilities:

- Manage policies
- Manage configurations
- Manage Management Console user accounts and roles
- Manage assets
- Manage security groups
- Manage alerts and notifications
- View events
- Manage audit function
- Manage users and groups
- Modify user passwords

The Management Console is also used to manage the Management Server itself. The Security Management function satisfies the following security functional requirements:

- FMT_MOF.1—the TOE restricts the ability to manage the behavior of the audit function to the System Administrator and Manager roles.
- FMT_MTD.1(*)—the TOE restricts the ability to manage TSF data to the System Administrators and System Administrator roles.
- FMT_SMF.1—the TOE provides the capabilities necessary to manage the security of the TOE.
- FMT_SMR.1—the TOE maintains the following built-in roles: System Administrators; Managers; Viewers. The TOE is able to associate users with these roles.

6.4 Protection of the TSF

The TOE is composed of the Management Server, Management Console, Communication Server, Agents, and database. The Management Console and the database can be installed on the same host as the Management Server or installed on separate hosts in a distributed deployment.

The TOE can be configured to protect communication between the Management server and Management Console and database instances, and between the Communication Server and Windows agents, using TLS. The TOE supports TLS v1.2 and TLS v1.3. The TOE supports the following TLS cipher suites, as defined in RFC 3268:

- TLS 1.2
 - TLS_DHE_RSA_WITH_AES_128_GCM_SHA256,
 - TLS_DHE_RSA_WITH_AES_256_GCM_SHA384,
 - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,
 - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,
 - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,
 - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,
 - TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256,
 - TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256,
- TLS 1.3
 - TLS_AES_128_GCM_SHA256,
 - TLS_AES_256_GCM_SHA384,
 - TLS_CHACHA20_POLY1305_SHA256

The Protection of the TSF function satisfies the following security functional requirement:

- FPT_ITT.1—the TOE uses TLS to protect TSF data communicated between distributed parts of the TOE.

6.5 TOE Access

The TOE will terminate interactive sessions after a period of inactivity configurable by a System Administrator. By default, interactive sessions are terminated after 60 minutes of inactivity. The TOE also allows user-initiated termination of the user's own interactive session by explicitly logging off.

The TOE access function satisfies the following security functional requirements:

- FTA_SSL.3—the TOE will terminate an interactive user session after 60 minutes of inactivity
- FTA_SSL.4—the TOE allows user-initiated termination of the user's own interactive session.

6.6 Trusted Path/Channels

The TOE provides a fully instrumented REST API that provides the ability to perform common workflows for managing the assets and policies. This provides the capability for System Administrators to remotely connect to the Management Server component of the TOE. All communication with the REST API is via HTTPS, using the versions of TLS and the cipher suites supported by the trusted path. The TOE supports TLS v1.2 and TLS v1.3. The TOE supports the following TLS cipher suites, as defined in RFC 2246:

- TLS 1.2
 - TLS_DHE_RSA_WITH_AES_128_GCM_SHA256,
 - TLS_DHE_RSA_WITH_AES_256_GCM_SHA384,
 -
 - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,
 - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,
 - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,
 - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,
 - TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256,
 - TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256,
- TLS 1.3
 - TLS_AES_128_GCM_SHA256,
 - TLS_AES_256_GCM_SHA384,
 - TLS_CHACHA20_POLY1305_SHA256

The trusted path is used for initial authentication and all subsequent administrative actions via the REST API over HTTPS. The use of HTTPS ensures all communication over the trusted path is protected from disclosure and modification.

The TOE also provides a fully instrumented REST API that provides the ability to perform common workflows for managing the assets and policies. This provides the capability for external IT entities to connect to the Management Server component of the TOE. All communication with the REST API is via HTTPS, using the versions of TLS and the cipher suites supported by the trusted path capability described above.

The Trusted Path/Channels function satisfies the following security functional requirements:

- FTP_TRP.1—the TOE provides a trusted path for System Administrators to communicate with the TOE via the REST API, using HTTPS.

6.7 Intrusion Prevention and Detection

The TOE provides a policy-based approach to intrusion prevention and intrusion detection. The TOE is able to control and monitor what programs and users can do to computers. Agent software at the endpoints controls and monitors behavior based on policy.

The TOE policy library contains prevention and detection policies that an System Administrator or Manager can deploy and customize to protect the network and endpoints, as follows:

- A **prevention policy** is a collection of rules that governs how processes and users access resources. Prevention policies can, for example: contain a list of files and registry keys that no program or user can access; contain a list of UDP and TCP ports that permit and deny traffic; deny access to startup folders; or define the actions to take when unacceptable behavior occurs.
- A **detection policy** is a collection of rules that are configured to detect specific events. An agent can enforce one or more detection policies simultaneously. Detection policies can, for example, be configured to generate events when the following are detected: files and registry keys are deleted; known, vulnerable CGI scripts are run on Microsoft Internet Information Server (IIS); USB devices are inserted and removed from computers; network shares are created and deleted.

