NFR Security, Inc.
NFR® Sentivist™ v4.0.2 – Updated to v4.0.6
and
Sentivist Sensors 310C, 320C and 320 F Models

Security Target

Version 2.7

April 18, 2005

Prepared for:

NFR® Security, Inc.
5 Choke Cherry Road
Suite 200
Rockville, MD 20850
(240) 632-9000

Prepared by:

Corsec Security, Inc.
10340 Democracy Lane, Suite 201
Fairfax, VA 22030
(703) 267-6050

Table of Contents

List of Figures

List of Tables

1.0 SECURITY TARGET INTRODUCTION

The Security Target (ST) introduction section presents introductory information on the NFR® Sentivist™ v4.0.2 – Updated to v4.0.6 (Sentivist™) and Sentivist Sensors 310C, 320C and 320F Models, Security Target; the Target of Evaluation (TOE) referenced in this Security Target, and a basic introduction to the TOE.

1.1 ST and TOE Identification

This section will provide information necessary to identify and control the Security Target and the TOE.

ST Title	NFR Sentivist™ v4.0.2 – Updated to v4.0.6, Sentivist Sensors 310C, 320C and 320F Models Security Target version 2.7 April 18, 2005
ST Author	Corsec Security, Inc.
TOE Identification	NFR Sentivist™ v4.0.2 – Updated to v4.0.6 and Sentivist Sensor Models 310C, 320C and 320F
CC Identification	Common Criteria for Information Technology Security Evaluation (CC), Version 2.1, August 1999 (aligned with ISO 15408). All International Common Criteria Interpretations through August 26, 2003 have been applied.
PP Identification	Intrusion Detection System System Protection Profile, Version 1.4 – February 4, 2002
Assurance Level	Evaluation Assurance Level 2
Keywords	Intrusion detection system (IDS), vulnerability assessor, network-based IDS, Signature analysis, security target

1.2 Security Target Overview

This Security Target describes the requirements for the NFR Sentivist™ v4.0.2 – Updated to v4.0.6, Sentivist Sensor Models 310C, 320C, 320F and specifies how the TOE meets those requirements. This specific ST is based on the Intrusion Detection System Protection Profile, Version 1.4 – February 4, 2002 issued by the National Security Agency (NSA). This document is the Security Target for NFR Sentivist™ Version 4.0.2 – Updated to v4.0.6.

The Target of Evaluation is the NFR Security, Inc., Sentivist™ Version 4.0.2 – Updated to v4.0.6 and Sentivist Sensor Models 310C, 320C and 320F (referred to as either "the TOE", the "Sentivist™ v4.0.2 – Updated to v4.0.6", or the "Sentivist™").

Sentivist™ v4.0.2 – Updated to v4.0.6 from NFR Security is a network intrusion detection system (IDS) consisting of three core components: ***Sentivist™ Sensor, Sentivist™ Server, and Sentivist™ Administration Interface***. It monitors network traffic in real time for suspicious activity misuse, abuse, attacks, anomalous behavior and previously undiscovered attacks. NFR Sentivist™ v4.0.2 – Updated to v4.0.6 provides attack detection while attempting to minimize false positives. The system includes signature analysis, customization and active response capabilities and provides central management of distributed environments.

The Sentivist™ adheres to the signature analysis method. That is, it matches specific signatures or patterns that may characterize attack attempts to a knowledge base of known attacks. This knowledge base can be updated and user customized to provide up to date coverage of known attacks.

This ST contains the following sections to provide mapping of the Security Environment to the Security Requirements that the Sentivist™ v4.0.2 – Updated to v4.0.6 product meets in order to mitigate the defined threats:

- **Security Target Introduction (Section 1):** Provides an overview of the ST.
- **NFR Sentivist™ Product Introduction (Section 2):** Provides an introduction to the NFR Sentivist™ Product.
- **TOE Description (Section 3)**: Provides an overview of the TOE security functions and describes the physical and logical boundaries for the TOE Security.
- **TOE Security Environment (Section 4)**: Describes the threats, organizational security policies, and assumptions that pertain to the TOE and the TOE environment.
- **TOE Security Objectives (Section 5)**: Identifies the security objectives that are satisfied by the TOE and the TOE environment.
- **TOE Environment IT Security Requirements (Section 6)**: Presents the Security Functional Requirements (SFRs) met by the TOE Environment
- **TOE Assurance Requirements (Section 7)**: Presents the Security Assurance Requirements (SARs) met by the TOE
- **TOE Summary Specification (Section 8)**: Describes the security functions provided by the TOE to satisfy the security requirements and objectives
- **Protection Profile Claims (Section 9)**: Presents the rationale concerning compliance of the ST with the Intrusion Detection System System Protection Profile.
- **Rationale (Section 10)**: Presents the rationale for the security objectives, requirements, and the TOE summary specifications as to their consistency, completeness, and suitability.
- **Glossary of Terms (Section 11)**: This section provides a glossary of terms used throughout this ST.

1.3 Common Criteria Conformance Claims

This Target of Evaluation is:

1) CC Version 2.1 Part 2 Extended
2) CC Version 2.1 Part 3-conformant

Additionally, the TOE claims conformance to the Evaluation Assurance Level 2 package.

1.4 Conventions and Terminology

Conventions:

There are several font variations within this ST. The conventions used in this ST are consistent with those used in the Protection Profile to which the TOE claims

conformance. Selected presentation choices are discussed here to aid the Protection Profile user.

The CC allows several operations to be performed on security requirements; *refinement*, *selection* and *assignment* are defined in paragraph 2.1.4 of Part 2 of the CC. Each of these operations is used in this ST.

- Assignment: Allows the specification of an identified parameter. Indicated with **bold text.**
- Refinement: Allows the addition of details. Indicated with ***bold text and italics.***
- Selection: The **selection** operation is used to select one or more options provided by the CC in stating a requirement. Selections are denoted by underlined text.
- Iteration: Allows a component to be used more than once with varying operations. (Not used in this ST)
- Operations that the PP author left for the ST author to complete are denoted with the above mentioned conventions but in parenthesis.

Terminology:

In the Common Criteria, many terms are defined in Section 2.3 of Part 1. The following are a subset of those definitions. They are listed here to aid the user of the ST.

User – Any entity (human user or external IT entity) outside the TOE that interacts with the TOE.

Human user – Any person who interacts with the TOE.

External IT entity – Any IT product or system, un-trusted or trusted, outside of the TOE that interacts with the TOE.

Role – A predefined set of rules establishing the allowed interactions between a user and the TOE.

Identity – A representation (e.g., a string) uniquely identifying an authorized user, which can be either the full or abbreviated name of that user or a pseudonym.

Authentication data – Information used to verify the claimed identity of a user.

From the above definitions given by the Common Criteria, the following terms can be derived:

Authorized external IT entity – Any IT product or system, outside the scope of the TOE that may administer the security parameters of the TOE. Such entities are not subject to any access control requirements once authenticated to the TOE and are therefore trusted to not compromise the security policy enforced by the TOE.

Authorized Administrator – A role which human users may be associated with to administer the security parameters of the TOE. Such users are not subject to any access control requirements once authenticated to the TOE and are therefore trusted to not compromise the security policy enforced by the TOE.

Additional terms and abbreviations are used throughout the body of this ST. They are listed in Section 11 to aid the user of the ST.

2.0 Introduction to the NFR Sentivist™

NFR Sentivist™ is an intrusion detection system. The system supports today's high-speed networks, includes customization of packages and provides central management of distributed environments. The NFR Sentivist™ monitors the network in real-time, raises alerts when attacks or misuse are detected, and actively responds if configured to do so. These include alert generation, data recording, resetting of TCP Session and changing of firewall rules[1]. The TCP reset functionality is implemented as an n-Code/signature language built-in function named reset. If reset is called from one of the signatures TCP resets are sent out by the monitoring interface to both end points of the connection. The three Sentivist™ Sensors which are a part of this evaluation are available for gigabit and full duplex 100 Mbps Ethernet networks. Available as pre-configured appliances, Sentivist™ Sensors can be operational immediately as long as there is a Sentivist Server operational and available. Multiple Sentivist™ Sensors can be managed from a central location, and their information consolidated for inquiry and reporting.

Sentivist™ works on the principles of detection, identification and response based on the signatures stored on the Sentivist's knowledge base. When a major new exploit is found, NFR Security customers with maintenance contracts are notified immediately and provided with new signatures as necessary so that they can update and protect their systems.

Cryptographic mechanisms are in place in the Sentivist to ensure confidentiality and integrity of all data transmitted. The TOE uses a NFR proprietary application protocol data exchange to ensure confidentiality of the data channel between the Sentivist™ Administration Interface, Sentivist™ Server and the Sentivist™ Sensor. The transmitted data is encrypted using AES to ensure confidentiality of the transmitted data.

An NFR Sentivist™ system comprises of:

Sentivist™ Sensors: The Sensors monitor network traffic. Three models are available:

- **Sentivist™ Sensor 310C:** This Sensor has two 10/100/1000 Ethernet interfaces (em0 or em1); one of these two NIC's is used to manage the Sentivist™ Sensor. The remaining NIC is used to monitor network traffic.
- **Sentivist™ Sensor 320C:** This Sensor has three 10/100/1000 Ethernet interfaces (em0, em1 and em2). One Ethernet NIC is used to manage the Sentivist™ Sensor. The second NIC is used for monitoring network traffic. The remaining third NIC can be used to monitor a backup circuit.
- **Sentivist™ Sensor 320F:** This Sensor has two 10/100/1000 Mbps copper Ethernet NICs (em0 and em1) and two Gigabit Ethernet fiber NICs (sk0 and sk1). The 10/100/1000 Mbps Ethernet NICs are used to manage the Sentivist™ Sensor. The Gigabit Ethernet NIC is used for monitoring network traffic. The remaining Gigabit Ethernet NIC can be used to monitor a backup circuit.

The primary difference among the models is the network throughput they are designed to monitor. All Sentivist™ Sensors are delivered as pre-configured appliances eliminating the need to purchase and configure hardware. NFR Security continually tests and certifies hardware configurations that can support Sentivist™ Sensor. A current listing of

[1] The changing of firewall rules response mechanism is not a part of the evaluated configuration of the TOE, Refer to Section 1.5 for the list of features not included in this evaluation.

supported hardware configurations is given in Section 3.2. Sentivist™ Sensor is also available as a software-only version for those customers who wish to source their own hardware that matches NFR certified configurations (Software only version of the Sentivist Sensor is not a part of the evaluated configuration of the TOE). Sentivist™ Sensors can be placed anywhere on the network, monitoring traffic coming to or through the firewall, crossing from one subnet to another, or accessing particular IT resources such as email servers, web servers, database servers, etc.

Sentivist™ Administration Interface: It provides a Windows based interface to the Sentivist™ Server for administration of distributed Sentivist™ Sensors, managing, viewing alerts, events generated by the Sensors.

Sentivist™ Server: Its main function is the management of multiple Sentivist™ Sensors. Sentivist™ Server provides central administration and storage of alert and event data for the sensors that report into it. Multiple servers can be distributed throughout the network as needed.

3.0 TOE DESCRIPTION

The TOE description provides context for the evaluation. It describes the TOE as an aid to understanding the security requirements for the TOE.

3.1 NFR Sentivist™ v4.0.2 – Updated to v4.0.6

Sentivist™ 4.0.2 – Updated to v4.0.6 from NFR Security is a network intrusion detection system (IDS) that monitors traffic in real time for suspicious activity misuse, abuse, attacks, anomalous behavior and previously undiscovered attacks. Previously undiscovered attacks are defined as attacks for which there are no known signatures in the Sentivist™'s knowledge base. The Sentivist™ employs protocol analysis and anomaly detection techniques to determine previously undiscovered attacks, checks are made to capture unexpected activity and deviations from standards specified in the RFCs[2]. Sentivist™ has a knowledge base of the various ways vulnerabilities can be exploited and how protocols should behave. Sentivist™ examines activity against these rather than a simple pattern of a known exploit.

The Sentivist™ adheres to the signature analysis method. That is, it matches specific signatures or patterns that may characterize attack attempts to a knowledge base of known attacks. This knowledge base can be updated and user customized to provide up to date coverage of known attacks.

Sentivist™ Sensors are placed at critical points throughout the network in order to detect potential attacks, attacks in progress, and attacks that have completed and to collect information on previously unidentified attacks that have evaded identification (and detection as a specific attack) because of their newness.

Sentivist™ consists of three components that make up the system. The three components are:

- **Sentivist™ Sensor;**
- **Sentivist™ Server; and**
- **Sentivist™ Administration Interface (SAI).**

[2] For the complete list of RFCs that are use for protocol analysis and anomaly detection in the Sentivist please refer to Appendix A.

Sentivist™ Sensor (Sensor):
Each Sentivist™ deployment will at least have one or more Sentivist™ Sensors, one Server and Administration Interface. The Sentivist™ Sensor is delivered as a pre-configured appliance consisting of the appropriate hardware and software combination for each Sensor model.

The hardware for the Sentivist™ Sensor is NFR's proprietary hardware. Sensor hardware configurations are a combination of commercially available hardware modules from various vendors. NFR Sensor hardware configuration for the various Sensor models is given in section 3.2: Physical Scope and Boundary of the TOE.

Three Sentivist Sensor models are included in this evaluation effort, Sentivist Sensor 310C, Sentivist Sensor 320C and Sentivist Sensor 320F respectively. The configuration of these three models is clearly laid out under Section 3.2: Physical Scope and Boundary of the TOE. The primary difference between these three models is the network throughput that they are designed to monitor.

The Sentivist Sensor 310C model is designed to monitor 100Mbps or less network environments. The Sentivist Sensors 320C and 320F models are designed to monitor >100Mbps/Gigabit network environments. The key discriminator between the 320C and 320F models is the PCI addin card/NIC, copper versus fiber.

The Sentivist™ Sensor software, which includes an embedded operating system (FreeBSD 4.8 and custom NFR code), runs from the product distribution CD. The Sentivist™ Sensor CD has the kernel of the operating system on it along with the various operating system applications that NFR needs. This eliminates the task of installing software or reconfiguring the operating system.

The Sentivist™ Sensors can be deployed to remote locations, by inserting the Sentivist Sensor CD and configuration information into the Sentivist Sensor hardware. Sensor Administrators login to the system and complete the installation process. Sentivist™ Sensors can be up and running within minutes if the above mentioned procedure is used with a Sentivist Server already up and running.

The Sentivist™ Sensor is delivered as a pre-configured appliance using default configuration values. In order for it to operate within the customer's Sentivist™ system, the default values need to be replaced with values specific to the organization where the Sensor will be deployed. This is accomplished using the Installation and Configuration mode of the Sentivist Sensor.

Installation and Configuration Mode:
This mode involves the initial installation and configuration of the Sentivist Server, Sentivist Administration Interface and Sentivist Sensor. As the initial configuration of the Sensor cannot be done remotely from the Sentivist Server, the configuration values can be either manually overwritten or a configuration file written onto a floppy disk can be used. Manual configuration can be accomplished by editing the default configuration values through the Sentivist™ Sensor Management Menu. Sentivist Sensor default configuration values can be replaced with values contained in a file on a floppy disk. This disk is referred to as the "configuration floppy". When powered on, Sentivist Sensor detects a valid installed Sentivist Sensor version, and knows to read the configuration values from the configuration floppy. The configuration information stored on the floppy is administered by the Sensor Administrator by logging into the Sentivist Sensor Management console. The floppy drive on each of the Sensor models comes with a lockable bezel (Black locking front 1U Bezel) which protects the floppy disk drive from unauthorized use. It is extremely important that the floppy drive is locked and secured

using the bezel at all times by the Sensor Administrator. By following this practice, the possibility of changing the configuration parameters of the Sensor by an unauthorized/malicious entity is significantly reduced. Thus it largely reduces the risk of a physical compromise of the Sensor by a malicious entity. Addition of new Sensors after the initial installation is also covered by the Installation and Configuration Mode.

The Sentivist Administration Interface is installed on a Windows machine that is local-login only, not a part of any domains, and is used only for communicating with the Sentivist Server. Instructions for configuring the Windows firewall so that it only allows traffic with the Server can be found in the Common Criteria Wrapper Guide.

The Installation and Configuration Mode also covers the initial setup of the Sentivist Server. The installation of the underlying operating system, system hardening, and configuration of users and audit functions are all part of the installation and configuration mode. The installation and configuration mode is not a part of the evaluated configuration of the TOE. Therefore, the TOE will be out of the evaluated configuration any time the root user logs into the Sentivist Server to modify or configure any system parameters. This includes configuration modifications to existing Sensors, and the addition of new Sensors.

Operation Mode:
Periodic Sentivist Sensor maintenance is conducted from the Sentivist Server CLI by the authorized system administrator/nfr. This account is created during the installation of the Sentivist Server. The `nfr` account on the Sentivist Server is allowed to read the various system logs on the Server. Instructions regarding the installation and configuration of `sudo` can be found in the Common Criteria Wrapper Guide v1.7. The operation mode is a part of the evaluated configuration of the TOE.

The Sentivist™ Sensor consists of a hardware appliance running the sensor software from a CD. The primary hardware components of the Sensor hardware are an Intel Xeon processor, an internal hard disk drive, DDR RAM, a CD ROM drive, a floppy drive, 10/100/1000 Mbps Ethernet NIC's and Gigabit Ethernet fiber NICs. There is a UNIX filesystem on the internal disk of the Sensor. It serves the following purposes:

- It serves as a persistent storage/spooling device for information (alerts, event, and status data) that has not yet transited to the Sentivist Server.
- Persistent storage of policy information that is "mirrored" to the Sensor from the Sentivist Server.

During the installation of the Sentivist Sensor a limited number of executables are copied from the Sentivist Sensor CD to the internal disk. These executables are run from/paged from the disk. The operating system and the executables that are not copied are run from the Sentivist Sensor CD.

The Sentivist™ Sensor monitors packets on the network to discover whether an intruder is attempting to break into a network or computer system. The Sentivist Sensor can be distributed at key points throughout the network that it is configured to monitor.

The Sentivist™ Sensor monitors networks in real time for activity such as known attacks, abnormal behavior, unauthorized access attempts, and policy infringements. Sensor records information associated with suspicious activity and raises applicable alerts. Configurable response mechanisms determine what action to take when Sentivist™ Sensor detects an intrusion. These include alert generation, data recording, resetting of

TCP session (when configured to do so) and changing firewall rules. The TOE allows modification of the parameters of an external firewall[3]. The external firewall and the configuration of the firewall rules are features that are not a part of the evaluated configuration of the TOE and hence outside the scope of this evaluation.

The Sentivist™ Sensor architecture separates monitoring activities from management activities. This allows the monitoring NIC to operate in stealth mode, hence other sniffing or active probing software cannot detect it. Therefore it will not respond to any network traffic or requests from any service on the monitored network.