6.7.1 Intrusion Prevention Policies

Prevention policies protect against inappropriate modification of system resources. The policies confine each process on a computer to its normal behavior. Programs that are identified as critical to system operation are given specific behavior controls; generic behavior controls provide compatibility for other services and applications.

Prevention policies take advantage of the common characteristics of services and applications. Prevention policies divide programs into sandboxes and provide behavior control to the programs based on the sandbox to which the program belongs. Each program that runs on a computer is placed in exactly one sandbox at any given time. Most sandboxes and their associated behavior controls are tailored for a specific service or application. The TOE takes advantage of this information by imposing stringent behavior controls for services and applications. Only the necessary resources for each program are given read and write access privileges.

Services and applications that do not have a specifically tailored sandbox are placed into default sandboxes. These default sandboxes use generic behavior controls that are not concerned with the allowed behavior of the services and applications, but contain behavior controls for events that should never be allowed.

Agents use the following mechanisms to control the behavior of processes running on the asset the agent is protecting, based on the configuration of the prevention policy it applies:

- Process Access Control—the TOE provides Process Access Control to enable the agent to control access to a running process. The prevention policy can specify if a calling process can open a target process and the permissions that it has to do so, except that a process can always access a direct child process and can always access itself. The following information can be specified in a Process Access Control rule:
 - The target process that is accessed, including: program name; user; group; command line arguments; signature flag; publisher name; and file hash.
 - The calling process that attempts to open the target process, using any of the process attributes of the calling process.
 - The process permissions allowed when accessing this target process. These permissions can include any combination of the individual operating system process access permissions.
 - The actions to take if the permissions requested are greater than those allowed.
- Memory controls—the TOE provides multiple techniques to defend against malicious code being inserted into or executed from undesirable memory locations within a running process. The TOE can detect when code is executed from overflowed memory space as well as control unusual memory modifications a process can make within itself. The specific memory related actions that are controllable via policy settings include the following:
 - Buffer overflow detection
 - Unusual memory allocations

- Unusual memory permission changes
- Disabling Data Execution Prevention

These in-process memory constraints are complementary to and provide separate defenses to the Process Access Control mechanism.

- Resource lists—resource lists specify the files, registry keys (for Windows platforms), and processes that are writable, read-only, or blocked for both read and write. Resource lists override any controls on the same resources that are specified by the specific application sandbox controls in a policy. When a resource is specified in more than one resource list, the policy applies the following general precedence rules:
 - Resource lists in an application sandbox take precedence over global resource lists
 - Within an option level, the least restrictive resource list takes precedence.
- Network controls—these options control connections to and from an agent computer. The network control options comprise the following:
 - The Components options define the IP addresses, TCP ports, and UDP ports that are referenced in the network rules
 - The Network Rules options define the ordered rules that control connections to and from an agent computer
 - The Default Rules options define default rule actions and log settings.

An System Administrator or Manager can explicitly authorize all processes to access all resources by setting the “Disable prevention” global policy option. This disables prevention of policy violations for the entire asset to which the policy is being applied.

For Windows-based assets, the prevention policy can be configured to apply one of the following protection strategies:

- Basic strategy—provides protection for the operating system and a set of common applications. This strategy confines the remaining applications (both services and interactive programs) within the Basic sandbox. The Basic sandbox is set up to be highly compatible for programs running within it.
- Hardened strategy—provides protection for the operating system and a set of common applications. This strategy confines the remaining applications (both service and interactive programs) within the Hardened sandbox. The Hardened sandbox is set up to confine the programs running within it so that they cannot modify operating system or other application resources.
- Protected Whitelisting strategy—provides protection for the operating system, and prohibits the launching of applications, except those that are explicitly listed in the policy configuration.

6.7.2 Intrusion Detection Policies

A detection policy is a collection of rules that are configured to detect specific events and take action. Detection policies define which system events or user-defined criteria are selected, which criteria are ignored, and what actions are performed after select and ignore criteria are met.

Detection policies monitor events and syslogs and report anomalous behavior. Features include policy-based auditing and monitoring; log consolidation for easy search, archival, and retrieval; event analysis and response capabilities; and file and registry protection and monitoring.