Sentivist™ Server (Server):
The Sentivist™ Server processes, and manages large volumes of event and alert data provided from the Sentivist™ Sensors. It provides a management interface to this data for applications such as the Sentivist™ Administration Interface.

The Sentivist™ Server provides services for storing and managing event, alert, and status data from the Sensor appliances.

The following are brief descriptions of each main function area within the Sentivist™ Server:

- **Data Reception and Management:** Receives data from multiple Sensors. Data includes events, alerts, and status. The data is processed, stored to a local data store, and made available to external management applications.
- **Configuration Management:** Provides an agent and command line interfaces for configuration. The management agent listens for requests from management clients (SAI) and attempts to carry out their requested management commands. The Management Agent is a web-server like component on the Sentivist Server. It listens for and services the Management Client's requests via the Management interface. The management agent is the executable guisrv and a set of CGI commands. This includes configuration of Sensor appliances and NFR Administration users communicating with the Server.
- **Fault Management:** Provides alert notification to a standard communications channel or an agent for queries on the stored alert data.

The Sentivist™ Server provides configuration information (e.g. which packages and backends are enabled) to the Sentivist™ Sensor. Packages and backends are configuration information which can be enabled or disabled based on what type of traffic flow in and out of the network, needs to be monitored by the Sensor. Packages and backends can be enabled/disabled from the Sentivist™ Administration Interface. Once the configuration information is changed the packages and backends are pushed out to the corresponding Sensors, the Sentivist™ Server to Sensor communication process is called "mirroring". The Server initiates a connection to the Sensor in order to "mirror-out" or push packages down to the appliance. The mirror process is a series of getserver/put commands. When all the packages are pushed, getserver[4] initiates package synchronization on the Sensor and the changes are recompiled and loaded.

Packages are used to group backends of similar functionality or, in some cases, the same origin. A backend is an n-code filter and the supporting configuration files. A package typically contains several backends and provides them with common configuration options as well as a name space that allows them to access shared variables.

[3] The external firewall and changing of firewall rules response mechanisms are not a part of the evaluated configuration of the TOE, Refer to Section 1.5 for the list of features not included in this evaluation.
[4] getserver is a communication protocol used by the Sentivist. It provides constant connection between the Sensor and the Server.

The Package screens in the Sentivist™ Administration Interface allow an authorized user to examine package and backend properties and query the data collected by each backend.

The Sentivist™ Server receives, processes, and manages (consolidates data for querying and reporting) large volumes of alert data generated by the Sentivist™ Sensors. It also provides a centralized facility for Sentivist™ Sensor management and for viewing captured data. The Sentivist Server is the primary storage device for Sensor generated data. Its storage goal is short-term, and if administrators require long term data storage they need to off-load the data to a larger storage facility such as an external MySQL database.

The key functions of the Sentivist™ Server are:

- Short term storage of alert and event data
- Centralized management of Sentivist™ Sensor configuration data
- Support for alert and event data queries using Sentivist™ Administration Interface
- Tuning filters for recording or ignoring alert and event data from Sentivist™ Sensor
- Tuning filters for allowing or preventing users from querying and viewing recorded alert and event data
- Integration with external applications for alert notification
- Audit of actions and events affecting the Sentivist™ system state and configuration

This is the management center of the Sentivist™. It interfaces with both the Sentivist™ Administration Interface and the various Sentivist™ Sensors. Each Sensor reports to a specified Server. A single Server may support multiple Sensors, and an optional backup Server can also be specified for each Sensor. In the event that the primary Server fails, the Sensor will report to the backup Server[5]. Sentivist™ Server software runs on both Linux and Solaris operating systems. The Server is the central repository for access control and user authentication information.

Various communication paths that could exist between the inter TOE components can be clearly seen in the figure below (next page).

[5] The backup server functionality is a feature that is not a part of the evaluated configuration of the TOE.

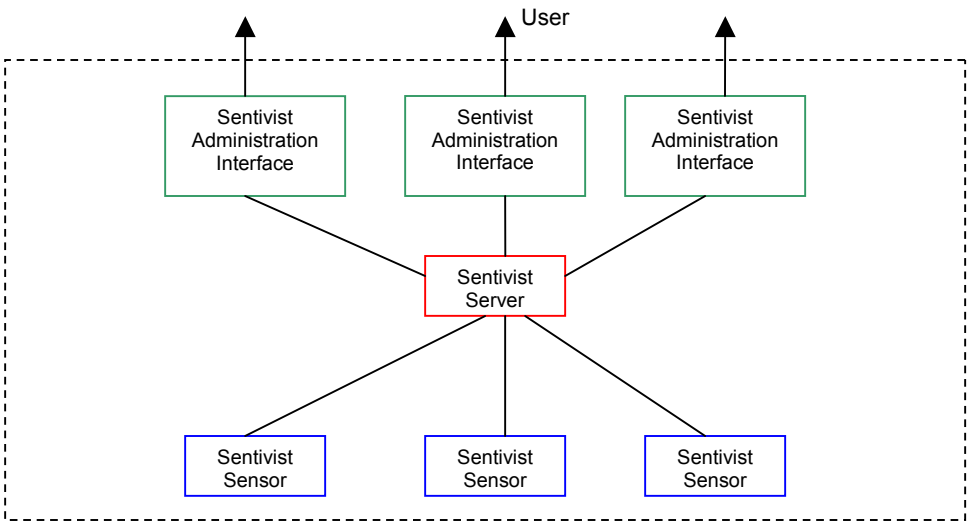

Figure 1: NFR Sentivist™ Interfaces

Sentivist™ Administration Interface (SAI): Sentivist™ Administration Interface (SAI) provides a Microsoft Windows-based interface to Sentivist™ Server for Sentivist™ Sensor configuration, viewing and managing Sensor alerts, and performing administration tasks. A single SAI may manage multiple Sensors; multiple SAIs may manage one or more Sensors. The Sentivist Server is the central repository for all the identification and authentication mechanisms of the Sentivist Administration Interface. Authorized personnel wishing to interact with the Sentivist Server via the Sentivist Administration Interface are provided with a single point of authentication by the SAI.

3.2 Sentivist Features Excluded from this Common Criteria Evaluation

The following features have been excluded from the Common Criteria Evaluated Configuration of the Sentivist.

1. Command Line Query provides an interface to Sentivist Server from any machine running Linux or Solaris that has network connectivity to a Sentivist Server. This tool is prepackaged with the Sentivist Server product distribution CD and its installation is detailed in Appendix E – Installing and Using Command Line Query, NFR Sentivist Getting Started Guide. This feature of the Sentivist is not a part of the evaluated configuration of the TOE and hence should not be used in the Common Criteria Evaluated Configuration of the Sentivist.
2. The Alert Continuity Installation Option of the Sentivist by installing a Backup Sentivist Server as detailed on Page 3 – Sentivist Sensor Installation Options, in the NFR Sentivist Getting Started Guide should not be used under the Common Criteria Evaluated Configuration of the Sentivist.
3. Sentivist Enterprise Console Version 4.0 provided along with NFR Sentivist Version 4.0.2 – Updated to v4.0.6 is a feature that is not a part of the evaluated configuration of the Sentivist.

4. Sentivist DBExport feature is not a part of the Common Criteria Evaluated Configuration of the Sentivist.
5. SAM (Suspicious Activity Monitoring) Client Integration with Checkpoint VPN-1/Firewall-1 and firewall updates feature is not a part of the Common Criteria Evaluated Configuration
6. NFR Plus for Tivoli SecureWay Risk Manager Feature is not a part of the Common Criteria Evaluated Configuration of the TOE.
7. NFR Integration Module for HP OpenView Operations is a not a part of the Common Criteria Evaluated Configuration of the TOE.
8. Ethereal – Network Protocol Analyzer V 0.9.13a which comes with the NFR Sentivist V4.0.2 – Updated to v4.0.6 is not a part of the Common Criteria Evaluated Configuration of the TOE.
9. WinPcap 3.0 – Packet Capture Utility which comes with the NFR Sentivist V4.0.2 – Updated to v4.0.6 is not a part of the Common Criteria Evaluated Configuration of the TOE
10. The sha1sum program used to create an encrypted password for the Sentivist Sensor Administrator.
11. Software only version of the Sentivist Sensor is not a part of the evaluated configuration of the TOE
12. The Installation and Configuration Mode of the Sentivist Sensor as detailed in Section 3.1 of this document is not a part of the evaluated configuration of the TOE.

3.3 Physical Scope and Boundaries of the TOE

The NFR Sentivist architecture consists of three physically distinct components namely the Sentivist™ Administration Interface, Sentivist™ Server and the Sentivist™ Sensor. The figure given below illustrates the physical scope and the physical boundary of the TOE.

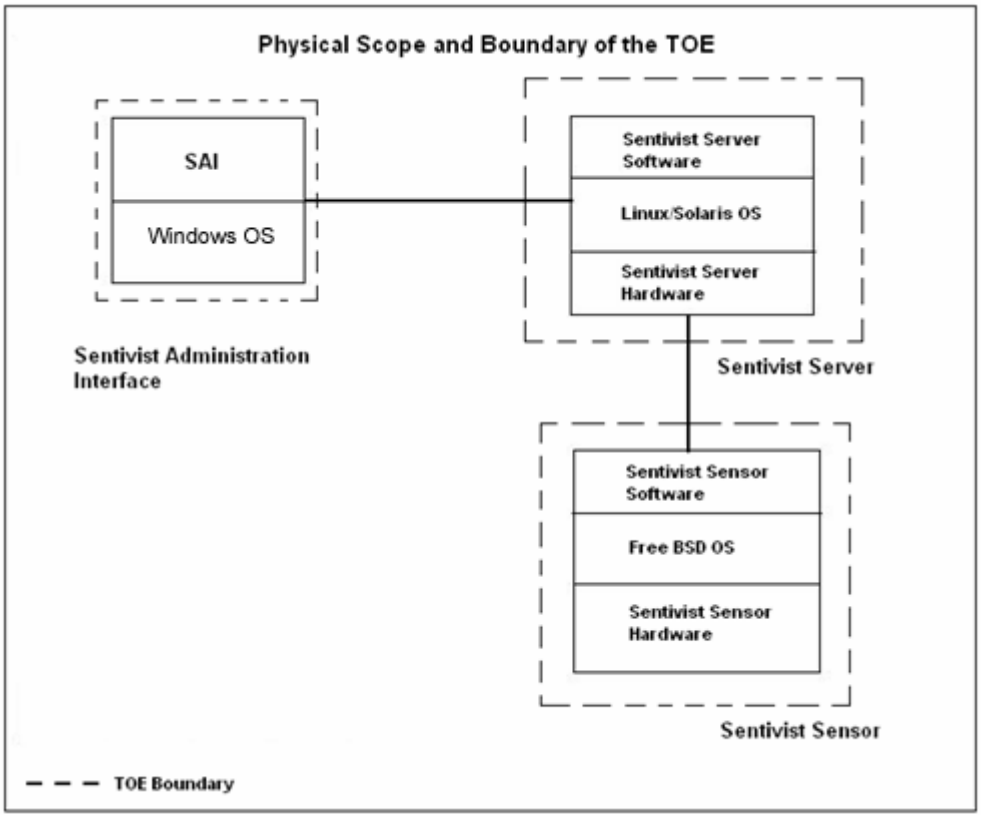

Figure 2: Physical Scope and Boundary of the TOE

The evaluated configuration of the NFR Sentivist at a minimum comprises of one Sentivist Server, one Sentivist Sensor and one Sentivist Administration Interface. The figure given below depicts the basic Sentivist deployment.

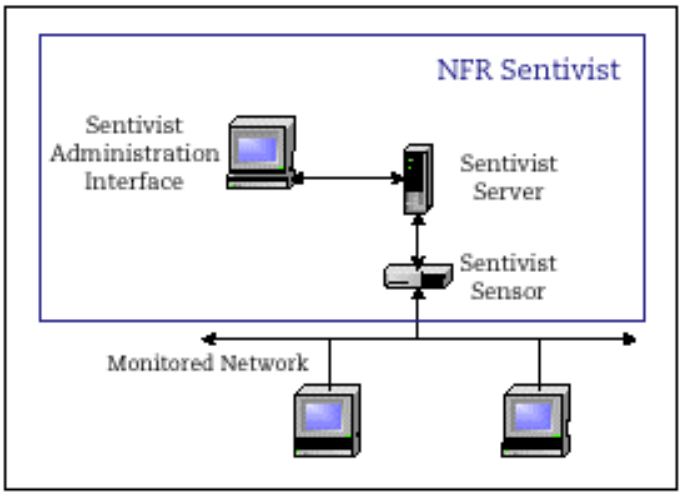

Figure 3: Typical System Environment

Below is the comprehensive list of the minimum and recommended (hardware and software) configuration of the various components of the TOE. The Sentivist™ components run on the following hardware platforms and operating systems:

Sentivist™ Administration Interface:
The Sentivist™ Administration Interface is required to operate under a subset of Windows operating environments. These environments are as follows:

- Intel platform running the following operating systems:
  - Windows 2000 Professional with Service Pack 2
  - Windows XP with Service Pack 1

Minimum Hardware:
The Sentivist™ Administration Interface will run on the following minimum set of hardware:

- Intel 800 MHz Pentium
- 512 MB of RAM
- 100 Mbps network connection
- 1+ GB usable disk drive

Sentivist™ Server:
The Sentivist™ Server is required to operate under a subset of UNIX operating environment. These operating environments are defined as follows.

- Intel platform running the following operating systems:
  - Red Hat Linux 7.3.
  - Red Hat Linux 8.0.
- Sun SPARC platform running the following operating systems:
  - Solaris 8
  - Solaris 9

Minimum Hardware:
The Sentivist™ Server must perform to an acceptable level on the following minimum sets of hardware:

- Red Hat Linux 7.3 or 8.0 running on the following hardware:
  - 1.4 GHz Pentium class machine
  - 1GB of RAM
  - Hard Disk Drive: Multiple SCSI drives in RAID 0+1, 1, or 5 configuration (10GB usable)
  - 100 Mbps network connection
- Solaris 8 or 9 running on the following hardware:
  - 400 MHz UltraSparc CPU
  - 2 GB of RAM
  - Multiple SCSI drives in RAID 1, 0+1, or 5 configuration (10GB usable)
  - 100 Mbps network connection

Recommended Hardware:
The Sentivist™ Server must outperform minimum hardware performance on the following recommended hardware:

- Red Hat Linux 7.3 or 8.0 running on the following hardware:
  - 1.4 GHz Pentium class machine (must be dual CPU's)
  - 2 GB of RAM

  - Multiple SCSI drives in a RAID 1, 0+1, 4 or 5 configuration (40 GB usable)
  - 100 Mbps network connection
  - Hot spare is also recommended
- Solaris 8 or 9 running on the following hardware:
  - Dual 400 MHz UltraSparc CPU (must be dual CPUs)
  - 4 GB of RAM
  - Multiple SCSI drives in RAID 1, 0+1,or 5 configuration (40GB usable)
  - 100 Mbps network connection
  - Hot spare is also recommended

Sentivist™ Sensor:

- The Sentivist™ Sensor appliance is required to operate under the Free BSD 4.8 operating system with the hardware configuration referenced below.
- The Sensor CD has the Free BSD 4.8 operating system and custom NFR code resident on it. The Free BSD operating system has been stripped to remove all that is not used to build the Sensor functionality.

The hardware specifications for the various Sentivist™ Sensor models that are part of this evaluation are:

- Sentivist™ Sensor 310C
- Sentivist™ Sensor 320C
- Sentivist™ Sensor 320F

Sentivist™ Sensor 310C Hardware:

- 1U Rack mountable Chassis for Intel WV533 board - 350W Power Supply, 2 x 1" ATA Cold-swap OR 3 x
- 1" U320 Hot-Swap HDD
- WV533 SCSI board for Dual Xeon processors w/512k cache, 400 or 533MHz bus
- WV533 SCSI 1U Backplane
- Intel® Xeon™ processor 2.8GHz, with 533MHz FSB & 512k cache, 1U Heat-sink variant
- Black locking front 1U Bezel 1
- Slimline CD/Floppy combo kit.
- 36 GB Ultra 320 15K RPM Cheetah LP 1" with SCA Connector
- 1GB DDR266 Low Profile Reg ECC RAM (requires 2 matching pieces) - 2GBs total
- RJ-45 to DB9 Serial Console adapter Cable
- US Power Cord, PS/2 "Y-cable" (included with original Intel packaging - enclosed in "Accessory box")
- front & mid-mount rack kit (included from original Intel kit, with Intel documentation removed)
- CD**:** Sentivist™ Sensor v4.0.2

Sentivist™ Sensor 320C Hardware:

- 1U Rack mountable Chassis for Intel WV533 board - 350W Power Supply, 2 x 1" ATA Cold-swap OR 3 x 1" U320 Hot-Swap HDD
- WV533 SCSI board for Dual Xeon processors w/512k cache, 400 or 533MHz bus
- WV533 SCSI 1U Backplane
- Intel® Xeon™ processor 2.8GHz, with 533MHz FSB & 512k cache, 1U Heat sink variant
- Black locking front 1U Bezel 1
- Slimline CD/Floppy combo kit.
- 36 GB Ultra 320 15K RPM Cheetah LP 1" with SCA Connector
- 1GB DDR266 Low Profile Reg ECC RAM (requires 2 matching pieces) - 4GBs total
- Single Port Copper (10/100/1000) Ethernet
- RJ-45 to DB9 Serial Console adapter Cable
- US Power Cord, PS/2 "Y-cable" (included with original Intel packaging - enclosed in "Accessory box")
- CD: Sentivist™ Sensor v4.0.2

Sentivist™ Sensor 320F Hardware

- 1U Rack mountable Chassis for Intel WV533 board - 350W Power Supply, 2 x 1" ATA Cold-swap OR 3 x 1" U320 Hot-Swap HDD
- WV533 SCSI board for Dual Xeon processors w/512k cache, 400 or 533MHz bus
- WV533 SCSI 1U Backplane
- Intel® Xeon™ processor 2.8GHz, with 533MHz FSB & 512k cache, 1U Heat sink variant
- Black locking front 1U Bezel
- Slimline CD/Floppy combo kit.
- 36 GB Ultra 320 15K RPM Cheetah LP 1" with SCA Connector

- 1GB DDR266 Low Profile Reg ECC RAM (requires 2 matching pieces) - 4GBs total
- SysKonnect SK-9844 SK-NET GE-SX dual link Optical Gb Ethernet
- RJ-45 to DB9 Serial Console adapter Cable
- US Power Cord, PS/2 "Y-cable" (included with original Intel packaging - enclosed in "Accessory box")
- CD: Sentivist™ Sensor v4.0.2

The three Sentivist Sensors that have been described above vary in hardware configurations as shown above. The software image that implements the security enforcing functionality is the same on all the Sentivist Sensors. The user interfaces are identical across all the Sensor models, hence all the Sensor models are considered identical. Similarly the operating system irrespective of whether it is a Red Hat Linux 7.3 or 8.0 or Solaris 8 or 9 distributions ensures that the same functionality is provided by the Sentivist Server.

The primary differences between the various Sensor models can be summed up in two points

- The Kernel/OS configuration: A uniprocessor kernel versus a multiprocessor kernel and the network interface device drivers included/supported in the Sentivist Sensor 310C model versus the Sentivist Sensor 320 Series models. The Sentivist Sensor 310C configuration does not support using the SysKonnect 9844 fiber Gigabit Ethernet card so it has not been included in the kernel configuration, thus reducing wastage of resources and memory.
- Sensor Software: The differences are in the cosmetic model designation that is displayed and in the configuration file. The configuration file has instructions to run multiple instances of the engine versus a single instance to handle greater network throughput.

3.4 Logical Scope and Boundary of the TOE

The logical boundary of the TOE contains all the components residing within the physical boundary of the TOE. The three logical components of distinction are the Sentivist™ Sensor, Sentivist™ Server and the Sentivist™ Administration Interface. The TOE resides on the network that it has been assigned to monitor. The Sentivist™ Sensor can be strategically placed at various points on the network where there is a need for monitoring the traffic flow through the network. Management of the TOE is accomplished by using Sentivist™ Administration Interface (SAI) to access Sentivist™ Server. Also associated with the three primary logical components that make up the TOE are the MySQL database and the Sentivist Server Command Line Interface. The MySQL database is the repository that stores all audit events. Figure 4 given below clearly illustrates the logical scope, boundary and all the logical interfaces of the TOE.

The Windows operating system installed on the machine hosting the Administration Interface is installed and configured in accordance with the evaluated secure installation guidance. Therefore the remaining security features of the Windows operating systems are irrelevant to the enforcement of the TOE Security Functions claimed in the ST and are not a part of the evaluation.

The security functional requirements implemented by the Sentivist™ are usefully grouped under the following classes or families:

- Security Audit
- Identification and Authentication
- Security Management

- Protection of the TOE Security Functions
- IDS Component Requirements

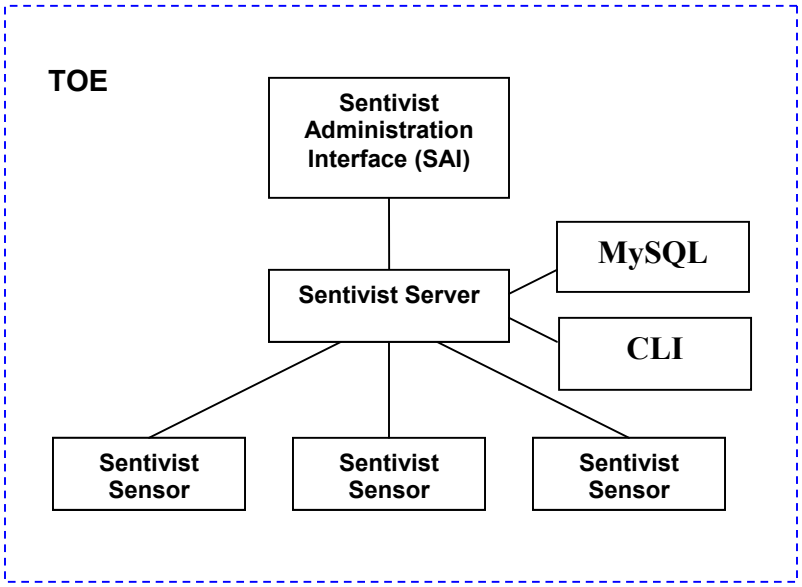

Figure 4: Logical Scope and Boundary of the TOE

Security Audit:

The Sentivist™ collects audit data from internal actions, user actions and provides the functionality to review the audit logs. It also enforces a restriction on access to the audit log. All the audit events are stored in the Sentivist™ Server in a MySQL database; this data is accessible only to authorized system administrators and authorized operators as defined in the Security Management requirements. The Sentivist™ provides configurable audit functions to record audit events. Auditable events include start-up and shutdown of audit functions, start-up and shutdown of Sentivist™ processes and access to the audit data. Each audit event holds date and time of the event, type of event and subject or user identity. The Sentivist™ detects the occurrence of selected events, gathers information concerning them, and records it. Audit reporting features are also provided by the Sentivist™. Included among the reporting features are the three logs; Auditlist, Systemlist and Networklist. These logs can be viewed by users who have the required permissions. System administrators of the Sentivist™ can reconfigure Space Management to purge audit records, but neither the administrator nor Space Management can modify the audit data. Deletion and modification of audit data is done by the space management system. The goal of the Space Management System is to maintain as much information that is useful to the user without exceeding a set limit on the number of bytes of disk storage used. The Space Management System monitors the size of the data-store. This begins a check, which lasts until Space management has no further action to take. When the size exceeds user-configured limits, Space Management sends an alert to notify users of the condition. Space Management then begins stepping through the event and alert records in chronological order. This process continues till the size of the data-store drops below the clearance limit.

Only authorized system administrators and authorized operators have the ability to read the audit data from the system and configure audit parameters. Some users can be endowed with read permissions alone and some can be given permissions to configure audit data too. Audit data can be viewed via the Sentivist™ Administration Interface and

configuration of audit parameters can be done via the SAI by logging into the Sentivist™ Server. The TOE has sufficient access controls in place to ensure that only authorized roles can read audit data; all other identities are denied access. The TOE has provisions by means of which authorized administrators and operators can sort audit data based on date and time, subject identity, type of event and success or failure of the related event. This is accomplished via the Sentivist™ Administration Interface.

The Sensor administrator has access to the Sentivist Sensor Management Menu during the installation and configuration mode of the Sentivist Sensor, This mode involves the initial installation and configuration (Configuring Time and Date, Configuring Sentivist Sensor, Accessing Administration) of the Sentivist Sensor. As the initial configuration of the Sensor cannot be done remotely from the Sentivist Server, this needs to be done either manually using the Sentivist Sensor Management menu at the Sensor console or by using a floppy which has the configuration information written on it. This mode will be outside the scope of the evaluated configuration of the TOE. Once communication is established between the Sentivist Server and the Sensors, every action (i.e configuration changes on the Sensor using the CLI etc.) is logged as an alert on the Server and can be viewed via the SAI, thus the audit is realized for the configuration parameters of the Sentivist Sensor.

The components of the Sentivist™ that implement the Security Audit functions are the Sentivist™ Sensor, Sentivist™ Server, the operating system underlying the Server (Red Hat Linux and Solaris), MySQL database, the Server Command Line Interface (CLI), Sentivist Sensor Management Menu and the Sentivist™ Administration Interface.

Identification and Authentication (I&A):

The NFR Sentivist™ requires identification and authentication before performing any security relevant functions. Users of the TOE are authenticated based on a username and/or a token combination. Prior to identification, a user is only authorized to initiate the authentication process and is presented with an initial login prompt before being identified. Before a user is identified successfully, no other TSF mediated actions can be performed on behalf of the user.

The TOE provides for three defined roles. Authorized Administrator, Authorized System Administrator\nfr, Authorized Operator.

- **Authorized Administrator:** Operating system administrator for the Sentivist™ Server, this person is responsible for the installation and configuration of the Sentivist™ Server and the underlying operating system.

- **Authorized System Administrator**: This default authorized system administrator role created during the installation of the Sentivist™ Server "nfr" has all privileges. The default user name for this role is "nfr". This role will be referred to as the "nfr" role in this document. This role is created with all of the Access Control, Audit View, Audit Configure, Configure, Alert View, Restart Sensor, Query and Diagnostic privileges. This role is the primary role through which one can log into the Sentivist Administration Interface and create Authorized Operators, who are defined as having equal or lesser privileges than the "nfr" account. In addition to this, the "nfr" account also performs administrative functions on the Sentivist Sensor via the Sentivist Server CLI; this is accomplished during the Operations Mode of the Sentivist Sensor. It is important to note that Authorized Operators are not to log into the Command Line Interface.

- The authorized system administrator 'nfr' role created during installation of the Sentivist Server has the ability to login to the Server via the SAI and create Authorized Operators with equal or lesser privileges. The Authorized Operators are created from the Sentivist Administration Interface Console. Permissions control what an Authorized Operator can see and do within the Sentivist Administration Interface. The SAI displays only the interface functions that the authorized system administrator is authorized to use. Due to the sensitive nature of the Sentivist system, it is advised that a user be assigned with the minimum level of permissions required by that user. The following permissions can either be enabled or disabled, in any combination.

  The authorized system administrator 'nfr' role can use the `sudo` command from the command line interface to be granted escalated privileges and act as the authorized administrator and view audit records.

  Below is a list of permissions that are granted to the authorized system administrator 'nfr' account on the Sentivist Administration Interface. The authorized operators created by the authorized system administrator can have some or all of these permissions depending on the level of access.

  - **Access Control**: Allows the user to add and delete user accounts, change user permissions, set authentication attempts for users, and unlock users. This is the highest possible level of permission, which should only be assigned to one other User in addition to the Authorized administrator or nfr user.
  - **Configure**: The configure permissions controls access to the Administration, Packages, and Status shortcut bars within Sentivist Administration Interface, allowing the user to change the configuration of various components. Enabling this capability allows a user to:
    - **Configure Variables**
    - **Configure Alerts**
    - **Configure Packages and Backends**
    - **Mirror package and backend configurations**
    - **Update package and backends**
  - **Diagnostics:** Allows a user to retrieve diagnostic files from Sentivist Server. This feature is rarely used. Diagnostics is usually done at the request of NFR Security Support to aid the user in troubleshooting extreme cases.
  - **Audit Configure:** Allows the user to configure audit-related alerts and functions. Users with this permission have the ability to change the rules that process security- related events.
  - **Audit View:** Allows the user to see audit related alerts (records written to the auditlist) through the Alert Window or popup windows. The auditlist is designed to track events related to internal security.
  - **Query:** The query permission allows access to the Packages shortcut bar. This allows the user to query the data recorded by all packages and backends.
  - **Alert View:** The alert view permission allows viewing of systemlist and networklist alerts. In addition, the permission allows the user to receive alerts via the popup window.
  - **Restart Sentivist Sensor:** This permission allows quick or full reboot of a Sentivist Sensor connected to a Sentivist Server
  - **Sensor Box:** This allows the user access to the Sensor for the sole purpose of using the Run Mirror command to push data out to the Sensor.

During the Installation and Configuration mode of the Sentivist Sensor a role namely the Sensor Administrator account is created with access to the Sentivist Sensor Management Menu. The Sensor Management menu is accessible via a serial port or direct keyboard and monitor connection to the Sensor. The Sensor Administrator once logged into the Sensor Management Menu can access the following functions and manually configure the Sensor the same configuration can be carried out using a hand crafted configuration floppy. The installation and configuration mode of the Sentivist Sensor is not a part of the evaluated configuration of the TOE.

- **Name of this Station:** A unique name for the Sentivist Sensor
- **Management Interface:** The NIC designated as the Management NIC
- **IP Address of the Sensor:** The IP address of the Sentivist Sensor
- **Network Mask of the Sensor:** The network mask for this Sentivist Sensor's IP Address.
- **Default Router:** The IP address for the default router for the Sentivist Sensor's network
- **IP Address of the Sentivist Server:** The IP address of the Sentivist Server used to manage this Sentivist Sensor.
- **Encryption Passphrase:** The passphrase used to encrypt communication between the Sentivist Sensor and the Sentivist Server used to manage the Sensor.
- **Time:** The path to the time zone file
- **Administrator Password:** The password used to access the Management Menu.
- **License Key:** The unique license key for the Sentivist Sensor.

The Sensor Administrator can access/change all of the above mentioned functions but cannot do any actual configuration changes such as changing package and backend configurations to the Sentivist Sensor.

The Sentivist™ Server enforces all permissions and authentication functionality for the Administration Interface. Authentication is performed via the SRP protocol. Secure Remote Password (SRP) protocol is used for negotiating secure connections and exchanging keys. It's considered a verifier based algorithm, as it uses a password verifier as a key to several of the computations. In the SRP protocol the password or a variation on the password is never stored on the disk. Instead, the SRP algorithm stores a *password verifier*, computed using the SHA of randomly generated salt, the username and the original raw password as given by the user. The username, password verifier, salt and lockout values, which are used to generate passwords are saved to disk in the file format:

<username, password verifier, salt, lockout>

In no form is the password saved to disk. The verifier is a **256-bit integer** computed using the following:

Verifier Computation:

<salt> = random()

x = SHA-1(<salt> | SHA-1(<username> ":" <raw password>))

<password verifier> = $v = g^x \% N$

The | indicates concatenation and % is the modulo operation. The 160-bit result of the SHA-1 operation is implicitly converted to an integer before it is operated upon. Neither the password, nor the password verifier is ever passed plaintext from client to server and back.

It can be clearly seen from the above computation that the password verifier is generated as a SHA digest of randomly chosen salt, user name and the raw password. The strength of the password verifier is directly proportional to the strength of the salt, user name and raw password.

The auditing mechanisms within the Sentivist™ ensure that all unsuccessful authentication attempts are audited and an alert is sent via the Sentivist™ Administration Interface. After a configurable number of unsuccessful authentication attempts have been made, Sentivist™ will prevent any other login attempts from that user account until "nfr" role or an authorized system administrator who has access control permissions restores the account.

The Sentivist™ Server implements all of the Identification and Authentication functions.

Security Management:

The Sentivist™ Server provides all Sentivist™ security management capabilities. The protection mechanisms within the Sentivist™ provide assurance that only authorized system administrators with required permissions are allowed to modify the system data collection, analysis and reaction functions. All remote access by subjects to objects is mediated by the Sentivist™ Server which in turn interfaces with the underlying operating system.

The protection mechanisms in place ensure that only an authorized system administrator has sufficient privileges to query and modify all other TOE data. Access to the TOE and TOE data is controlled by the authentication and access control mechanisms that the TOE provides and implements successfully. The Sentivist™ maintains the following roles: Authorized Administrator (Operating System root account), Authorized System Administrator, and Authorized Operators. The Authorized Administrator (OS root account) is responsible for the installation and configuration of the Sentivist Server and the Authorized System Administrator (nfr) role is created upon installation. The Authorized System Administrator/ "nfr" account has the permissions to log into the Sentivist™ Server using the Sentivist™ Administrator Interface.

The Guisrv enforces the management application interface; Guisrv acts as a secure broker for requests between the clients (Sentivist Administration Interface) and the Sentivist™ Server. It works as an agent with the clients wishing to interact with the Sentivist Server. The requesting service makes a series of HTTP POST requests to the server that encapsulates a command. The Server processes the command by either running hard-coded routines or running a specified command program. The program executes, and the results are sent back to the requesting agent in the form of an HTTP response. The difference between **guisrv** and a normal web server is that it supports authentication, wraps the password in encryption, and folds all data into an encrypted envelope.

Guisrv supports two cryptographic features:

- Encryption to protect access to the system using Secure Remote Password (SRP-6) algorithm.
- Calculating and attaching a MAC to the text inside the envelope to verify the correctness of the text after decryption.

The encryption protocol is broken into two major pieces: the authentication handshake, which verifies the user's password and authority to connect to the server, and the envelope protocol, which encrypts the message between the client and server. A single call down to the client is one, complete ***session***. During the session, the client connects, authenticates, requests, and reads a response. The authentication handshake is kicked off by the client when a user wishes to log into the system and access its features. The algorithm computes sets of numbers and compares them as session keys to verify the correctness of both sides of the connection before sending any actual data.

The Sentivist Sensor to Sentivist Server communication process is **getserver** in daemon mode. The Sentivist Sensor initiates a connection with the Sentivist Server and a secure channel is negotiated through a handshake. The **getserver** communication protocol with the Sensor consists of a series of commands and responses.

getserver protocol uses a six-step handshake to authenticate between the client and the server. This authentication mechanism uses a shared secret. This shared secret is created upon installation of the Sentivist Server and stored in the Sentivist Server. The same is provided to the Sensors in two ways, via a configuration floppy or by accessing the Sentivist Sensor Management Menu. This shared secret is known only to the Authorized System Administrator/ "nfr" role; only this role has the privilege to perform administrative functions on the Sentivist Sensor.

The components of the Sentivist™ that implement the Security Management functions are the Sentivist™ Server, Sentivist Sensors, Red Hat Linux, Sun Solaris the operating systems underlying the Server, and the Sentivist™ Administration Interface.

Protection of TOE Security Functions:

The TOE protection security functions are responsible for the confidentiality and integrity of all data transmitted between the Sentivist™ Administration Interface, Sensor and the Server ensuring non-bypassability of the TOE security policy and maintaining domain separation on the Sentivist™. The Sentivist Sensor's time is recorded in the MySQL database and used when the Sentivist Administration Interface displays events and alerts.

Sentivist™ uses a proprietary application protocol data exchange mechanism to ensure confidentiality of all data transmitted to any remote trusted IT product. In the Sentivist, remote trusted IT products are the various TOE components interacting with one another. To ensure integrity the Sentivist™ uses Message Authentication Codes. Sentivist™ has sufficient protection mechanisms in place to ensure that TSP enforcement functions are invoked and they succeed before each function in the TSC is allowed to proceed. Also non-physical access to the system is mediated by the TOE which acts as a reference monitor and therefore the TOE shall validate all actions between the components that require policy enforcement, before allowing the action to succeed. The Sentivist™ provides access controls that enforce separation between the security domains of the subjects.

The components of the Sentivist™ that implement the Protection of TOE Security functions are the Sentivist™ Sensor, Sentivist™ Server, the operating system underlying the Server (Red Hat Linux and Solaris), and the Sentivist™ Administration Interface.

IDS Component Requirements:

The IDS component requirements security functions are responsible for data collection and analysis from the target IT network. Initial data collection and analysis occurs on the Sentivist™ Sensor which monitors traffic in near real time; examining packets for suspicious activity, attacks and anomalous behavior. Raw packets are captured and compared against signatures of known attacks. These signatures can be edited by authorized administrators and operators of the Sentivist™. When there is a match to a signature, information on the packet (or packets) is collected by the Sentivist™ Sensor. This information is relayed to the Sentivist™ Server. The Server in turn applies rules which are configurable by an authorized administrator or operator via the Sentivist™ Administration Interface. These rules can be configured in such a way that a pop-up warning can be generated from the SAI, if a specific event is detected. These events are then written to a MySQL database residing on the Server through an alert daemon (alertd).

The Sentivist™ has sufficient access controls to ensure that only authorized system administrator/ "nfr" role and authorized operators with read/query privileges will be able to view the Sentivist™ event data.