The detection policies use the following collectors:

- Event log—this collector looks for matches in Windows event log files. The event log files are the Microsoft standard format .evt files. In standard installations, three event log files exist: Security; System; and Application

- Registry—watches changes to user-specified registry keys. The collector can watch changes to key/value, operations (created, modified, deleted), operation results (success, failure, either), and process.
- File—determines how agents monitor files. The file collector detects changes to system critical files and is associated with the filewatch rule type, which logs activity to files and directories. The policy can specify the file/directory to watch, the file operation, and the protection settings.
- Syslog—watches for syslog daemon tampering on UNIX operating systems. The `syslogd` daemon must run for the syslog collector to work. Normally, `syslogd` runs at all times on a secured UNIX system. Upon initialization, the syslog collector checks that `syslogd` is running and starts it if it is not running. Subsequently, if `syslogd` is killed while an agent is running, an error event is generated and matched against a suitable Syslogd Tampering policy. No attempts are made to restart `syslogd`.

The syslog collector monitors and parses the following pipe: `/var/log/ids_syslog.pipe`. This pipe is specified in `/etc/syslog.conf`.

- Audit – this collector monitors events from Windows standard system audit logs. By default, the audit collector is enabled. The system audit log sources in Windows are Security, Application, and System
- C2 log—looks for matches in the C2 audit logs on agent computers that run Solaris, HP-UX, and AIX operating systems.
- WTMP—this collector monitors the WTMP logging system on assets of UNIX operating systems. By default, the WTMP Collector is enabled. WTMP collects user authentication and account information. You can specify text patterns to parse. The WTMP file captures successful login events. The WTMP file that is watched varies, depending on the operating system. All UNIX operating systems at one point used the WTMP format, but many now use the newer WTMPX format. On some systems, this filename may be WTMP, WTMPX, or WTMPX, even though the format internally is WTMPX
- BTMP – this collector monitors the BTMP logging system on assets of UNIX operating systems. By default, the BTMP collector is enabled.

Detection policies include various policy options a System Administrator or Manager can use to configure a detection policy for assignment to a target computer. Policy options comprise a simplified set of controls that are used to enable or disable features in a policy. Some options have associated parameters that enable the System Administrator or Manager to customize the behavior of an option.

6.7.3 Reaction

Agents generate events in response to identified violations of the protection or detection policies they enforce. An agent logs events to the Management Server based on the agent's log rules. Log rules comprise the following:

- Filter rules—each filter rule comprises `<field, operator, value>`. A System Administrator or Manager can configure multiple filter rules for each log rule. Events must match all filter rules. Valid fields include: event type; event severity; event date; disposition; event priority; process; username; remote IP. Valid operators include: equals; not equals; in; not in; contains; not contains; greater than; less than. Some operators support the use of wildcard characters in a value. Valid wildcard characters are asterisk (*), which represents zero or more consecutive characters, and question mark (?), which represents exactly one character.
- Transmit action—the following transmit actions are supported:
 - Transmit in real-time—real-time events are actionable events that are transmitted to the Management Server for storage in the Management Server database
 - Bulk log only—bulk log events are events of long-term interest that have no immediate reporting or actionable purpose. They are recorded in log files on the agent computer. When full, the log files are compressed and transferred to the Management Server for storage. Bulk log events are loaded into the Management Server database using the bulk loader utility. The events are loaded into the analysis event table (the default) or the real-time event table.

The details recorded in an event include:

- The name of the agent computer that generated the event
- The date and time when the event occurred
- The type of the event
- The severity of the event
- A brief description of the event.

An agent's prevention policy generates prevention events when applications access computer and network resources that violate the policy's behavior control. The TOE defines the following prevention event types:

- Buffer overflow—contains information about applications that execute code that was inserted using buffer overflows. Buffer overflow events apply to agent computers that run Windows operating system.
- File Access—contains information about applications that access files and directories.
- Mount—contains information about applications that mount or unmount file systems.
- Network Access—contains information about applications that access the TCP/IP network.
- OS Call—contains information about applications that make selected operating system calls that are often exploited by attackers.
- Process Access—contains information about the process modification or process access.
- Process Set—contains information about the assignment of a process to a process set.
- Process Create—contains information about the creation of a process.
- Process Destroy—contains information about the termination of a process.
- Registry Access—contains information about applications that access registry keys.

An agent's detection policy generates detection events when monitored files or registry keys change, or when system or application logs generate events that match the policy's criteria. The TOE defines the following detection event types:

- Audit watch—contains information about audit watch events
- Filewatch—contains information about filewatch events for Windows and UNIX operating systems
- Generic Log—contains information about generic log events
- NT Event Log—contains information about NT event log events
- Prevention Watch—contains information about prevention watch events
- Registry Watch—contains information about registry watch events
- Syslog—contains information about syslog events
- UNIX C2 Log—contains information about C2 events
- UNIX Activity Log—contains information about WTMP events.
- Network Protection
- Threat Blocked—contains information about the blocked events.

Alerts are used to send events of interest to the following destinations:

- Email messages
- SNMP traps
- Text files.