The Sentivist™ has a Space Management System which periodically monitors the size of the data store. The data store houses all the alert and event data. The primary goal of the Space Management System is to maintain as much information that is useful to the user without exceeding a set limit on the number of bytes. When the data store has reached its maximum configuration size, the oldest audit events will be deleted and an alarm will be generated and sent to the Sentivist™ Administration Interface. Only the authorized system administrator and authorized operators with the necessary permissions are able to configure the Space Management System. The components of the Sentivist™ that implement the IDS Component Requirements functions are the Sentivist™ Sensor, Sentivist™ Server, The operating system underlying the Server (Red Hat Linux and Solaris), MySQL database and the Sentivist™ Administration Interface.

The Sentivist™ logical boundary includes the Sentivist™ Administration Interface, Sentivist™ Sensor Hardware and the CD on which the Sensor software is resident, Sentivist™ Server, the underlying Server Operating System (Red Hat Linux, Solaris), The Server Command Line Interface and the MySQL database. The evaluated secure configuration must contain the same physical and logical isolation. The logical scope of the Sentivist™ extends to the five classes or families of security functional requirements just mentioned.

3.5 TOE Security Services

This section lists and describes, at a high level, the security services that are provided by the TOE. The security functional requirements implemented by the Sentivist™ are grouped under the following security classes or families:

Security Audit: Sentivist™ collects audit data for internal actions, user actions, and provides the ability to review audit logs. It also restricts access to the audit log. The TOE tracks authentication attempts and changes to each user's permissions and access.

Identification and Authentication: Sentivist™ requires a user to identify and authenticate itself to the TOE before performing any security relevant functions. Users of the TOE are authenticated based on a username and token combination. All communication through a network operates via AES and HMAC SHA-1.

Security Management: Sentivist™ provides for two defined levels of users. Of these levels, only the Authorized System Administrator/ "nfr" user and Authorized Operators endowed with view/query privileges can query system and audit data. No authorized system administrator or operator with other permission levels (e.g. without view/query privileges) can add or query system/audit data. The protection mechanisms of the TOE ensure this property. The concept of users in the NFR Sentivist is covered in detail under the Identification and Authentication class in Section 3.4: Logical Scope and Boundary of the TOE.

Protection of the TOE Security Functions: The Sentivist™ uses cryptographic mechanisms to ensure the confidentiality and integrity of all data transmitted to any remote trusted IT products. The Sentivist™ uses a proprietary application protocol data exchange with the appliances and the SAI over TCP. The Sentivist™ also provides access controls to enforce the security policy of the TOE.

IDS Component Requirements: The Sentivist™ has in place sufficient access controls to ensure that only authorized administrators can delete and/or modify IDS event data.

4.0 TOE Security Environment

This section identifies the following components for the TOE:

1) Significant assumptions about the TOE's operational environment

2) IT-related threats addressed countered by TOE components

3) Organizational security policies for which this TOE is appropriate

This information provides the basis for the Security Objectives, the Security Requirements for the IT Environment, and the TOE Security Functional Requirements. The TOE Security Environment described below is taken directly from the Intrusion Detection System System Protection Profile Version 1.4, February 4, 2002.

4.1 ASSUMPTIONS

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

4.1.1 Intended Usage Assumptions

A.ACCESS The TOE has access to all the IT System data it needs to perform its functions.

A.DYNMIC The TOE will be managed in a manner that allows it to appropriately address changes in the IT System the TOE monitors.

A.ASCOPE The TOE is appropriately scalable to the IT System the TOE monitors.

4.1.2 Physical Assumptions

A.PROTCT The TOE hardware and software critical to security policy enforcement will be protected from unauthorized physical modification.

A.LOCATE The processing resources of the TOE will be located within controlled access facilities, which prevent unauthorized physical access.

4.1.3 Personnel Assumptions

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

A.NOEVIL The authorized administrators are not careless, willfully negligent, or hostile, and will follow and abide by the instructions provided by the TOE documentation.

A.NOTRST The TOE can only be accessed by authorized users.

4.2 THREATS

The following are threats identified for the TOE and the IT System the TOE monitors. The TOE itself has threats and the TOE is also responsible for addressing threats to the environment in which it resides. The assumed levels of expertise of the attacker for all the threats are unsophisticated.

4.2.1 TOE Threats

T.COMINT An unauthorized user may attempt to compromise the integrity of the data collected and produced by the TOE by bypassing a security mechanism.

T.COMDIS An unauthorized user may attempt to disclose the data collected and produced by the TOE by bypassing a security mechanism.

T.LOSSOF An unauthorized user may attempt to remove or destroy data collected and produced by the TOE.

T.NOHALT An unauthorized user may attempts to compromise the continuity of the System's collection and analysis functions by halting execution of the TOE.

T.PRIVIL An unauthorized user may gain access to the TOE and exploit system privileges to gain access to TOE security functions and data.

T.IMPCON An unauthorized user may inappropriately change the configuration of the TOE causing potential intrusions to go undetected.

T.INFLUX An unauthorized user may cause malfunction of the TOE by creating an influx of data that the TOE cannot handle.

T.FACCNT Unauthorized attempts to access TOE data or security functions may go undetected.

4.2.2 IT System Threats

The following identifies threats to the IT System that may be indicative of vulnerabilities in or misuse of IT resources.

T.SCNCFG	Improper security configuration settings may exist in the IT System the TOE monitors.
T.SCNMLC	Users could execute malicious code on an IT System that the TOE monitors which causes modification of the IT System protected data or undermines the IT System security functions.
T.SCNVUL	Vulnerabilities may exist in the IT System the TOE monitors.
T.FALACT	The TOE may fail to react to identified or suspected vulnerabilities or inappropriate activity.
T.FALREC	The TOE may fail to recognize vulnerabilities or inappropriate activity based on IDS data received from each data source.
T.FALASC	The TOE may fail to identify vulnerabilities or inappropriate activity based on association of IDS data received from all data sources.
T.MISUSE	Unauthorized accesses and activity indicative of misuse may occur on an IT System the TOE monitors.
T.INADVE	Inadvertent activity and access may occur on an IT System the TOE monitors.
T.MISACT	Malicious activity, such as introductions of Trojan horses and viruses, may occur on an IT System the TOE monitors.

4.3 ORGANIZATIONAL SECURITY POLICIES

An organizational security policy is a set of rules, practices, and procedures imposed by an organization to address its security needs. This section identifies the organizational security policies applicable to the Intrusion Detection System System Protection Profile to which this ST conforms.

P.DETECT	Static configuration information that might be indicative of the potential for a future intrusion or the occurrence of a past intrusion of an IT System or events that are indicative of inappropriate activity that may have resulted from misuse, access, or malicious activity of IT System assets must be collected.
P.ANALYZ	Analytical processes and information to derive conclusions about intrusions (past, present, or future) must be applied to IDS data and appropriate response actions taken.
P.MANAGE	The TOE shall only be managed by authorized users.
P.ACCESS	All data collected and produced by the TOE shall only be used for authorized purposes.
P.ACCACT	Users of the TOE shall be accountable for their actions within the IDS.
P.INTGTY	Data collected and produced by the TOE shall be protected from modification.
P.PROTCT	The TOE shall be protected from unauthorized accesses and disruptions of TOE data and functions.

5.0 TOE SECURITY OBJECTIVES

This section identifies the security objectives of the TOE and its supporting environment. The security objectives identify the responsibilities of the TOE and its environment in meeting the security needs. The Security Objectives described below are taken directly from the Intrusion Detection System System Protection Profile Version 1.4, February 4, 2002.

5.1 INFORMATION TECHNOLOGY (IT) SECURITY OBJECTIVES

The following are the TOE security objectives:

O.PROTCT The TOE must protect itself from unauthorized modifications and access to its functions and data.

O.IDSCAN The Scanner must collect and store static configuration information that might be indicative of the potential for a future intrusion or the occurrence of a past intrusion of an IT System.

O.IDSENS The Sensor must collect and store information about all events that are indicative of inappropriate activity that may have resulted from misuse, access, or malicious activity of IT System assets and the IDS.

O.IDANLZ The Analyzer must accept data from IDS Sensors or IDS Scanners and then apply analytical processes and information to derive conclusions about intrusions (past, present, or future).

O.RESPON The TOE must respond appropriately to analytical conclusions.

O.EADMIN The TOE must include a set of functions that allow effective management of its functions and data.

O.ACCESS The TOE must allow authorized users to access only appropriate TOE functions and data.

O.IDAUTH The TOE must be able to identify and authenticate users prior to allowing access to TOE functions and data.

O.OFLOWS The TOE must appropriately handle potential audit and System data storage overflows.

O.AUDITS The TOE must record audit records for data accesses and use of the System functions.

O.INTEGR The TOE must ensure the integrity of all audit and System data.

O.EXPORT When any IDS component makes its data available to another IDS component, the TOE will ensure the confidentiality of the System data.

5.2 SECURITY OBJECTIVES FOR THE ENVIRONMENT

The TOE's operating environment must satisfy the following objectives. These objectives do not levy any IT requirements but are satisfied by procedural or administrative measures.

O.INSTAL	Those responsible for the TOE must ensure that the TOE is delivered, installed, managed, and operated in a manner which is consistent with IT security.
O.PHYCAL	Those responsible for the TOE must ensure that those parts of the TOE critical to security policy are protected from any physical attack.
O.CREDEN	Those responsible for the TOE must ensure that all access credentials are protected by the users in a manner which is consistent with IT security.
O.PERSON	Personnel working as authorized administrators shall be carefully selected and trained for proper operation of the System.
O.INTROP	The TOE is interoperable with the IT System it monitors.

6.0 TOE IT Security Requirements

This section specifies the Security Functional Requirements (SFRs) for the TOE. This section defines the functional requirements for the TOE. Functional requirements in this ST were drawn from Part 2 of the CC. These requirements are relevant to supporting the secure operation of the TOE. This Security Target also responds to explicitly stated requirements that are present in the IDS PP. These new requirements are indicated with the text (EXP) in the title. Additionally all international interpretations are stated as International Interpretation # xxx in parenthesis along with the SFR statement. An overview of the TOE Security Functional Requirements is presented in Table 1 below.

Table 1 - TOE Security Functional Requirements

SFR ID	Functional Component	ST Operation
FAU_GEN.1	Audit data generation	Refinement
FAU_SAR.1	Audit review	Assignment
FAU_SAR.2	Restricted audit review	None
FAU_SAR.3	Selectable audit review	None
FAU_SEL.1	Selective audit	Assignment
FAU_STG.2	Guarantees of audit availability	Assignment/Selection and Refinement
FAU_STG.4	Prevention of audit data loss	Selection
FIA_UAU.1	Timing of authentication	Assignment
FIA_AFL.1	Authentication failure handling	Refinement
FIA_ATD.1	User attribute definition	Assignment
FIA_UID.1	Timing of identification	Assignment
FMT_MOF.1	Management of security functions behavior	None
FMT_MTD.1	Management of TSF data	Assignment
FMT_SMF.1.1	Specification of management functions	Assignment
FMT_SMR.1	Security roles	Assignment
FPT_ITA.1	Inter-TSF availability within a defined availability metric	Assignment
FPT_ITC.1	Inter-TSF confidentiality during transmission	None
FPT_ITI.1	Inter-TSF detection of modification	Assignment
FPT_RVM .1	Non-bypassability of the TSP	None
FPT_SEP.1	TSF domain separation	None
FPT_STM.1	Reliable time stamps	None
IDS_SDC.1 (EXP)	System data collection	Selection/Assignment
IDS_ANL.1 (EXP)	Analyzer analysis	Selection/Assignment
IDS_RCT.1 (EXP)	Analyzer react	Assignment
IDS_RDR.1 (EXP)	Restricted data review	Assignment
IDS_STG.1 (EXP)	Guarantee of system data availability	Selection/Assignment
IDS_STG.2 (EXP)	Prevention of system data loss	Selection

The Common Criteria standard defines four basic operations that can be performed on the requirements to further clarify and define them: Assignment, Selection, Iteration, and Refinement. This ST will highlight instantiations of the three operations that are used and completed:

- Assignment: Allows the specification of an identified parameter. Indicated with **bold text.**
- Refinement: Allows the addition of details. Indicated with ***bold text and italics.***

- Selection: The **selection** operation is used to select one or more options provided by the CC in stating a requirement. Selections are denoted by underlined text.
- Iteration: Allows a component to be used more than once with varying operations. (Not used in this ST)
- Operations that the PP author left for the ST author to complete are denoted with the above mentioned conventions but in square brackets.

The following sections present the TOE Security Functional Requirements (SFRs) with any ST operations performed on them based on the requirements from the Intrusion Detection System System PP.

6.1 SECURITY AUDIT (FAU)

6.1.1 FAU_GEN.1 **Audit data generation (International Interpretation # 202)**

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

a) Start-up and shutdown of the audit functions;

b) All auditable events for the basic level of audit; and

c) **Access to the System and access to the TOE and System data.**

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:
a) Date and time of the event, type of event, subject identity and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, **the additional information specified in the Details column of Table 2 Auditable Events.**

Table 2 – Auditable Events

Component	Event	Details
FAU_GEN.1	Start-up and shutdown of audit functions	
FAU_GEN.1	Access to System	
FAU_GEN.1	Access to the TOE and System data	**Object IDS, Requested access**
FAU_SAR.1	Reading of information from the audit records	
FAU_SAR.2	Unsuccessful attempts to read information from the audit records	
FAU_SEL.1	All modifications to the audit configuration that occur while the audit collection functions are operating	

Component	Event	Details
FIA_UAU.1	All use of the authentication mechanism	**User identity, location**
FIA_UID.1	All use of the user identification mechanism	**User identity, location**
FMT_MOF.1	All modifications in the behavior of the functions of the TSF	
FMT_MDT.1	All modifications to the values of TSF data	
FMT_SMR.1	Modifications to the group of users that are part of a role	**User identity**
FPT_ITI.1	The action taken upon detection of modification of transmitted TSF data	

6.1.2 FAU_SAR.1 **Audit Review**

FAU_SAR.1.1 The TSF shall provide **[an Authorized System Administrator/nfr and authorized operators with audit view, audit configure and query permissions]** with the capability to read **[all audit data]** from the audit records.

FAU_SAR.1.2 The TSF shall provide the audit records in a manner suitable for the user to interpret the information.

6.1.3 FAU_SAR.2 **Restricted audit review**

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

6.1.4 FAU_SAR.3 **Selectable audit review**

FAU_SAR.3.1 The TSF shall provide the ability to perform sorting of audit data based on **date and time, subject identity and type of event and success or failure of related event.**

6.1.5 FAU_SEL.1 **Selective audit**

FAU_SEL.1.1 The TSF shall be able to include or exclude auditable events from the set of audited events based on the following attributes:

a) event type;

b) [No other attributes]

6.1.6 FAU_STG.2 **Guarantees of audit data availability (International Interpretation 141)**

FAU_STG.2.1 The TSF shall protect the stored audit records ***in the audit trail*** from unauthorized deletion.

FAU_STG.2.2 The TSF shall be able to detect unauthorized modifications to the audit records in the audit trail.

FAU_STG.2.3 The TSF shall ensure that **[a configurable percentage value of Maximum Space defined as Clearance limit]** audit records will be maintained when the following conditions occur: [audit storage exhaustion].

6.1.7 FAU_STG.4 **Prevention of audit data loss**

FAU_STG.4.1 The TSF shall [delete the oldest stored audit records] and **send an alarm** if the audit trail is full.

6.2 IDENTIFICATION AND AUTHENTICATION (FIA)

6.2.1 FIA_UAU.1 **Timing of authentication**

FIA_UAU.1.1 The TSF shall allow **[initiation of the authentication process, presentation of the login prompt]** on behalf of the user to be performed before the user is authenticated.

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

6.2.2 FIA_AFL.1 **Authentication failure handling**

FIA_AFL.1 Authentication failure handling

FIA_AFL.1.1 The TSF shall detect when **a settable, non-zero number** of unsuccessful authentication attempts occur related to **external IT products attempting to authenticate.**

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall **prevent the offending external IT product from successfully authenticating until an authorized administrator takes some action to make authentication possible for the external IT product in question.**

6.2.3 FIA_ATD.1 **User attribute definition**

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users:

a) User identity;

b) Authentication data;

c) Authorizations; and

d) [The number of sequential failed login attempts].

6.2.4 FIA_UID.1 **Timing of identification**

FIA_UID.1.1 The TSF shall allow **[initiation of the authentication process, presentation of the login prompt]** on behalf of the user to be performed before the user is identified.

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

6.3 SECURITY MANAGEMENT (FMT)

6.3.1 FMT_MOF.1 **Management of security functions behavior**

FMT_MOF.1.1 The TSF shall restrict the ability to modify the behavior of the functions **System data collection, analysis and reaction** to **Authorized System Administrators and Authorized Operators.**

6.3.2 FMT_MTD.1 **Management of TSF data**

FMT_MTD.1.1 The TSF shall restrict the ability to query **and add System and audit data, and shall restrict the ability to query and modify all other TOE data** to **[authorized administrator, authorized system administrator/nfr and authorized operators with configure, audit configure, query, audit view, or alert view permissions]**.

6.3.3 FMT_SMF.1 **Specification of Management Functions (International Interpretation-065)**

FMT_SMF.1.1 The TSF shall be capable of performing the following security management functions: **Management of system data collection, analysis and reaction, Management of system and audit data, Management of all other TOE data.**

6.3.4 FMT_SMR.1 **Security roles**

FMT_SMR.1.1 The TSF shall maintain the ***following*** roles**: authorized administrator, authorized System administrators, authorized operators and [none].**

FMT_SMR.1.2 The TSF shall be able to associate users with roles.

6.4 PROTECTION OF THE TOE SECURITY FUNCITONS (FPT)

6.4.1 FPT_ITA.1 **Inter-TSF availability within a defined availability metric**

FPT_ITA.1.1 The TSF shall ensure the availability of **audit and System data** provided to a remote trusted IT product within **[60 seconds]** given the following conditions **[normal traffic on the communications network and both IT products operational and available].**

6.4.2 FPT_ITC.1 Inter-**TSF confidentiality during transmission**

FPT_ITC.1.1 The TSF shall protect all TSF data transmitted from the TSF to a remote trusted IT product from unauthorized disclosure during transmission.

6.4.3 FPT_ITI.1 **Inter-TSF detection of modification**

FPT_ITI.1.1 The TSF shall provide the capability to detect modification of all TSF data during transmission between the TSF and a remote trusted IT product within the following metric: **[one detected message authentication code (MAC) error within a transmission]**.

FPT_ITI.1.2 The TSF shall provide the capability to verify the integrity of all TSF data transmitted between the TSF and a remote trusted IT product and perform **[a retransmission before disregarding the session]** if modifications are detected.