The Alert function polls the Management Server database for events that match an alert filter. When creating an alert, the System Administrator or Manager specifies the following:

- The filter the Alert function uses to compare to events looking for a match
- The minimum number of events that should trigger an alert notification. For example, 100 file access events within <number of> minutes should trigger an alert notification
- The time in minutes that should be the threshold for the alert notification. For example, <X number of> file access events within a time window of 5 minutes should trigger an alert notification.

When a match is found, the Alert function generates and sends notifications in the form of email messages, SNMP traps, and text files that are associated with the alert. The TOE supports email aggregation that combines multiple alerts occurring within a specified time interval into a single email message. Email aggregation prevents flooding email addresses with excessive emails or with messages that exceed size limitations.

System Administrator or Manager can: configure alert settings; add, edit, copy, delete, import and export alerts; enable and disable alerts. The System Administrator or Manager can also view the notifications for the alerts the System Administrator or Manager has created and perform the following actions: view details of notifications; acknowledge a notification so the TOE will not generate further notifications for the alert; mark an acknowledged notification as not acknowledged.

6.7.4 IDS Data Review

The TOE provides various capabilities for System Administrator or Manager to view events via the management console.

The **Events** page in the management console is used to view events that are sent to the Management Server. The **Events** page displays event information reported to the Management Server for the entire agent deployment. The Monitor page provides the System Administrator or Manager with capabilities to view all events or categories of events, search for events, and filter events based on event type and period of time. Searches can be simple or advanced. To perform a simple search, the System Administrator or Manager selects one of the event fields and specifies a value to search for in that field. When performing an advanced search, the System Administrator or Manager can select multiple event fields, specify a search value for each selected field, and combine the fields using logical operators (e.g., AND, OR, NOT) and conditions (e.g., equals, does not equal, less than, greater than).

All data is presented in such a manner that it can be read and the contents of the data can be interpreted; thus the reader can understand the content of the information presented.

6.7.5 Event Storage

The TOE uses log files to record events and messages related to agent and Management Server activity.

Agent log files contain all events processed by an agent. Agent log files are stored on the local computer on which the agent is installed. Agents maintain the following log files for event storage:

- Event log (SISIDSEvents*.csv)—contains all events recorded by the agent. The asterisk in the file name represents a version number.
- Real-time event log (SISIPSRTEvents*.csv)—contains real-time events processed by the agent. This is a temporary file that is used to speed processing of real-time events. Events are also forwarded to the Management Server based on the agent's configured log rules. The asterisk in the file name represents a version number.

If bulk logging is enabled for the agent, the Event log file is uploaded to the Management Server. Bulk logging captures events to compressed log files instead of transmitting all events in real-time to the database for storage. On the Management Server, the bulk loader utility is run from a command line to load bulk log events into the Management Server database where they can then be viewed as events from the **Events** page of the Management Console. The database protects stored events from unauthorized modification and deletion.

The agent processes its log files as follows:

- The agent creates its log files (the SISIDSEvents*.csv and SISIPSRTEvents*.csv files identified above).
- A log file is closed and a new log file is opened based on the agent's log rotation schedule. Rollover of SISIDSEvents.csv and SISIPSRTEvents.csv are controlled by the same parameters, but the rollover decision is made independently for each file.
- Once a SISIDSEvents.csv log file is closed, the file is renamed and then compressed into a .zip file. The renamed file uses the format YYYYMMDD_HHMMSS_YYYY-FT_HOSTNAME, where YYYY is a sequence number, F is the file type, T is the OS type, and HOSTNAME is the agent name, host name, or IP address.
- Log files that are queued to be uploaded for bulk logging are copied to the upload folder in <Install Folder>\Symantec\Data Center Security Server\Agent\sdcsslog\upload.
- If the option to delete log files after processing is disabled, the SISIDSEvents.csv files that were successfully uploaded are copied to the archive folder in <Install Folder>\Symantec\Data Center Security Server\Agent\sdcsslog\archive.

The Intrusion Prevention and Detection function satisfies the following security functional requirements:

- IPD_IDC.1—the TOE can collect IDS data from various monitored resources, based on configured detection rules.
- IPD_IER.1—the TOE provides authorized users with the capability to read all information from the event data. The event data are displayed in a manner suitable for the authorized user to interpret the information.
- IPD_IER.2—the TOE provides capabilities for authorized users to search for events and to filter displayed events based on event type and time span.
- IPD_PBP.1—the TOE can enforce intrusion prevention policies that control how processes access various resources, including other processes, memory, files, registry keys, and network connections.
- IPD_RCT.1—the TOE can generate an event when a breach of a configured prevention or detection policy rule occurs.
- IPD_RCT.2—the TOE can trigger an alert when a policy violation is detected and send a notification to a configured destination.
- IPD_SBD.1—the TOE can perform scans of network traffic and network endpoints deployed in a virtual network infrastructure for matches with signatures defining possible threats.
- IPD_STG.1—the TOE protects stored event data from modification and unauthorized deletion.