6.4.4 FPT_RVM.1 **Non-bypassability of the TSP**

FPT_RVM.1.1 The TSF shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

6.4.5 FPT_SEP.1 **TSF domain separation**

FPT_SEP.1.1 The TSF shall maintain a security domain for its own execution that protects it from interference and tampering by un-trusted subjects.

FPT_SEP.1.2 The TSF shall enforce separation between the security domains of subjects in the TSC.

6.4.6 FPT_STM.1 **Reliable time stamps**

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

6.5 IDS COMPONENT REQUIREMENTS (IDS)

6.5.1 IDS_SDC.1 **System Data Collection (EXP)**

IDS_SDC.1.1 The System shall be able to collect the following information from the targeted IT System resource(s):

a) [start-up and shutdown, identification and authentication events, data accesses, service requests, network traffic, data introduction, detected malicious code, service configuration, detected known vulnerabilities]; and

b) [No other specifically defined events]. (EXP)

IDS_SDC.1.2 At a minimum, the System shall collect and record the following information:

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

b) The additional information specified in the Details column of Table 3 System Events. **(EXP)**

Table 3 - System Events

Component	Event	Details
IDS_SDC.1	Start-up and shutdown	**None**
IDS_SDC.1	Identification and authentication events	**User identity, location, Sources address, Destination address**
IDS_SDC.1	Data accesses	**Object IDS, requested access, sources address, destination address**
IDS_SDC.1	Service requests	**Specific service, Source address, Destination address**
IDS_SDC.1	Network traffic	**Protocol, source address, destination Address**
IDS_SDC.1	Data Introduction	**Object IDS, location of address, destination address**
IDS_SDC.1	Detected malicious code	**Location, identification of code**
IDS_SDC.1	Service configuration	**Service identification (name or port), interface, protocols**
IDS_SDC.1	Detected known vulnerabilities	**Identification of the known vulnerability**

6.5.2 IDS_ANL.1 **Analyzer analysis (EXP)**

IDS_ANL.1.1 The System shall perform the following analysis function(s) on all IDS data received:

a) **[**statistical, signature]; and

b) **[stateful protocol analysis and anomaly detection]**. **(EXP)**

IDS_ANL.1.2 The System shall record within each analytical result at least the following information:

a) Date and time of the result, type of result, and identification of data source; and

b) [Data associated with the result]. **(EXP)**

6.5.3 IDS_RCT.1 **Analyzer react (EXP)**

IDS_RCT.1.1 The System shall send an alarm to **[Sentivist Server]** and take **[a) no action,** or **b) execute user-configurable scripts or executables]** when an intrusion is detected. **(EXP)**

6.5.4 IDS_RDR.1 **Restricted Data Review (EXP)**

IDS_RDR.1.1 The System shall provide an **[authorized system administrator/nfr, and authorized operator with alert view, and query privileges]** with the capability to read **[all event data]** from the System data. **(EXP)**

IDS_RDR.1.2 The System shall provide the System data in a manner suitable for the user to interpret the information. **(EXP)**

IDS_RDR.1.3 The System shall prohibit all users read access to the System data, except those users that have been granted explicit read-access. **(EXP)**

6.5.5 IDS_STG.1 **Guarantee of System Data Availability (EXP)**

IDS_STG.1.1 The System shall protect the stored System data from unauthorized deletion. **(EXP)**

IDS_STG.1.2 The System shall protect the stored System data from modification. **(EXP)**

IDS_STG.1.3 The System shall ensure that **[a configurable percentage value of Maximum Space defined as Clearance limit]** System data will be maintained when the following conditions occur: [System data storage exhaustion]**. (EXP)**

6.5.6 IDS_STG.2 **Prevention of System data loss (EXP)**

IDS_STG.2.1 The System shall [delete the oldest stored System data] and send an alarm if the storage capacity has been reached. **(EXP)**

7.0 TOE ASSURANCE REQUIREMENTS

This section specifies the Security Assurance Requirements (SAR) for the TOE. The assurance requirements are taken from Part 3 of the CC and comprise of EAL2 level of assurance.

7.1 CONFIGURATION MANAGEMENT (ACM)

7.1.1 Configuration Items (ACM_CAP.2) International Interpretation – 003.

ACM_CAP.2.1D The developer shall provide a reference for the TOE.

ACM_CAP.2.2D The developer shall use a CM system.

ACM_CAP.2.3D The developer shall provide CM documentation.

ACM_CAP.2.1C The reference for the TOE shall be unique to each version of the TOE

ACM_CAP.2.2C The TOE shall be labeled with its reference.

ACM_CAP.2.3C The CM documentation shall include a configuration list.
The configuration list shall uniquely identify all configuration items that comprise the TOE.

ACM_CAP.2.4C The configuration list shall describe the configuration items that comprise the TOE.

ACM_CAP.2.5C The CM documentation shall describe the method used to uniquely identify the configuration items.

ACM_CAP.2.6C The CM system shall uniquely identify all configuration items.

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

7.2 DELIVERY AND OPERATION (ADO)

7.2.1 Delivery Procedures (ADO_DEL.1)

ADO_DEL.1.1D The developer shall document procedures for delivery of the TOE or parts of it to the user.

ADO_DEL.1.2D The developer shall use the delivery procedures.

ADO_DEL.1.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.

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

7.2.2 Installation, Generation, and Start-up Procedures (ADO_IGS.1) International Interpretation - 051

ADO_IGS.1.1D The developer shall document procedures necessary for the secure installation, generation, and start-up of the TOE.

ADO_IGS.1.1C The installation, generation and start-up documentation shall describe all the steps necessary for secure installation, generation and start-up of the TOE.

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

ADO_IGS.1.2E The evaluator shall determine that the installation, generation, and start up procedures result in a secure configuration.

7.3 DEVELOPMENT (ADV)

7.3.1 Informal Functional Specification (ADV_FSP.1)

ADV_FSP.1.1D The developer shall provide a functional specification.

ADV_FSP.1.1C The functional specification shall describe the TSF and its external interfaces using an informal style.

ADV_FSP.1.2C The functional specification shall be internally consistent.

ADV_FSP.1.3C The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing details of effects, exceptions and error messages, as appropriate.

ADV_FSP.1.4C The functional specification shall completely represent the TSF.

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

ADV_FSP.1.2E The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

7.3.2 Descriptive High-Level Design (ADV_HLD.1)

ADV_HLD.1.1D The developer shall provide the high-level design of the TSF.

ADV_HLD.1.1C The presentation of the high-level design shall be informal.

ADV_HLD.1.2C The high-level design shall be internally consistent.

ADV_HLD.1.3C The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV_HLD.1.4C The high-level design shall describe the security functionality provided by each subsystem of the TSF.

ADV_HLD.1.5C The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.

ADV_HLD.1.6C The high-level design shall identify all interfaces to the subsystems of the TSF.

ADV_HLD.1.7C The high-level design shall identify which of the interfaces to the subsystem of the TSF are externally visible.

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

ADV_HLD.1.2E The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

7.3.3 Informal Correspondence Demonstration (ADV_RCR.1)

ADV_RCR.1.1D The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided.

ADV_RCR.1.1C For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

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

7.4 GUIDANCE DOCUMENTS (AGD)

7.4.1 Administrator Guidance (AGD_ADM.1)

AGD_ADM.1.1D The developer shall provide administrator guidance addressed to system administrative personnel.

AGD_ADM.1.1C The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.

AGD_ADM.1.2C The administrator guidance shall describe how to administer the TOE in a secure manner.

AGD_ADM.1.3C The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

AGD_ADM.1.4C The administrator guidance shall describe all assumptions regarding user behavior that are relevant to secure operation of the TOE.

AGD_ADM.1.5C The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.

AGD_ADM.1.6C The administrator guidance shall describe each type of security relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_ADM.1.7C The administrator guidance shall be consistent with all other documentation supplied for evaluation.

AGD_ADM.1.8C The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.

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

7.4.2 User Guidance (AGD_USR.1)

AGD_USR.1.1D The developer shall provide user guidance.

AGD_USR.1.1C The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.

AGD_USR.1.2C The user guidance shall describe the use of user-accessible security functions provided by the TOE.

AGD_USR.1.3C The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.

AGD_USR.1.4C The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behavior found in the statement of TOE security environment.

AGD_USR.1.5C The user guidance shall be consistent with all other documentation supplied for evaluation.

AGD_USR.1.6C The user guidance shall describe all security requirements for the IT environment that is relevant to the user.

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

7.5 TESTS (ATE)

7.5.1 Evidence of Coverage (ATE_COV.1)

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 identified in the test documentation and the TSF as described 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.

7.5.2 Functional Testing (ATE_FUN.1)

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, test procedure descriptions, expected test results and actual test results.

ATE_FUN.1.2C The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.

ATE_FUN.1.3C The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.4C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.5C The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.

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

7.5.3 Independent Testing (ATE_IND.2)

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 test a subset of the TSF as appropriate to confirm that the TOE operates as specified.

ATE_IND.2.3E The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

7.6 VULNERABILITY ASSESSMENT (AVA)

7.6.1 Strength of TOE Security Function Evaluation (AVA_SOF.1)

AVA_SOF.1.1D The developer shall perform a strength of TOE security function analysis for each mechanism identified in the ST as having a strength of TOE security function claim.

AVA_SOF.1.1C For each mechanism with a strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the minimum strength level of SOF-basic.

AVA_SOF.1.2C For each mechanism with a specific strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the specific strength of function metric of SOF-basic.

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

AVA_SOF.1.2E The evaluator shall confirm that the strength claims are correct.

7.6.2 Developer Vulnerability Analysis (AVA_VLA.1 – Interp - 051) International Interpretation -051

AVA_VLA.1.1D The developer shall perform a vulnerability analysis.

AVA_VLA.1.2D The developer shall provide vulnerability analysis documentation.

AVA_VLA.1.1C The vulnerability analysis documentation shall describe the analysis of the TOE deliverables performed to search for obvious ways in which a user can violate the TSP.

AVA_VLA.1.2C The vulnerability analysis documentation shall describe the disposition of obvious vulnerabilities.

AVA_VLA.1.3C The vulnerability analysis documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.

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

AVA_VLA.1.2E The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure obvious vulnerabilities have been addressed.

8.0 TOE SUMMARY SPECIFICATION

This section provides a high-level definition of the IT Security Functions and the Assurance Measures provided by the TOE to meet the SFRs and SARs specified in the IDS System PP.

8.1 TOE Security Functions

The TOE provides the following five Security Functions:

- Security Audit
- Identification and Authentication
- Security Management
- Protection of the TOE Security Functions
- IDS Component Requirements

8.2 Security Audit

Audit data generation

All audit events are stored in a MySQL database of the Sentivist™ server which is accessible to authorized users as defined by FMT_SMR.1. The Sentivist™ provides configurable audit functions which are used to record audit events. Auditable events include start-up and shutdown of audit functions, and access to the Sentivist™ and associated data. Associated data encompasses all of the alert and event data. Both alert and event data are Sensor and Server generated information. All start-up and shutdown of Sentivist™ processes are also audited. Each audit event holds the following information: date and time of the event, type of event and subject or user identity.

There are three types of logs: Auditlist, Systemlist, and Networklist. A user can create additional custom logs if desired or needed.

Alert rules tell the Sentivist™ Server where and how to deliver messages. Alerts can be sent to multiple destinations.

The Sentivist™ Sensors include two types of rules:

Built-in rules:

Built-in rules are the destinations included with Sentivist™ Server. The Sentivist™ Sensors sends alerts to Sentivist™ Server to be processed. These destinations are:

Systemlist: Writes the alerts to the Sentivist™ Server and displays them in the alert viewer. Sentivist™ Server uses this destination for alerts or events related to the Sentivist™ Server, such as those that indicate routine system activity. The moment an event occurs may not be important. However, the recorder does record the event for future reference.

Networklist: Writes the alerts to the Sentivist™ Server and displays them in the alert viewer. Sentivist™ Server uses this destination for most alerts generated by packages and backends.

Auditlist: Sentivist™ Server uses this destination for audit related alerts. Only Sentivist™ Server generates auditlist entries not the Sentivist™ Sensor.

The permissions required to view the contents of these logs are:

- Auditlist: audit view
- Systemlist: alert view
- Networklist: alert view

The various components that an alert log encompasses are given below.

The Alert view window consists of several fields

- Time (alert time)
- Host
- Alert ID (alert message)
- Source (alert source)
- Alert Message

In addition to these the bottom of each log displays,

- Number of records displayed
- Alert order
- Time period

The Time field: The alert time is the time Sentivist™ Sensor recorded the alert

The Host field: The host column indicates which Sentivist™ Sensor is sending the alert.

The Alert ID: This distinguishes one alert form another and can be used to determine which package and backend generated the alert.

The Source field: The alert source is the program or processes that send the alert. There are two alert sources:

- Built-in Sources: Built-in sources are programs and processes that send alerts. The built-in sources include programs such as the authentication events in systemlist and auditlist.

- User Sources: Packages and backends (user built or NFR sourced) are user sources because packages and backends can be enabled or disabled to send alerts. Each package and backend has its own alert source and N-Code can send alerts from that source.

The Alert Message field: Sentivist™ Sensors includes two types of alert messages:

- Built-in messages: Built-in messages are the messages that are sent by the built-in sources (systemlist, networklist or auditlist). One cannot delete systemlist, networklist and auditlist.

- User Messages: User messages are triggered by packages and backends.

Alert Message Content: Each message contains message text and a message severity rating. The message severity indicates the importance of the message. Some alert facilities display the severity as a part of the message. Sentivist™ Server uses the following severities.

- Informational: A routine system event occurred

- Warning: Something unusual occurred
- Error: Something occurred that degrades the ability to collect and record information
- Attack: A potential threat occurred.

The Source IP field: The source or originator IP address that triggered the alert

The Destination IP field: The destination or target IP address of the alert.

There are no roles to which viewing permissions are granted; only user permissions exist in the TOE.

The authorized system administrator/nfr account or authorized operators with the correct permissions have the ability to overwrite existing logs by configuring the Space Management System. Authorized operators created by the authorized system administrator/nfr with view permissions can flag records as read or unread.

Permissions required to configure space management configuration are:

- Auditlist: audit configure
- Systemlist: configure
- Networklist: configure

IDS logs can not be accessed from the CLI, only the SAI.

The SAI controls to view the logs can be found by selecting the 'Status' tab and then the 'view alerts' icon. At the top of the screen the available logs are listed. Only those the user has permissions for are available in the pull-down menu.

The SAI controls for space management can be found by selecting the 'Administration' tab and then the 'space management' icon. To set the log space management parameters, the _alert package is selected and each backend represents the audit, system, and network logs.

Audit Data for Solaris and Red Hat Linux:
All auditable events generated from the command line are stored in log files in the Operating System's file hierarchy. All audited events are accessible to authorized users as defined by FMT_SMR.1. The Sentivist™ provides configurable audit functions which are used to record audit events. Auditable events include start-up and shutdown of audit functions, and access to the Sentivist™ and associated data. Associated data on the Operating System consists of configuration of date and time, and audit data configuration. The start-up and shutdown of Sentivist™ processes are also audited. Each audit event holds the following information: date and time of the event, type of event and subject or user identity.

TOE Functional Requirements Satisfied: FAU_GEN.1- INTERP - 202

Audit review
Only the authorized system administrator 'nfr' role, and authorized operators with correct permissions have the ability to read all audit data from the system. IDS data and data pertaining to the operation of the Sentivist Sensor and Server can be viewed via the Sentivist™ Administration Interface. All other audit data, specifically data pertaining to the operation and configuration through the command line interface can be viewed

through the command line interface. Audit data on the command line interface can be viewed by the nfr account by using the `sudo` command to access the various logs stored on the Sentivist Server.

All permission checks are enforced on the Sentivist Server.

All permission checks are performed on both the SAI (for a better UI) and the Sentivist™ Server. However, they are always enforced on the Sentivist™ Server.

The user interface for this request is via the SAI's Status/View Alerts as mentioned previously.

TOE Functional Requirements Satisfied: FAU_SAR.1

Restricted audit review

Sentivist™ has in place sufficient access controls to ensure that only authorized users can read audit data; all others are denied access to this data.

Access controls are as follows:

- Access control: Allows a user to change another user's access controls.
- Audit View: Allows a user to view audit messages.

These permissions are managed via the SAI and are on a per user basis. A user must have the access control permission to change them. As mentioned previously, the Sentivist™ Server stores and enforces these permissions via the SAI. The actions that the user cannot perform are not displayed to that user by the SAI.

Audit data on the command line interface can be viewed by the nfr account by using the `sudo` command to access the various logs stored on the Sentivist Server.

TOE Functional Requirements Satisfied: FAU_SAR.2

Selectable audit review

The TOE provides the means by which administrators can sort audit data based on date and time, subject identity, type of event, and success or failure of the related event. The primary means by which this is accomplished is via the Sentivist™ Administration Interface, which can pull audit data from the Server and quickly display and sort this data in a variety of ways.

Sorting is available to any user that can view the audit records. There are many options in the SAI for retrieving specified data to view and how that data are sorted.

The filters button and various pull-down menus allow a user to select what is displayed in the SAI.

The Sentivist™ Server performs the bulk of the sorting but the retrieved records can be displayed in different orders by clicking on the top of the selected columns in the SAI.
In addition, there are filtering functions for the TOE – a post-filter (simply does not send them to the SAI), and a pre-filter.

Audit data on the Operating System contains timestamps so that events can be easily sorted. Audit data on the command line interface can be viewed by the nfr account by using the `sudo` command to access the various logs stored on the Sentivist Server.

TOE Functional Requirements Satisfied: FAU_SAR.3

Selective audit

Only the authorized system administrator\nfr account or authorized operators with the correct permissions can configure audit parameters for IDS data. This must be performed via Sentivist™ Administration Interface by logging on to Sentivist™ Server. Selection/de-selection can be done based on the event type; the event type here is a package. Packages can further be granularised into backends. The following is a list of available packages via the Sentivist™ Administration Interface;

- Authentication package
- Badfiles package
- DNS package
- Finger package
- FTP package
- ICMP package
- IMAP package
- IRC package
- Miscellaneous package
- POP package
- Rcommands package
- RPC package
- Secure Log Archive package
- SMTP package
- SSH package
- TCP package
- Telnet package
- TFTP (Beta) package
- WWW package
- Distributed Denial of Service package
- LPD/LPrng package
- Policy package
- SNMP package
- Trojans and Remote Administration package
- UDP package

Audit configuration on the command line interface is covered in the Installation and Configuration Mode and is therefore outside of the evaluated configuration.

TOE Functional Requirements Satisfied: FAU_SEL.1

Guarantees of audit data availability

A user can never 'clear' an audit record. Audit records are deleted after time through Space Management functions. Space Management System can be configured by an authorized user. Deletion of records (audit and other) is recorded in an audit message summarizing the total deleted. All audit records are stored in the MySQL database resident on the Sentivist™ Server. Access to this database is granted only to the user NFR (this is the default user created during the installation of the Sentivist™ Server).

Audit records on the underlying operating system of the Sentivist Server are not cleared or deleted.

TOE Functional Requirements Satisfied: FAU_STG.2 - INTERP- 141

Prevention of audit data loss

Sentivist™ uses a MySQL database to store audit data. The space management system periodically monitors the size of the datastore. When the size exceeds the user-configured action limit an alert is sent; until the space management system determines the size of the datastore has dropped below the clearance limit. A user-defined amount of space on the database is maintained if audit storage exhaustion should occur. The user may configure the system to automatically delete the oldest records in an attempt to reclaim datastore space. The information required to configure the Space Management System is requested during the installation of the Sentivist Server. The Space Management Settings can be changed anytime using the Sentivist Administration Interface. If the system is configured this way, the space management system will step through event and alert records in chronological order. If the record's timestamp falls outside the time window specified for its associated backend the record will be deleted. This will continue until the size of the datastore drops below the clearance limit or each remaining record is within the specified time span for the associated backend. If this condition is determined to be true an alert is sent to the user. Further, the NFR Sentivist User Guide, Chapter 5, Page 3 details how the Space Management System determines the order in which alerts and events are deleted.

Audit records which can be viewed through the command line are never deleted or overwritten.

TOE Functional Requirements Satisfied: FAU_STG.4.

8.3 Identification and Authentication

Timing of Authentication

Prior to authentication a user or an appliance (Sentivist Sensor) is only authorized to initiate the authentication process and is presented with an initial login prompt before being authenticated. Before a user or an appliance is successfully authenticated, no other TSF-mediated actions can be performed on behalf of the user.

When the underlying operating system of the Sentivist Server is started, a login prompt is provided at the Sentivist Server CLI before being authenticated. The Authorized System Administrator/nfr can provide the user name/password combination and authenticate successfully. Before authentication no other TSF-mediated actions can be performed on behalf of the Authorized Administrator/nfr user.

The Sentivist™ Server enforces all permissions and authentication. User authentication is performed via the SRP protocol. The strength of the TOE's password mechanism is rated SOF-basic. The TOE incorporates user defined authentication tokens (i.e., passwords) that can be analyzed via probabilistic or permutational means. The TOE requires that the minimum password length used to authenticate be equal to or greater than 8 characters. The Strength of Function analysis is provided under Section 8.8: Strength of Function Claims.

TOE Functional Requirements Satisfied: FIA_UAU.1

Authentication failure handling

The auditing mechanisms within the Sentivist™ ensure that the system detects the number of unsuccessful authentications to the Server from the Sentivist Administration Interface(s). On any authentication failure, it increments the user's current sequential authentication failures value. When this exceeds the maximum authentication failure

value for that connection type, the Server prevents the user from authenticating, even if the authentication would normally succeed.

- The Lockout applies to the Sentivist Administration Interface and the Sentivist Server.
- Sentivist Administration Interface: This is tracked on a per user name basis. If the same user exceeds the maximum number of sequential authentication attempts, they are locked out based on their user name. This applies to the Sentivist™ Administration Interface application.

Each Sentivist Administration Interface has a configurable, non-zero, maximum number of sequential authentication values associated with it. The Server provides a management application CGI to set this configurable value for each of the interface types. It must be a value greater than zero with a default of three (if never configured). The current count of failed authentication attempts is incremented for each user or appliance every time an attempt is made. Unknown users or appliances do not have a tracking field authentication count as they are locked out by default.

Sentivist Server CLI: The authentication failure mechanism for the Server CLI is based on the underlying operating system. The lockout mechanism can be specified using the operating system utilities. The pam_tally.so module Linux operating systems maintains a count of attempted accesses, resets the count on success, and denies access if too many failed attempts occurred. Solaris has a similar mechanism that is controlled by the Comsmiths login_limit PAM module. The number of unsuccessful authentication attempts allowed before a user is locked out is a configurable number. The Common Criteria Wrapper Guide, version 1.7 states that the number of unsuccessful attempts before a lockout needs to be set somewhere in the range of one to three. For more information on using the pam_tally.so module please refer to this web resource http://www.puschitz.com/Security.shtml, or the Common Criteria Wrapper Guide.

The authentication failure mechanism is in place only for the Sentivist Administration Interface and the Sentivist Server CLI. The same mechanism has not been implemented for the Sentivist Sensor. If a lockout mechanism is in place, there is a possibility of the entire system being compromised (e.g. Denial of Service attack) during the time between when the Sensor is locked out and authorized administrator takes action to reset the lockout condition.

To unlock a user, an administrative user with configure permissions can reset the current sequential authentication failures for a user or application. The Server provides a CGI to allow a management application to retrieve the current value and to reset it to zero. This is available through the SAI's Administration/User Management tab. The name of the user must be supplied. If all users get locked out there is a command line utility to unlock the users. Instructions for unlocking users on the command line interface can be found in the Common Criteria Wrapper Guide.

TOE Functional Requirements Satisfied: FIA_AFL.1

User attribute definition

The Sentivist Server enforces the user permissions in the Sentivist system. The Sentivist Server enforces permissions for the Authorized System Administrator \ nfr account logging in locally via the console by comparing the credentials received to those in /etc/passwd. Before the Authorized System Administrator \ nfr account is granted access to the Server, a check is made via the pam_tally.so and login_limit PAM modules to see if the account has been locked out by exceeding the maximum failed login attempts. If

the account has not exceeded the number of failed login attempts, then the Authorized System Administrator \ nfr account is granted access.

The authentication application within Sentivist™ Server uses a username and an authentication token (i.e. a password) in order to authenticate a user via the Sentivist Administration Interface. This application then returns a response that contains the user's authentication status, along with the associated user's credentials. Sentivist™ Server then uses this permission set to resolve what information is to be provided to the user via the Sentivist™ Administration Interface via Guisrv. ***Guisrv*** is a broker that works between agents wishing to interact with the Server. The requesting service makes a series of HTTP POST requests to the server that encapsulates a command. The server processes the command by either running hard-coded routines or running a specified command program. The program executes, and the results are sent back to the requesting agent in the form of an HTTP response. The user interface that appears at the SAI depends on the associated user's credentials with menu options performing only those actions which the associated user is allowed to perform.

The Sentivist Sensor to Sentivist Server communication process is getserver in daemon mode. The Sentivist Sensor initiates a connection with the Sentivist Server and a secure channel is negotiated through a handshake. The getserver communication protocol with the Sensor consists of a series of commands and responses.

getserver protocol uses a six-step handshake to authenticate between the client and the server. This authentication mechanism uses a shared secret. This shared secret is created upon installation of the Sentivist Server and stored in the central.cfg file on the Sentivist Server. The same is provided to the Sensors in two ways, via a configuration floppy or by accessing the Sentivist Sensor Management Menu. This shared secret is known to the Sensor Administrator and is stored in the remote.cfg file on the Sentivist Sensor. Only this role has the privilege to access the Sentivist Sensor Management Menu.

The Sentivist™ Server via the Sentivist™ Administration Interface provides a User Interface to view and edit user attributes, including permissions and passwords.

TOE Functional Requirements Satisfied: FIA_ATD.1

Timing of identification

Prior to identification a user or an appliance (Sentivist Sensor) is only authorized to initiate the authentication process and is presented with an initial login prompt before being identified. Before a user or an appliance is successfully identified, no other TSF-mediated actions can be performed on behalf of the user.

TOE Functional Requirements Satisfied: FIA_UID.1

8.4 Security Management

Management of security functions behavior

The protection mechanisms within Sentivist™ provide assurance that only an authorized system administrator/nfr or authorized system administrators endowed with the required permissions are allowed to modify the system data collection, analysis and reaction functions. All remote access by subjects to objects is mediated by Sentivist™ Server which in turn interfaces with the underlying Linux/Solaris Operating Systems. Once the authorized system administrator is authenticated his/her permissions are checked by Sentivist™ Server. In addition, only the authorized system administrator/nfr and

authorized operators are authorized to modify data collection behavior, analysis and reactions to those events. The authorized administrator (Operating System root account) is responsible for the installation and operation of the Sentivist™ Server, including the configuration of the system, users, and auditing. – which is covered in the Installation and Configuration Mode.

The Sentivist™ has its own concept of roles. The Server authenticates all management applications via the Sentivist™ Administration Interface user, password and a set of permissions. These users are independent of the underlying operation system.

During installation of the Sentivist™, a default authorized system administrator/nfr is created to ensure that at least on user can manage the Server and add additional authorized system administrators as necessary.

There are two protections:

- Operating System: This is provided by the OS and setup during installation. The TOE is installed so no other OS users (except root) are able to change any data or setup.

- MySQL: The DB server is protected via a user/password at install. No access from applications external to the Server is permitted. Access is restricted to a single database user. The database user password is stored in a file and accessible to the authorized system administrator/nfr only via the Sentivist Server CLI. This file is protected by file permissions.

Guisrv acts as a secure broker for requests between the clients and the Sentivist™ Server. It works as an agent controlling the communications between the clients (management applications) and the Sentivist Server. The requesting service makes a series of HTTP POST requests to the server that encapsulates a command. The Server processes the command by either running hard-coded routines or running a specified command program. The program executes, and the results are sent back to the requesting agent in the form of an HTTP response. The difference between **guisrv** and a normal web server is that it supports authentication, wraps the password in encryption, and folds all data into an encrypted envelope.

Guisrv supports three cryptographic features:

- Encryption to protect access to the system using Secure Remote Password (SRP-6) algorithm.
- Encryption to protect data within the “envelope” using AES (Rijnadel) 128-bit in Cipher Feedback Mode.
- Calculating and attaching a MAC to the text inside the envelope to verify the correctness of the text after decryption.

The encryption protocol is broken into two major pieces: the authentication handshake, which verifies the user’s password and authority to connect to the server, and the envelope protocol, which encrypts the message between the client and server.

The Sensor to Server communication process is getserver in daemon mode. The Sensor initiates a connection with the Server and a secure channel is negotiated through a handshake. The spooler process on the Sensor, once a connection is established, pushes alerts and events out to getserver on the Sentivist™ Server, which in-turn sends it to another daemon program for local storage.

The Sentivist Sensor Management Menu can be accessed by attaching a keyboard and monitor to the Sentivist Sensor and providing a password for identification and

authentication to the menu. The password used to authenticate to the Sentivist Sensor Management Menu is created upon product installation which is part of the Sentivist Sensor Installation Mode. The Installation Mode is outside of the evaluated configuration of the TOE.

The Sentivist Sensor to Sentivist Server communication process is getserver in daemon mode. The Sentivist Sensor initiates a connection with the Sentivist Server, and a secure channel is negotiated through a handshake. The spooler process on the Sensor, once a connection is made, pushes alerts and events out to getserver on the Server, which sends it to another daemon program for local storage.

The Sentivist Server initiates a connection to the Sentivist Sensor in order to "mirror out" or push packages down to the appliance. The mirror process is a series of getserver/put commands. When all the packages are pushed, getserver initiates a package_sync on the Sentivist Sensor to cause the packages to load.

The getserver communication protocol with the Sensor consists of a series of commands and responses. getserver uses a six-step handshake to authenticate between the client (Sensor) and the Server. It takes the following steps to establish a connection between systems. The client is *listening* on to a socket and waiting for a connection. The *server* actively opens a connection and connects to the client. The client, after authentication, receives data from the server.

TOE Functional Requirements Satisfied: FMT_MOF.1

Management of TSF data

The protection mechanisms within Sentivist™ ensure that only an authorized administrator has sufficient privileges to query and modify all other TOE data. No un-authorized user can add system and audit data.

The authorized system administrator \ nfr account and authorized operators with a combination of configure, audit configure, query, access control, audit view, or alert view permissions are allowed to query and modify data on the Sentivist Administration Interface.

Below is a list of permissions that are granted to the authorized system administrator 'nfr' account on the Sentivist Administration Interface. The authorized operators created by the authorized system administrator can have some or all of these permissions depending on the level of access.

- **Access Control**: Allows the user to add and delete user accounts, change user permissions, set authentication attempts for users, and unlock users. This is the highest possible level of permission, which should only be assigned to one other User in addition to the Authorized administrator or nfr user.
- **Configure**: The configure permissions controls access to the Administration, Packages, and Status shortcut bars within Sentivist Administration Interface, allowing the user to change the configuration of various components. Enabling this capability allows a user to:
  - **Configure Variables**
  - **Configure Alerts**
  - **Configure Packages and Backends**
  - **Mirror package and backend configurations**
  - **Update package and backends**

- **Diagnostics:** Allows a user to retrieve diagnostic files from Sentivist Server. This feature is rarely used. Diagnostics is usually done at the request of NFR Security Support to aid the user in troubleshooting extreme cases.
- **Audit Configure:** Allows the user to configure audit-related alerts and functions. Users with this permission have the ability to change the rules that process security- related events.
- **Audit View:** Allows the user to see audit related alerts (records written to the auditlist) through the Alert Window or popup windows. The auditlist is designed to track events related to internal security.
- **Query:** The query permission allows access to the Packages shortcut bar. This allows the user to query the data recorded by all packages and backends.
- **Alert View:** The alert view permission allows viewing of systemlist and networklist alerts. In addition, the permission allows the user to receive alerts via the popup window.
- **Restart Sentivist Sensor:** This permission allows quick or full reboot of a Sentivist Sensor connected to a Sentivist Server
- **Sensor Box:** This allows the user access to the Sensor for the sole purpose of using the Run Mirror command to push data out to the Sensor.

The authorized administrator and authorized system administrator \ nfr account are the only roles able to query and modify data using the command line interface.

TOE Functional Requirements Satisfied: FMT_MTD.1

Specification of Management Functions

The TOE manages system and audit data through the Sentivist Administration Interface. Access to the Administration Interface is controlled by the authentication and access control mechanisms that the TOE provides and implements. The ability to manage TOE data is defined by the roles explained in the above section (FMT_MTD.1). There are two different types of roles that have access to the Administration Interface: the Authorized System Administrator \ nfr account and Authorized Operators. The Authorized System Administrator \ nfr account and those Authorized Operators with sufficient permissions can manage the way that system alert and audit data is collected and actions to be taken after the data is collected.

TOE Functional Requirements Satisfied: FMT_SMF.1

Security roles

Sentivist™ maintains three types of roles: Authorized Administrator, Authorized System Administrators, Authorized Operator.

- **Authorized Administrator:** Operating system administrator for the Sentivist™ Server, this person is responsible for the installation and configuration of the Sentivist™ Server and the underlying operating system.

- **Authorized System Administrators**: This default authorized system administrator role created during the installation of the Sentivist™ Server "nfr" has all the privileges. The default user name for this role is "nfr". This role will be referred to as the "nfr" role in this document. This role is created with all of the Access Control, Audit View, Audit Configure, Configure, Alert View, Restart Sensor, Query and Diagnostic privileges. This role is the primary role

through which one can log into the Sentivist Administration Interface and create Authorized Operators, who are defined as having equal or lesser privileges than the "nfr" account. In addition to this, the "nfr" account also performs administrative functions on the Sentivist Sensor via the Sentivist Server CLI; this is accomplished during the Operations Mode of the Sentivist Sensor. It is important to note that Authorized Operators are not to log into the Command Line Interface.

- The authorized system administrator 'nfr' role created during installation of the Sentivist Server has the ability to login to the Server via the SAI and create Authorized Operators with equal or lesser privileges. The Authorized Operators are created from the Sentivist Administration Interface Console. Permissions control what an Authorized Operator can see and do within the Sentivist Administration Interface. The SAI displays only the interface functions that the authorized system administrator is authorized to use. Due to the sensitive nature of the Sentivist system, it is advised that a user be assigned with the minimum level of permissions required by that user. The following permissions can either be enabled or disabled, in any combination.

  The authorized system administrator 'nfr' role can use the `sudo` command from the command line interface to be granted escalated privileges and act as the authorized administrator and view audit records.

  Below is a list of permissions that are granted to the authorized system administrator 'nfr' account on the Sentivist Administration Interface. The authorized operators created by the authorized system administrator can have some or all of these permissions depending on the level of access.

  - **Access Control**: Allows the user to add and delete user accounts, change user permissions, set authentication attempts for users, and unlock users. This is the highest possible level of permission, which should only be assigned to one other User in addition to the Authorized administrator or nfr user.
  - **Configure**: The configure permissions controls access to the Administration, Packages, and Status shortcut bars within Sentivist Administration Interface, allowing the user to change the configuration of various components. Enabling this capability allows a user to:
    - **Configure Variables**
    - **Configure Alerts**
    - **Configure Packages and Backends**
    - **Mirror package and backend configurations**
    - **Update package and backends**
  - **Diagnostics:** Allows a user to retrieve diagnostic files from Sentivist Server. This feature is rarely used. Diagnostics is usually done at the request of NFR Security Support to aid the user in troubleshooting extreme cases.
  - **Audit Configure:** Allows the user to configure audit-related alerts and functions. Users with this permission have the ability to change the rules that process security- related events.
  - **Audit View:** Allows the user to see audit related alerts (records written to the auditlist) through the Alert Window or popup windows. The auditlist is designed to track events related to internal security.
  - **Query:** The query permission allows access to the Packages shortcut bar. This allows the user to query the data recorded by all packages and backends.

- **Alert View:** The alert view permission allows viewing of systemlist and networklist alerts. In addition, the permission allows the user to receive alerts via the popup window.
- **Restart Sentivist Sensor:** This permission allows quick or full reboot of a Sentivist Sensor connected to a Sentivist Server
- **Sensor Box:** This allows the user access to the Sensor for the sole purpose of using the Run Mirror command to push data out to the Sensor.

During the Installation and Configuration mode of the Sentivist Sensor a role namely the Sensor Administrator account is created with the privilege to access the Sentivist Sensor Management Menu. The Sensor Management menu is accessible via a serial port or direct keyboard and monitor connection to the Sensor. The Sensor Administrator once logged into the Sensor Management Menu can access the following functions and manually configure the Sensor the same configuration can be carried out using a hand crafted configuration floppy. The installation and configuration mode of the Sentivist Sensor is not a part of the evaluated configuration of the TOE.

- **Name of this Station:** A unique name for the Sentivist Sensor
- **Management Interface:** The NIC designated as the Management NIC
- **IP Address of the Sensor:** The IP address of the Sentivist Sensor
- **Network Mask of the Sensor:** The network mask for this Sentivist Sensor's IP Address.
- **Default Router:** The IP address for the default router for the Sentivist Sensor's network
- **IP Address of the Sentivist Server:** The IP address of the Sentivist Server used to manage this Sentivist Sensor.
- **Encryption Passphrase:** The passphrase used to encrypt communication between the Sentivist Sensor and the Sentivist Server.
- **Time:** The path to the time zone file
- **Administrator Password:** The password used to access the Management Menu.
- **License Key:** The unique license key for the Sentivist Sensor.