7. Rationale

This section provides the rationale for completeness and consistency of the ST. The rationale addresses the following areas:

- Security Objectives
- Security Functional Requirements
- Security Assurance Requirements
- Requirement Dependencies
- TOE Summary Specification.

7.1 Security Objectives Rationale

This section shows that all secure usage assumptions and threats are completely covered by security objectives for the TOE or operational environment. In addition, each objective counters or addresses at least one assumption or threat.

	T. B R U T E _ F O R C E	T. I N T E G R I T Y _ C O M P R O M I S E	T. I N T R U S I O N _ A T T E M P T	T. N E T W O R K _ C O M P R O M I S E	T. N O _ A C C O U N T A B I L I T Y	T. U N A T T E N D E D _ S E S S I O N	T. U N A U T H O R I Z E D _ A C C E S S	T. U N A U T H O R I Z E D _ A C T I V I T Y	T. U N D E T E C T E D _ T H R E A T S	A. M A N A G E	A. P L A T F O R M	A. P R O T E C T
O.AUDIT					X							
O.AUDIT_REVIEW					X							
O.I_AND_A							X					
O.INTRUSION			X						X			
O.PASSWORD_CONTROLS	X											
O.PROTECTED_COMMS				X								
O.RESPONSE									X			
O.REVIEW									X			
O.SECURITY_MANAGEMENT								X				
O.SESSION_TERMINATION						X						
O.STORAGE		X										
OE.PERSONNEL										X		
OE.PHYSICAL												X
OE.PLATFORM							X				X	
OE.TIME					X				X			

Table 8 – Security Problem Definition to Security Objective Correspondence

T.BRUTE_FORCE

An unauthorized user may gain access to the TOE through repeated password-guessing attempts.

This threat is countered by the following security objective:

- O.PASSWORD_CONTROLS—addresses this threat by providing a mechanism that encourages users to choose difficult-to-guess passwords and a mechanism that places a limit on failed authentication attempts.

T.INTEGRITY_COMPROMISE

An unauthorized user may attempt to modify or destroy audit or IDS data, thus removing evidence of unauthorized or malicious activity.

This threat is countered by the following security objective:

- O.STORAGE—addresses this threat by ensuring the TOE is able to protect stored audit records and IDS data from unauthorized modification and deletion.

T.INTRUSION_ATTEMPT

An unauthorized user or process may attempt to perform actions on a host system that could compromise the security of the host system or its resources, or make improper use of system resources.

This threat is countered by the following security objective:

- O.INTRUSION—addresses this threat by ensuring the TOE is able to prevent and detect intrusion attempts on monitored host systems (assets) based on configured prevention policies.

T.NETWORK_COMPROMISE

TSF data communicated between components of the TOE, or between the TOE and external entities, is disclosed or modified.

This threat is countered by the following security objective:

- O.PROTECTED_COMMS—addresses this threat by ensuring communications between components of the TOE and between the TOE and external entities are protected from disclosure and undetected modification.

T.NO_ACCOUNTABILITY

Authorized users of the TOE perform adverse actions on the TOE, or attempt to perform unauthorized actions, which go undetected.

This threat is countered by the following security objectives:

- O.AUDIT—addresses this threat by ensuring the TOE is able to generate audit records of security relevant events.
- O.AUDIT_REVIEW—supports O.AUDIT in addressing the threat by ensuring the TOE provides capabilities for effective review of stored audit records.
- OE.TIME—supports O.AUDIT by ensuring the operational environment is able to provide the TOE with a reliable time source that can be used to generate time stamps for inclusion within generated audit records.

T.UNATTENDED_SESSION

An unauthorized user gains access to the TOE via an unattended authorized user session.

This threat is countered by the following security objectives:

- O.SESSION_TERMINATION—addresses this threat by providing users with a mechanism to terminate their interactive sessions with the TOE, and by ensuring sessions that have been inactive for a configurable period of time will be terminated by the TOE.

T.UNAUTHORIZED_ACCESS

An unauthorized user may gain access to the TOE security functions and data.

This threat is countered by the following security objectives:

- O.I_AND_A—addresses this threat by ensuring all users of the TOE are identified and authenticated prior to gaining further access to the TOE and its services.
- OE.PLATFORM—supports O.I_AND_A by ensuring the operating system underlying each TOE component protects the component and its configuration from unauthorized access.

T.UNAUTHORIZED_ACTIVITY

Authorized users perform unauthorized actions on the TOE.

This threat is countered by the following security objective:

- O.SECURITY_MANAGEMENT—addresses this threat by providing a mechanism that requires authorized users to have appropriate privileges in order to perform actions on the TOE.

T.UNDETECTED_THREATS

Events generated by entities in the IT system indicative of misuse or unauthorized or malicious activity go undetected.

This threat is countered by the following security objectives:

- O.INTRUSION—addresses this threat by ensuring the TOE is able to collect IDS data in order to identify and prevent misuse and unauthorized or malicious activity in the IT system being monitored.
- O.RESPONSE—supports O.INTRUSION in addressing this threat by ensuring the TOE is able to respond to identified misuse and unauthorized or malicious activity.
- O.REVIEW—supports O.INTRUSION in addressing this threat by ensuring the TOE provides capabilities for reviewing the results of its analysis of collected IDS data.
- OE.TIME—supports O.RESPONSE by ensuring the operational environment is able to provide the TOE with a reliable time source that can be used to generate time stamps for inclusion within events.

A.MANAGE

There will be one or more competent individuals assigned to manage the TOE and the security of the information it contains.

This assumption is satisfied by the following security objective:

- OE.PERSONNEL—this objective satisfies the assumption by ensuring those assigned as authorized users are carefully selected and trained in proper operation of the TOE.

A.PLATFORM

The underlying operating system of each TOE component will protect the component and its configuration from unauthorized access.

This assumption is satisfied by the following security objective:

- OE.PLATFORM—this objective satisfies the assumption by ensuring the operating system underlying each TOE component protects the component and its configuration from unauthorized access.

A.PROTECT

The TOE hardware and software critical to the security policy enforcement will be located within controlled access facilities which will prevent unauthorized physical access.

This assumption is satisfied by the following security objective:

- OE.PHYSICAL—this objective satisfies the assumption by ensuring the TOE is protected from physical attack.

7.2 Security Functional Requirements Rationale

All security functional requirements identified in this ST are fully addressed in this section and each is mapped to the objective it is intended to satisfy. Table 5 summarizes the correspondence of functional requirements to TOE security objectives.

	O. A U D I T	O. A U D I T _ R E V I E W	O. I _ A N D _ A	O. I N T R U S I O N	O. P A S S W O R D _ C O N T R O L S	O. P R O T E C T E D _ C O M M S	O. R E S P O N S E	O. R E V I E W	O. S E C U R I T Y _ M A N A G E M E N T	O. S E S S I O N _ T E R M I N A T I O N	O. S T O R A G E
FAU_GEN.1	X										
FAU_SAR.1		X									
FAU_SAR.2		X									
FAU_SAR.3		X									
FAU_STG.2											X
FIA_AFL.1					X						
FIA_ATD.1			X								
FIA_SOS.1					X						
FIA_UAU.2			X								
FIA_UAU.5			X								
FIA_UID.2			X								
FMT_MOF.1								X			
FMT_MTD.1(*)								X			
FMT_SMF.1								X			
FMT_SMR.1								X			
FPT_ITT.1						X					
FTA_SSL.3										X	
FTA_SSL.4										X	
FTP_TRP.1						X					
IPD_IDC.1				X							
IPD_IER.1								X			
IPD_IER.2								X			
IPD_PBP.1				X							
IPD_RCT.1							X				
IPD_RCT.2							X				
IPD_SBD.1				X							
IPD_STG.1											X

Table 9 – Objectives to Requirement Correspondence

O.AUDIT

The TOE shall be able to generate audit records of security-relevant events.

The following security functional requirement contributes to satisfying this security objective:

- FAU_GEN.1—the ST includes FAU_GEN.1 to specify the capability to generate audit records of security-relevant events, and to specify the specific events to be audited and the content of generated audit records of those events.

O.AUDIT_REVIEW

The TOE shall provide a means for the System Administrator to review the audit records generated by the TOE.

The following security functional requirements contribute to satisfying this security objective:

- FAU_SAR.1—the ST includes FAU_SAR.1 to specify the role(s) that is able to read data from stored audit records.
- FAU_SAR.2—the ST supports FAU_SAR.1 by including FAU_SAR.2 to specify that the ability to read data from stored audit records is restricted to only the role specified in FAU_SAR.1.
- FAU_SAR.3—the ST supports FAU_SAR.1 by including FAU_SAR.3 to specify capabilities for searching audit records and for filtering displayed audit records based on audit event type and time span, which assists the authorized roles in effectively reviewing the audit trail.

O.I_AND_A

The TOE shall provide a means for users to be identified and authenticated before gaining access to TOE services.

The following security functional requirements contribute to satisfying this security objective:

- FIA_UID.2, FIA_UAU.2—the ST includes FIA_UID.2 and FIA_UAU.2 to specify that users must be successfully identified and authenticated by the TOE before being able to perform any other TSF-mediated actions.
- FIA_ATD.1—the ST supports FIA_UID.2 and FIA_UAU.2 by including FIA_ATD.1 to ensure the user identity, password, roles, and E-mail address security attributes are associated with individual users.
- FIA_UAU.5—the ST supports FIA_UAU.2 by including FIA_UAU.5 to specify the authentication mechanisms supported by the TOE and the rules by which the TOE authenticates a user's claimed identity.