The Sensor Administrator can access/change all of the above mentioned functions but cannot do any actual configuration changes such as changing package and backend configurations to the Sentivist Sensor.

We note that the authorized administrator account is used for the setup and maintenance of the operating system on which Sentivist™ Server resides. The authorized administrator is created before installing Sentivist™ Server software on the operating system.

TOE Functional Requirements Satisfied: FMT_SMR.1

8.5 Protection of the TOE Security Functions

The Sentivist™ Server enforces all permissions and authentication functionality in the TOE. Authentication is performed via the SRP protocol. Secure Remote Password (SRP) protocol is used for negotiating secure connections and exchanging keys. It's considered a verifier based algorithm, as it uses a password verifier as a key to several of the computations.

Secure Remote Password (SRP) is a cryptographic key-exchange protocol used for negotiating secure connections and exchanging keys without the need of a certificate or key server. It's considered a *verifier-based* algorithm, as it uses a password verifier as a key to several of the computations.

This mechanism is suitable for negotiating secure connections using a user-supplied password, while eliminating the security problems traditionally associated with reusable passwords. This protocol also performs a secure key exchange in the process of authentication, allowing security layers (privacy and/or integrity protection) to be enabled during the session. Trusted key servers and certificate infrastructures are not required, and management clients are not required to store or manage any long-term keys.

The Management client (SAI) and Sentivist Server are the same as the *user* and *host* parties in a direct authentication protocol. The terms ***password*** **and** ***verifier*** correspond to conventional private and public keys, differing in only two aspects: Unlike typical private keys, the password has limited entropy, constrained by the memory of the user. A verifier has similar mathematical properties to a public key, since it is easily computed from the password; yet deriving the password from the verifier is computationally infeasible. Instead of being a publicly-known quantity, however, the verifier is kept secret by the server. An authentication mechanism that requires the server to store a copy of the user's password or private key is known as a ***plaintext-equivalent*** **mechanism**, while one that only requires a verifier to be stored will be called a ***verifier-based*** **mechanism**.

Verifier-based protocols have a significant advantage over ones that are plaintext-equivalent. A system that uses plaintext-equivalent authentication becomes instantly compromised once the **password database** is revealed, since every **user's password** is stored there. A **database of verifiers**, on the other hand, can be protected just as easily and effectively as a database of plaintext-equivalent passwords, except that failure of said protection is not as catastrophic if only verifiers are compromised.

It involves the generation of large random numbers and passing numbers across the wire between client and server. It also stores the password in a form that is not *plaintext-equivalent*, i.e, the stored version has no plaintext equivalent to itself, so the password is secure to forward attacks.

The Secure Remote Password algorithm uses the following notation:

Abbreviations
N – A large, safe prime g – A generator modulo N v – Password verifier u – Scrambling parameter H() – Hash function S – session key M – MAC'd session key

Password Verification Storage

The password or a variation on the password is never stored on the disk. Instead, the SRP algorithm stores a *password verifier*, computed using the SHA of randomly generated salt, the username and the original raw password as given by the user. The passwords are saved to disk in the file format:

<username, password verifier, salt, lockout>

In no form is the password saved to disk. The verifier is an **256-bit integer** computed using the following:

Verifier Computation
<salt> = random() x = SHA-1(<salt> | SHA-1(<username> ":" <raw password>)) <password verifier> = v = g^x % N

The | indicates concatenation and % is the modulo operation. The 160-bit result of the SHA-1 operation is implicitly converted to an integer before it is operated upon. Neither the password nor the password verifier is ever passed plaintext from client to server and back.

The Sentivist Sensor to Sentivist Server communication process is getserver in daemon mode. The Sentivist Sensor initiates a connection with the Sentivist Server, and a secure channel is negotiated through a handshake. The spooler process on the Sensor, once a connection is made, pushes alerts and events out to getserver on the Server, which sends it to another daemon program for local storage. The Sentivist Sensor Management Menu can be accessed by attaching a keyboard and monitor to the Sentivist Sensor and providing a password for identification and authentication to the menu.

getserver protocol uses a six-step handshake to authenticate between the Sensor and the Server. This authentication mechanism uses a shared secret. This shared secret is created upon installation of the Sentivist Server and stored in the Sentivist Server. The same is provided to the Sensors in two ways, via a configuration floppy or by accessing the Sentivist Sensor Management Menu. This shared secret is known only to the Administrator/NFR User. Only this role has the privilege to access the Sentivist Sensor Management Menu.

The Sentivist Server initiates a connection to the Sentivist Sensor in order to "mirror out" or push packages down to the appliance. The mirror process is a series of getserver/put commands. When all the packages are pushed, getserver initiates a package_sync on the Sentivist Sensor to cause the packages to load.

The getserver communication protocol with the Sensor consists of a series of commands and responses. getserver uses a six-step handshake to authenticate between the client (Sensor) and the Server. It takes the following steps to establish a connection between systems. The client is *listening* on to a socket and waiting for a connection. The *server* actively opens a connection and connects to the client. The client, after authentication, receives data from the server.

- (Server) Make a connection with the Sentivist Server.
- (Client) Send a header and the hostname of the Sentivist Server. Begin the handshake, and read the shared secret from the configuration file on disk.
- (Server) Send header information about the hostname of the server. Begin the handshake, and read the shared secret from the configuration file on the disk.
- (Client) Create a challenge by generating 40 random bytes and computing the SHA-1 digest of the data. Hex-encode the challenge and send it preceded by the string "challenge."

- (Server) Receive a challenge and hex-decode it. Generate an SHA-1 digest of the challenge and a shared secret passphrase. Hex-encode it and send the result back preceded by the string "response ". Generate an SHA-1 digest of the challenge and a shared secret passphrase.
- (Client) Receive the response and hex-decode it. Compare our challenge/passphrase combination with what we received from the server. They must be identical.
- (Server) Create a new challenge by generating 40 new random bytes and computing their SHA-1 digest. Encode the challenge and send it preceded by the string "challenge ". Generate a SHA-1 digest of the challenge with our second secret passphrase.
- (Client) Receive challenge and hex-decode it. Generate a SHA-1 digest of the challenge and a second secret passphrase. Hex-encode and send the result back preceded by the string "response ".
- (Server) Receive the response and hex-decode it. Compare our challenge/passphrase combination with what we received from the client. They must be identical. Respond with an "all clear" message.
- (Client) Send the message crypt to switch into crypto mode.
- (Server) Receive crypt message and activate crypto for all further communication. Send an acknowledgement.

Once this handshake is successfully completed, the system is authenticated and uses AES-128 CFB for all future communications.

Inter-TSF availability within a defined availability metric

The availability of audit and system data provided to a remote trusted IT product is dependent on a number of factors. The two most dominating factors are the performance of the intervening network and the performance of both the TSF and the remote trusted IT product. Because the performance of a given network is architecturally specific we assume that the network is operational and provides no latency in the communications path. We also assume that both IT products are operating within their specified parameters. Sentivist™ has been designed to provide fast and efficient analysis and reporting of all system data. If all the above factors are ensured, data will be provided within 60 seconds of initiating the request or action.

TOE Functional Requirements Satisfied: FPT_ITA.1

Inter-TSF confidentiality during transmission

Sentivist™ uses cryptographic mechanisms to ensure the confidentiality of all data transmitted to any remote trusted IT product. The Sentivist™ uses a NFR proprietary application protocol data exchange to ensure confidentiality of the data channel between the Sentivist™ Administration Interface, Sentivist™ Server and the Sentivist™ Sensor. The transmitted data is encrypted using AES to ensure confidentiality of the transmitted data.

TOE Functional Requirements Satisfied: FPT_ITC.1

Inter-TSF detection of modification

The Sentivist™ uses a NFR proprietary application protocol data exchange to provide integrity for the data transmitted between the Sentivist™ Sensor, Sentivist™ Server and the Sentivist™ Administration Interface. If verification fails, the current function (e.g.

authentication, data reception) is halted and the connection terminated. If modification is detected the TOE will discard the packet and send an alarm as well as generate an audit message. The transmitted data is encrypted and HMAC'd (Hashed Message Authentication coded) to ensure confidentiality and integrity of the transmitted data.

TOE Functional Requirements Satisfied: FPT_ITI.1

Non-bypassability of the TSP

Sentivist™ has sufficient protection mechanisms to ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed. All non-physical access to the system is mediated by the TOE which acts as a reference monitor and therefore the TOE shall validate all actions between subjects and objects that require policy enforcement, before allowing the action to succeed.

When a user wishes to log into the Server and access its features, an authentication handshake is kicked off by the SAI. The corresponding algorithm computes sets of numbers and compares them as session keys to verify the correctness of both sides of the connection before sending any actual data. Once these steps are completed successfully, the authorized user has access to functions as specified by their roles. Similarly, a secure communication channel is established between the Server and the Sensor.

The permission-enforcement entity, Sentivist Server compares the user access level for all incoming requests to the access level of the user role provided and maintained. If the access level of the requested action is greater than the current user's access level, the requested action is denied. The Server receives and sends data at the Sentivist Server Port and the Sensor sends and receives data at the Sentivist Sensor port. The relevant security measures that are taken at these access points to ensure non-bypassability are described below:

Sentivist Sensor Network Interface:

The Sentivist Sensor Network Interface handles traffic from the network that the Sensor is designated to monitor and sends relevant data to the Server.

The Sentivist Sensor Network Interface consists of a single process, nfrd. nfrd monitors the network packets from the target IT system on a dedicated Network Interface Card, separate from the one used to communicate with the Sentivist Server. nfrd consumes raw packets for analysis within the TOE. The Sensor Network Interface runs in promiscuous mode, meaning that the TOE does not respond to any network traffic on the network it monitors. The network traffic is never executed, but rather is safely copied to memory and parsed for analysis.

Sentivist Sensor Management Interface:

There is a single process managing the communications between the Sentivist Sensor appliances to the Sentivist Server. This is the Sentivist Sensor Management Interface on the Sensor. Its primary function is to create an encrypted channel and receive data to process and store. The communication protocol between the Sentivist Server and the Sentivist Sensor consists of a set of commands and responses. The Sentivist Server waits for a connection request from an appliance. When it receives a request from an appliance, the two exchange header information containing their user-assigned names. When both have opened a channel, they start an authentication handshake. If successful, the communications switch to an encrypted mode and all data is encrypted with AES.

Sentivist Sensor Console Port:

The Sentivist Sensor Console port is used only during the Installation and Configuration Mode of the Sentivist Sensor, this mode is not a part of the evaluated configuration of the TOE. The Sentivist Sensor port is used to provide a local console to make both security and non security relevant changes and monitor performance.

The Sensor management menu cannot be used to change any management functions. The Sensor Administrator can access/change all of the below mentioned functions but cannot do any actual configuration changes such as changing package and backend configurations to the Sentivist Sensor.

The Sensor administrator has access to the Sentivist Sensor Management Menu through the console port. Once logged into the menu this entity can make changes to the various Sensor configuration parameters such as the name of the Sensor, Management Interface, IP address of the Sensor, Network Mask of the Sensor, Default Router, IP address of the Sentivist Server, Encryption Passphrase, Time, Administrator Password and the License Key.

Sentivist Administration Interface
The Sentivist Administration Interface is installed on a Windows machine that is local-login only, not a part of any domains, and is used only for communicating with the Sentivist Server. The Windows operating system on the machine hosting the Sentivist Administration Interface is not security-enforcing in its evaluated configuration. However, it has been included in the TOE Boundary to eliminate the potential for bypassing the security policy. If it were outside of the TOE Boundary it might be configured or managed in such a way that is not consistent with the physical and personnel assumptions for the TOE.

TOE Functional Requirements Satisfied: FPT_RVM.1

TSF domain separation
The Sentivist™ provides access controls to enforce the security policy of the TOE. This includes Access Control mechanisms that enforce separation between the security domains of subjects. The TOE executes all of its processes internally. It is accessible only via the defined interfaces and only authorized users are able to modify the functioning of the TOE.

The Sentivist Sensor interface is a dedicated physical and logical interface that is associated with network interface ports and used to monitor network packets from the target IT system. It receives raw packets for analysis within the TOE. This interface enforces domain separation in that any data sent to this interface (which is presumed untrusted) is logically separated from all other TOE data. The Sensor Interface runs in promiscuous mode, meaning that the TOE does not respond to any network traffic on the network it monitors. The network traffic is never executed, but rather is parsed for analysis. Traffic flowing through the TOE is subject to the policies as defined by the authorized users. Finally, the TOE's management interface is on a physically protected network. At all physical interfaces, the TOE intercedes to ensure domain separation. Traffic and/or unauthorized users cannot bypass the identification and authentication mechanisms, preventing interference and tampering by untrusted subjects and thereby maintaining a domain for its own execution.

The Sentivist Server is installed on a system that is available only to the authorized system administrator who installs it and access to the OS on which the Server is installed is protected by the OS access control mechanisms. The OS which runs the Server runs

only the processes of the Server and no other user processes are executed, Traffic flowing through the TOE is subject to the policies as defined by the authorized administrators. The Sensor runs on an operating system that doesn't provide any user account management functionality and runs only the processes that are required by the Sensor.

The Sentivist Administration Interface is installed on a Windows machine that is local-login only, not a part of any domains, and is used only for communicating with the Sentivist Server. The Windows operating system on the machine hosting the Sentivist Administration Interface is not security-enforcing in its evaluated configuration. However, it has been included in the TOE Boundary to eliminate the potential for bypassing the security policy. If it were outside of the TOE Boundary it might be configured or managed in such a way that is not consistent with the physical and personnel assumptions for the TOE.

TOE Functional Requirements Satisfied: FPT_SEP.1

Reliable time stamps

The Sentivist Server uses time from the underlying Linux/Solaris operating systems. The Sentivist™ Server uses the OS timestamp when it adds its own events (audit and system). This is protected via OS mechanisms. The Sentivist™ Sensor uses the NTP service provided on the Server for reliable timestamps. NTP can be configured on the Sentivist Sensor and Server during the Installation and Configuration Mode.

TOE Functional Requirements Satisfied: FPT_STM.1

8.6 IDS Component Requirements (IDS)

IDS System data collection

Sentivist™ Sensor unobtrusively monitors traffic in real time for suspicious activity misuse, abuse, attacks and anomalous behavior. This is accomplished through the nfrd process which is running on the Sensor. nfrd consumes raw network data by listening on the Network Interface Card configured to monitor network traffic. Initial data collection and analysis occurs on the Sensor. This data is compared against a rule set (signatures of known attacks) that indicates typical intrusion activity. When there is a match to a signature, information on the packet (or packets) is collected by the Sentivist™ Sensor. This information is relayed to the Sentivist™ Server. The Server in turn applies rules which are configurable by an authorized administrator or operator via the Sentivist™ Administration Interface. These rules can be configured in such a way that a pop-up warning can be generated from the SAI, if a specific event is detected. These events are then written to a MySQL database residing on the Server through an alert daemon (alertd). The Sentivist™ also detects the occurrence of selected events, gathers information concerning them, and records it. Reporting features are also provided by the Sentivist, included among the reporting features are the three logs Auditlist, Systemlist and Networklist. These logs can be viewed by users who have the required permissions.

The information encompassed in these logs and other user defined logs are collected with each event includes startup/shutdown of audit functions, startup/shutdown of Sentivist processes, access to the audit data, date and time of the event, identification and authentication events, Source File, Line number, Host name, Alert ID, Source ID, Source Name, Source Description, Source Process ID, Alert Message, Severity Index, Destination IP, Source IP, service requests, network traffic, security configuration changes, detected malicious code, service configuration, detected known vulnerabilities.

Data collection is implemented by a set of packages and backends. Management of this configuration is via the Sentivist™ Server and the SAI. Packages and backends are configuration information which can be enabled or disabled based on what type of traffic flow in and out of the network, needs to be monitored by the Sensor. Packages and backends can be enabled/disabled from the Sentivist™ Administration Interface. The Server stores and distributes packages and backends to appliances. These are responsible for driving the intrusion detection and allowing for dynamic updates without changes in the appliance code. When new packages and backends are installed on the Server, the state of the package is as defined inside of the package/backend. The state can be either enabled or disabled. If a package/backend is already installed on the Server and is updated, the state of the package is as defined in the Server package/backend prior to the upgrade.

Through the Sentivist Administration Interface, alerts within packages and backends can be configured to record to the networklist or to send alerts to other locations. These locations include having specific types of alerts be sent to a list other than networklist, systemlist or auditlist. The authorized system administrator or authorized operators are able to update package and backend configuration if they have the configuration permission. They can also update the audit package/backend if they have the audit configure permission. The alert ID in conjunction with the Alert Message field describes the outcome of the event.

TOE Functional Requirements Satisfied: IDS_SDC.1

Analyzer analysis

The Sentivist™ adheres to the signature analysis method. That is, it matches specific signatures or patterns that may characterize attack attempts to a data base of known attacks. This data base can be updated and user customized to provide up to date coverage of known attacks. The Sentivist™ also employs protocol analysis and anomaly detection techniques to determine previously undiscovered attacks, checks are made to capture unexpected activity and deviations from standards specified in the RFCs.

All data is stored on the Sentivist™ Server in the form of events. The contents of these events change based on the package/backend configuration. A user can also change these and modify them themselves.

TOE Functional Requirements Satisfied: IDS_ANL.1

Analyzer react

The Sentivist™ Sensor unobtrusively monitors networks in real time for activity such as known attacks, abnormal behavior, unauthorized access attempts, and policy infringements. Sensor records information associated with suspicious activity and raises applicable alerts.

The Sensor in turn provides this information to the Server which captures, processes, and manages large volumes of event and alert data provided from multiple Sentivist™ Sensors. The processed data is stored in a local data store and made available to external management applications such as the Sentivist™ Administration Interface

The Server also provides alert notification to a standard communications channel (email, page, etc.) to a third party application, or an agent for queries on the stored alert data. By default, Sentivist™ only generates an alarm when an intrusion is detected. However, it can also be configured to perform a TCP reset on the connection in question if an

intrusion is detected. However, notification to third party applications is outside of the evaluated TOE configuration.

TOE Functional Requirements Satisfied: IDS_RCT.1

Restricted data review

Only authorized Sentivist™ users have the ability to view all event data generated by the IDS components of the system through the Sentivist™ Administration Interface. These users include authorized system administrator and authorized operators with alert view, audit view, and query privileges. These are the only users that are afforded read access to this system data.

All data is presented via the Sentivist™ Administration Interface. The SAI uses a list that allows the user to drill down and pinpoint the exact alarm and analyze its threat potential. Further information can also be obtained by viewing the contents of each entry in the list.

Data viewing is enforced by the Sentivist™ Server based on a user's permissions. Query permission allows a user to look at event data. View alert and view audit permissions allow an authorized user to look at alert and audit data.