O.INTRUSION

The TOE shall provide capabilities to prevent and detect intrusion attempts on monitored assets, based on configured prevention and detection policies.

The following security functional requirements contribute to satisfying this security objective:

- IPD_IDC.1—the ST includes IPD_IDC.1 to specify requirements for capabilities to collect IDS data from various monitored resources, based on configured detection rules.
- IPD_PBP.1—the ST includes IPD_PBP.1 to specify requirements for capabilities to enforce intrusion prevention policies that control how processes access various resources, including other processes, memory, files, registry keys, and network connections (IPD_PBP.1.1), to specify requirements for capabilities to govern access and operations amongst controlled subjects and resources (IPD_PBP.1.2), to specify requirements for capabilities to grants access based on security attributes (IPD_PBP.1.3), and to specify requirements for capabilities to deny access to resources based on security attributes (IPD_PBP.1.4).
- IPD_SBD.1—the ST includes IPD_SBD.1 to specify a requirement for capabilities to perform signature-based scans of network traffic and network endpoints in a virtual network infrastructure.

O.PASSWORD_CONTROLS

The TOE must ensure that passwords used to protect TOE access are sufficiently strong, and that a limit is placed on failed authentication attempts.

The following security functional requirement contributes to satisfying this security objective:

- FIA_AFL.1—the ST includes FIA_AFL.1 to specify that an upper limit on unsuccessful password authentication attempts will force the account to be locked out.
- FIA_SOS.1—the ST includes FIA_SOS.1 to specify that passwords must meet minimum construction requirements, in terms of length and character set.

O.PROTECTED_COMMS

The TOE shall protect communications between distributed parts of the TOE, and between the TOE and external entities, from disclosure and modification.

The following security functional requirements contribute to satisfying this security objective:

- FPT_ITT.1—the ST includes FPT_ITT.1 to specify that communications between distributed parts of the TOE will be protected from disclosure and modification.
- FTP_ITC.1, FTP_TRP.1—the ST includes FTP_ITC.1 and FTP_TRP.1 to specify that communications between the TOE and trusted external entities (remote users using the REST API or external IT entities being the AD server) will be protected from disclosure and modification.

O.RESPONSE

The TOE shall respond to misuse and unauthorized or malicious activity it identifies based on its configuration.

The following security functional requirements contribute to satisfying this security objective:

- IPD_RCT.1—the ST includes IPD_RCT.1 to specify the capability to generate an event when a breach of a configured prevention or detection policy rule occurs .
- IPD_RCT.2—the ST includes IPD_RCT.2 to specify the capability to trigger an alert when a policy violation is detected and send a notification to a configured destination.

O.REVIEW

The TOE shall provide capabilities for effective review of stored IDS data.

The following security functional requirements contribute to satisfying this security objective:

- IPD_IER.1—the ST includes IPD_IER.1 to specify the capability for the TOE to provide authorized users with the capability to read all event information from the stored IDS data.
- IPD_IER.2—the ST includes IPD_IER.2 to specify the capability to search for events and to filter displayed events based on event type and time span.

O.SECURITY_MANAGEMENT

The TOE shall restrict the ability to perform security management functions on the TOE to authorized users having appropriate privileges.

The following security functional requirements contribute to satisfying this security objective:

- FMT_SMF.1, FMT_SMR.1, FMT_MOF.1, FMT_MTD.1(*)—the ST includes these requirements to specify the security management functions to be provided by the TOE (FMT_SMF.1), to specify security management roles and privileges (FMT_SMR.1), and to specify the restrictions on management of security function behavior and TSF data (FMT_MOF.1, FMT_MTD.1(*)).

O.SESSION_TERMINATION

The TOE shall provide mechanisms to terminate a user session after a period of inactivity or at the request of the user.

The following security functional requirements contribute to satisfying this security objective:

- FTA_SSL.3—the ST includes FTA_SSL.3 to specify the capability for the TSF to terminate an interactive user session after a period of inactivity.
- FTA_SSL.4—the ST includes FTA_SSL.4 to specify the capability for users to terminate their own interactive sessions.

O.STORAGE

The TOE shall protect stored audit records and IDS data from unauthorized modification or deletion.

The following security functional requirements contribute to satisfying this security objective:

- FAU_STG.2—the ST includes FAU_STG.2 to specify the capability to protect audit data stored in the audit trail from unauthorized deletion and to prevent unauthorized modification of the audit data.
- IPD_STG.1—the ST includes IPD_STG.1 to specify the capability to protect stored IDS data from modification and unauthorized deletion.

7.3 Security Assurance Requirements Rationale

EAL 2 was selected as the assurance level because the TOE is a commercial product whose users require a low to moderate level of independently assured security. The TOE is intended for use in an environment with good physical access security where it is assumed that attackers will have Basic attack potential. The target assurance level of EAL 2 is appropriate for such an environment. Augmentation was chosen to provide the added assurance that is gained by defining flaw remediation procedures. Therefore, the target assurance level of EAL 2 augmented with ALC_FLR.1 is appropriate for such an environment.

7.4 Requirement Dependency Rationale

The following table identifies the SFRs claimed in the ST, their dependencies as defined in CC Part 2 or the extended components definition (Section 5.1 of this ST), and how the dependency is satisfied in the ST. It can be seen that all dependencies have been satisfied, either by inclusion in the ST of the appropriate dependent SFRs, or by functionality provided by the operational environment.

Requirement	Dependencies	How Satisfied
FAU_GEN.1	FPT_STM.1	See Timestamp Note below.
FAU_SAR.1	FAU_GEN.1	FAU_GEN.1
FAU_SAR.2	FAU_SAR.1	FAU_SAR.1
FAU_SAR.3	FAU_SAR.1	FAU_SAR.1
FAU_STG.2	FAU_GEN.1	FAU_GEN.1
FIA_AFL.1	FIA_UAU.1	FIA_UAU.2 (hierarchical to FIA_UAU.1)
FIA_ATD.1	None	None
FIA_SOS.1	None	None
FIA_UAU.2	FIA_UID.1	FIA_UID.2 (hierarchical to FIA_UID.1)
FIA_UAU.5	None	None
FIA_UID.2	None	None
FMT_MOF.1	FMT_SMR.1, FMT_SMF.1	FMT_SMR.1, FMT_SMF.1
FMT_MTD.1(*)	FMT_SMR.1, FMT_SMF.1	FMT_SMR.1, FMT_SMF.1
FMT_SMF.1	None	None
FMT_SMR.1	FIA_UID.1	FIA_UID.2 (hierarchical to FIA_UID.1)

Requirement	Dependencies	How Satisfied
FPT_ITT.1	None	None
FTA_SSL.3	FMT_SMR.1	FMT_SMR.1
FTA_SSL.4	None	None
FTP_TRP.1	None	None
IPD_IDC.1	None	None
IPD_IER.1	IPD_RCT.1	IPD_RCT.1
IPD_IER.2	IPD_IER.1	IPD_IER.1
IPD_PBP.1	None	None
IPD_RCT.1	IPD_IDC.1 or IPD_PBP.1 or IPD_SBD.1	IPD_IDC.1, IPD_PBP.1, IPD_SBD.1
IPD_RCT.2	IPD_RCT.1	IPD_RCT.1
IPD_SBD.1	None	None
IPD_STG.1	IPD_RCT.1	IPD_RCT.1

Table 10 – Requirement Dependencies

Timestamp Note: The TOE is not a physical device and operates as an application within a process provided by the environment. Thus, the environment is providing resources for the TOE. The environmental objective OE.TIME requires that the TOE's environment provide a reliable timestamp which the TOE can use as needed (e.g., within audit records). Therefore, the functionality specified in the dependency of FAU_GEN.1 upon FPT_STM.1 is available to the TOE from its environment.

7.5 TOE Summary Specification Rationale

Section 6, the TOE Summary Specification, describes how the security functions of the TOE meet the claimed SFRs. The following table provides a mapping of the SFRs to the security function descriptions to support the TOE Summary Specification.

	Security Audit	I&A	Security Management	Protection of the TSF	TOE Access	Trusted Path	Intrusion Prevention and Detection
FAU_GEN.1	X						
FAU_SAR.1	X						
FAU_SAR.2	X						
FAU_SAR.3	X						
FAU_STG.2	X						
FIA_AFL.1		X					
FIA_ATD.1		X					
FIA_SOS.1		X					
FIA_UAU.2		X					
FIA_UAU.5		X					
FIA_UID.2		X					
FMT_MOF.1			X				
FMT_MTD.1(*)			X				
FMT_SMF.1			X				
FMT_SMR.1			X				
FPT_ITT.1				X			
FTA_SSL.3					X		
FTA_SSL.4					X		
FTP_TRP.1						X	
IPD_IDC.1							X
IPD_IER.1							X
IPD_IER.2							X

	Security Audit	I&A	Security Management	Protection of the TSF	TOE Access	Trusted Path	Intrusion Prevention and Detection
IPD_PBP.1							X
IPD_RCT.1							X
IPD_RCT.2							X
IPD_SBD.1							X
IPD_STG.1							X

Table 11 – Security Functions vs. Requirements Mapping