TOE Functional Requirements Satisfied: IDS_RDR.1

Guarantee of System Data Availability

Sentivist™ Server is responsible for keeping the amount of space used for Sentivist™ Sensor generated data storage within user-defined limits. The data storage limits can be configured by the user using the Sentivist™ Administration Interface.

The amount of space available for any system to store data is limited. The Space management system aims to maintain as much information that is useful to the user without exceeding a set limit on the number of bytes of disk storage used.

Sentivist™ uses a MySQL database to store audit data. The space management system periodically monitors the size of the datastore, and the time intervals are configurable. When the size exceeds the user-configured action limit an alert is sent; until the space management system determines the size of the datastore has dropped below the clearance limit. A user-defined amount of space on the database is maintained if audit storage exhaustion should occur. The user may configure the system to automatically delete the oldest records in an attempt to reclaim datastore space. The information required to configure the Space Management System is requested during the installation of the Sentivist Server. The Space Management Settings can be changed anytime using the Sentivist Administration Interface. If the system is configured this way, the space management system will step through event and alert records in chronological order. If the record's timestamp falls outside the time window specified for its associated backend the record will be deleted. This will continue until the size of the datastore drops below the clearance limit or each remaining record is within the specified time span for the associated backend. If this condition is determined to be true an alert is sent to the user. Further, the NFR Sentivist User Guide, Chapter 5, Page 3 details how the Space Management System determines the order in which alerts and events are deleted.

Action limit is specified as the percentage of the maximum space used before the Space Management System begins removing aged alerts and events. The default value for the action limit is 95% of the maximum space used.

Clearance limit is specified as the percentage of the maximum space at which the Space Management System will stop removing aged alerts and events. The default value for the clearance Limit is 90 % of the maximum space used.

Audit records are deleted after time through space management functions. Space Management System can be configured by an authorized administrator or operator. Deletion of records (audit and other) is recorded in an audit message summarizing the total deleted. All audit records are stored in the MySQL database resident on the Sentivist™ Server. Access to this database is granted only to the Administrator/NFR User.

Sentivist™ has in place sufficient access controls, as described above, to ensure that only authorized administrators and users with read/query privileges can view IDS event data; all other users are prohibited from this. No administrator or user is allowed to delete the IDS event data. Sentivist™ mediates all access requests from subjects to objects. Sentivist™ therefore acts as a reference monitor and insures that no subject can bypass its controls and directly access the subsystem.

TOE Functional Requirements Satisfied: IDS_STG.1

Prevention of System data loss

Sentivist™ uses a MySQL database to store audit data. The space management system periodically monitors the size of the datastore. When the size exceeds the user-configured action limit an alert is sent; until the space management system determines the size of the datastore has dropped below the clearance limit. A user-defined amount of space on the database is maintained if audit storage exhaustion should occur. The user may configure the system to automatically delete the oldest records in an attempt to reclaim datastore space. The information required to configure the Space Management System is requested during the installation of the Sentivist Server. The Space Management Settings can be changed anytime using the Sentivist Administration Interface. If the system is configured this way, the space management system will step through event and alert records in chronological order. If the record's timestamp falls outside the time window specified for its associated backend the record will be deleted. This will continue until the size of the datastore drops below the clearance limit or each remaining record is within the specified time span for the associated backend. If this condition is determined to be true an alert is sent to the user. Further, the NFR Sentivist User Guide, Chapter 5, Page 3 details how the Space Management System determines the order in which alerts and events are deleted.

The access to this database is through userID/ password authentication by the administrator/NFR User.

TOE Functional Requirements Satisfied: IDS_STG.2

8.7 TOE Security Assurance Measures

The Sentivist™ v4.0.2 – Updated to v4.0.6 was developed with the following security Assurance measures in place, which constitutes a Common Criteria EAL 2 level of assurance.

- Configuration management
- Delivery and operation
- Development

- Guidance documents
- Tests
- Vulnerability assessment

This section of the ST provides a mapping demonstrating that the Assurance Measures listed meet the Assurance Requirements necessary to achieve an EAL 2. In this case the specification of assurance measures is done by referencing the appropriate documentation. Analysis of the referenced documentation to ensure that the documentation listed meets the requirements of the Assurance Requirements for EAL 2. In the TOE, there are no non-administrative user interfaces i.e. the user is not provided with a direct interface to the product or an account to use to log into the product. Hence, the requirement for user guidance (AGD_USR.1) does not apply since the TSF does not provide any interfaces for direct use by non-administrative users. Thus the AGD_USR.1 requirement is vacuously satisfied. (PD-0106: Situations where AGD_USR may be vacuously satisfied)

Table 4 – Assurance Measures Mapping to SARs

CC Assurance Requirements	NFR Sentivist™ v4.0.2 – Updated to v4.0.6
ACM_CAP.2	Configuration Management v1.7
ADO_DEL.1	Secure Delivery Guidance v1.6
ADO_IGS.1	Common Criteria Wrapper Guide v1.7
ADV_FSP.1	Functional Specification v1.8
ADV_HLD.1	High-level Design Document v1.6
ADV_RCR.1	Functional Specification v1.8 High-level Design Document v1.6
AGD_ADM.1	NFR Sentivist Getting Started Guide NFR Sentivist Users Guide NFR Sentivist Server v4.0.2 Release Notes NFR Sentivist Server v4.0.6 Release Notes NFR Sentivist Sensor v4.0.2 Release Notes NFR Sentivist Administration Interface v4.0.2 Release Notes
AGD_USR.1	This requirement does not apply.
ATE_COV.1	Common Criteria Test Plan v1.0
ATE_FUN.1	Consolidated Test Plan v1.0
ATE_IND.2	NFR NID System v1.0 Test plans NFR NID System v3.0 Test results CC Lab Testing report
AVA_SOF.1	Strength of TOE Security Function Analysis v0.6
AVA_VLA.1	Vulnerability Assessment v1.0

8.8 Strength of Function Claims

The TOE (specifically, the TOE's password mechanism) minimum strength of function claim is SOF-basic. The TOE incorporates user defined authentication tokens (i.e., passwords) that can be analyzed via probabilistic or permutational means. The TOE requires that the minimum password length used to authenticate an entity be greater than 8 alphanumeric characters. We also note that passwords are case sensitive. The related security functional requirement is FIA_UAU.1: Timing of

Authentication. A Strength of Function analysis of the cryptographic algorithms used within this TOE is outside the scope of the evaluation.

9.0 PP Claims

This section provides PP conformance claims

9.1 PP Conformance

This Security Target claims conformance to the following PP:

Intrusion Detection System System Protection Profile Version 1.4. Security level EAL 2. Authored by Science Applications International Corporation, February 4, 2002.

In addition to the IT Security Requirements contained in the above mentioned Protection Profile the following has been added FMT_SMF.1-INTERP-065: Specification of Management Functions.

10 RATIONALE

This section provides the rationale for the selection of the IT security requirements, objectives, assumptions, and threats. In particular, it shows that the IT security requirements are suitable to meet the security objectives, which in turn are shown to be suitable to cover all aspects of the TOE security environment.

10.1 RATIONALE FOR IT SECURITY OBJECTIVES

This section provides a rationale for the existence of each assumption, threat, and policy statement that compose the Intrusion Detection System System Protection Profile. Table 5 Security Environment vs. Objectives demonstrates the mapping between the assumptions, threats, and polices to the security objectives is complete. The following discussion provides detailed evidence of coverage for each assumption, threat, and policy.

Table 5 - Relationship of Security Environment to Objectives

	O.PROTCT	O.IDSCAN	O.IDSENS	O.IDANLZ	O.RESPON	O.EADMIN	O.ACCESS	O.IDAUTH	O.OFLOWS	O.AUDITS	O.INTEGR	O.EXPORT	O.INSTAL	O.PHYCAL	O.CREDEN	O.PERSON	O.INTROP
A.ACCESS																	X
A.DYNMIC																X	X
A.ASCOPE																	X
A.PROTCT														X			
A.LOCATE														X			
A.MANAGE																X	
A.NOEVIL													X	X	X		
A.NOTRST														X	X		
T.COMINT	X						X	X			X						
T.COMDIS	X						X	X				X					
T.LOSSOF	X						X	X			X						
T.NOHALT		X	X	X			X	X									
T.PRIVIL	X						X	X									
T.IMPCON						X	X	X					X				
T.INFLUX									X								
T.FACCNT										X							
T.SCNCFG		X															
T.SCNMLC		X															
T.SCNVUL		X															
T.FALACT					X												
T.FALREC				X													
T.FALASC				X													
T.MISUSE			X														
T.INADVE			X				X										
T.MISACT			X				X										
P.DETECT		X	X							X							
P.ANALYZ				X													
P.MANAGE	X					X	X	X					X		X	X	

	O.PROTCT	O.IDSCAN	O.IDSENS	O.IDANLZ	O.RESPON	O.EADMIN	O.ACCESS	O.IDAUTH	O.OFLOWS	O.AUDITS	O.INTEGR	O.EXPORT	O.INSTAL	O.PHYCAL	O.CREDEN	O.PERSON	O.INTROP
P.ACCESS	X						X	X									
P.ACCACT								X		X							
P.INTGTY											X						
P.PROTCT									X					X			

A.ACCESS The TOE has access to all the IT System data it needs to perform its functions.

The O.INTROP objective ensures the TOE has the needed access.

A.DYNMIC The TOE will be managed in a manner that allows it to appropriately address changes in the IT System the TOE monitors.

The O.INTROP objective ensures the TOE has the proper access to the IT System. The O.PERSON objective ensures that the TOE will manage appropriately.

A.ASCOPE The TOE is appropriately scalable to the IT System the TOE monitors.

The O.INTROP objective ensures the TOE has the necessary interactions with the IT System it monitors.

A.PROTCT The TOE hardware and software critical to security policy enforcement will be protected from unauthorized physical modification.

The O.PHYCAL provides for the physical protection of the TOE hardware and software.

A.LOCATE The processing resources of the TOE will be located within controlled access facilities, which will prevent unauthorized physical access.

The O.PHYCAL provides for the physical protection of the TOE.

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

The O.PERSON objective ensures all authorized administrators are qualified and trained to manage the TOE.

A.NOEVIL The authorized administrators are not careless, willfully negligent, or hostile, and will follow and abide by the instructions provided by the TOE documentation.

The O.INSTAL objective ensures that the TOE is properly installed and operated and the O.PHYCAL objective provides for physical protection of the TOE by authorized administrators. The O.CREDEN objective supports this assumption by requiring protection of all authentication data.

A.NOTRST The TOE can only be accessed by authorized users.

The O.PHYCAL objective provides for physical protection of the TOE to protect against unauthorized access. The O.CREDEN objective supports this assumption by requiring protection of all authentication data.

T.COMINT An unauthorized user may attempt to compromise the integrity of the data collected and produced by the TOE by bypassing a security mechanism.

The O.IDAUTH objective provides for authentication of users prior to any TOE data access. The O.ACCESS objective builds upon the O.IDAUTH objective by only permitting authorized users to access TOE data. The O.INTEGR objective ensures no TOE data will be modified. The O.PROTCT objective addresses this threat by providing TOE self-protection.

T.COMDIS An unauthorized user may attempt to disclose the data collected and produced by the TOE by bypassing a security mechanism.

The O.IDAUTH objective provides for authentication of users prior to any TOE data access. The O.ACCESS objective builds upon the O.IDAUTH objective by only permitting authorized users to access TOE data. The O.EXPORT objective ensures that confidentiality of TOE data will be maintained. The O.PROTCT objective addresses this threat by providing TOE self-protection.

T.LOSSOF An unauthorized user may attempt to remove or destroy data collected and produced by the TOE.

The O.IDAUTH objective provides for authentication of users prior to any TOE data access. The O.ACCESS objective builds upon the O.IDAUTH objective by only permitting authorized users to access TOE data. The O.INTEGR objective ensures no TOE data will be deleted. The O.PROTCT objective addresses this threat by providing TOE self-protection.

T.NOHALT An unauthorized user may attempt to compromise the continuity of the System's collection and analysis functions by halting execution of the TOE.

The O.IDAUTH objective provides for authentication of users prior to any TOE function accesses. The O.ACCESS objective builds upon the O.IDAUTH objective by only permitting authorized users to access TOE functions. The O.IDSCAN, O.IDSENS, and O.IDANLZ objectives address this threat by requiring the TOE to collect and analyze System data, which includes attempts to halt the TOE.

T.PRIVIL An unauthorized user may gain access to the TOE and exploit system privileges to gain access to TOE security functions and data.

The O.IDAUTH objective provides for authentication of users prior to any TOE function accesses. The O.ACCESS objective builds upon the O.IDAUTH objective by only permitting authorized users to access TOE functions. The O.PROTCT objective addresses this threat by providing TOE self-protection.

T.IMPCON An unauthorized user may inappropriately change the configuration of the TOE causing potential intrusions to go undetected.

The O.INSTAL objective states the authorized administrators will configure the TOE properly. The O.EADMIN objective ensures the TOE has all the necessary administrator functions to manage the product. The O.IDAUTH objective provides for authentication

of users prior to any TOE function accesses. The O.ACCESS objective builds upon the O.IDAUTH objective by only permitting authorized users to access TOE functions.

T.INFLUX An unauthorized user may cause malfunction of the TOE by creating an influx of data that the TOE cannot handle.

The O.OFLOWS objective counters this threat by requiring the TOE handle data storage overflows.

T.FACCNT Unauthorized attempts to access TOE data or security functions may go undetected.

The O.AUDITS objective counters this threat by requiring the TOE to audit attempts for data accesses and use of TOE functions.

T.SCNCFG Improper security configuration settings may exist in the IT System the TOE monitors.

The O.IDSCAN objective counters this threat by requiring a TOE that contains a Scanner, collect and store static configuration information that might be indicative of a configuration setting change. The ST will state whether this threat must be addressed by a Scanner.

T.SCNMLC Users could execute malicious code on an IT System that the TOE monitors which causes modification of the IT System protected data or undermines the IT System security functions.

The O.IDSCAN objective counters this threat by requiring a TOE that contains a Scanner, collect and store static configuration information that might be indicative of malicious code. The ST will state whether this threat must be addressed by a Scanner.

T.SCNVUL Vulnerabilities may exist in the IT System the TOE monitors.

The O.IDSCAN objective counters this threat by requiring a TOE that contains a Scanner, collect and store static configuration information that might be indicative of vulnerability. The ST will state whether this threat must be addressed by a Scanner.

T.FALACT The TOE may fail to react to identified or suspected vulnerabilities or inappropriate activity.

The O.RESPON objective ensures the TOE reacts to analytical conclusions about suspected vulnerabilities or inappropriate activity.

T.FALREC The TOE may fail to recognize vulnerabilities or inappropriate activity based on IDS data received from each data source.

The O.IDANLZ objective provides the function that the TOE will recognize vulnerabilities or inappropriate activity from a data source.

T.FALASC The TOE may fail to identify vulnerabilities or inappropriate activity based on association of IDS data received from all data sources.

The O. IDANLZ objective provides the function that the TOE will recognize vulnerabilities or inappropriate activity from multiple data sources.

T.MISUSE Unauthorized accesses and activity indicative of misuse may occur on an IT System the TOE monitors.

The O.AUDITS and O.IDSENS objectives address this threat by requiring a TOE, that contains a Sensor, collect audit and Sensor data.

T.INADVE Inadvertent activity and access may occur on an IT System the TOE monitors.

The O.AUDITS and O.IDSENS objectives address this threat by requiring a TOE, that contains a Sensor, collect audit and Sensor data.

T.MISACT Malicious activity, such as introductions of Trojan horses and viruses, may occur on an IT System the TOE monitors.

The O.AUDITS and O.IDSENS objectives address this threat by requiring a TOE, that contains a Sensor, collect audit and Sensor data.

P.DETECT Static configuration information that might be indicative of the potential for a future intrusion or the occurrence of a past intrusion of an IT System or events that are indicative of inappropriate activity that may have resulted from misuse, access, or malicious activity of IT System assets must be collected.

The O.AUDITS, O.IDSENS, and O.IDSCAN objectives address this policy by requiring collection of audit, Sensor, and Scanner data.

P.ANALYZ Analytical processes and information to derive conclusions about intrusions (past, present, or future) must be applied to IDS data and appropriate response actions taken.

The O.IDANLZ objective requires analytical processes be applied to data collected from Sensors and Scanners.

P.MANAGE The TOE shall only be managed by authorized users.

The O.PERSON objective ensures competent administrators will manage the TOE and the O.EADMIN objective ensures there is a set of functions for administrators to use. The O.INSTAL objective supports the O.PERSON objective by ensuring administrator follow all provided documentation and maintain the security policy. The O.IDAUTH objective provides for authentication of users prior to any TOE function accesses. The O.ACCESS objective builds upon the O.IDAUTH objective by only permitting authorized users to access TOE functions. The O.CREDEN objective requires administrators to protect all authentication data. The O.PROTCT objective addresses this policy by providing TOE self-protection.

P.ACCESS All data collected and produced by the TOE shall only be used for authorized purposes.

The O.IDAUTH objective provides for authentication of users prior to any TOE function accesses. The O.ACCESS objective builds upon the O.IDAUTH objective by only permitting authorized users to access TOE functions. The O.PROTCT objective addresses this policy by providing TOE self-protection.

P.ACCACT Users of the TOE shall be accountable for their actions within the IDS.

The O.AUDITS objective implements this policy by requiring auditing of all data accesses and use of TOE functions. The O.IDAUTH objective supports this objective by ensuring each user is uniquely identified and authenticated.

P.INTGTY Data collected and produced by the TOE shall be protected from modification.

The O.INTEGR objective ensures the protection of data from modification.

P. PROTCT The TOE shall be protected from unauthorized accesses and disruptions of TOE data and functions.

The O.OFLOWS objective counters this policy by requiring the TOE handle disruptions. The O.PHYCAL objective protects the TOE from unauthorized physical modifications.

10.2 Rationale for Security Objectives for the Environment

The purpose for the environmental objectives is to provide protection for the TOE that cannot be addressed through IT measures. The defined objectives provide for physical protection of the TOE, proper management of the TOE, and interoperability requirements on the TOE. Together with the IT security objectives, these environmental objectives provide a complete description of the responsibilities of TOE in meeting security needs.

10.3 Rationale for Security Requirements

This section demonstrates that the functional components selected for the NFR Sentivist™ Security Target provide complete coverage of the defined security objectives. The following discussion provides detailed evidence of coverage for each security objective. The mapping of components to security objectives is depicted in the following table.

Table 6 - Requirements vs. Objectives Mapping

	O.PROTCT	O.IDSCAN	O.IDSENS	O.IDANLZ	O.RESPON	O.EADMIN	O.ACCESS	O.IDAUTH	O.OFLOWS	O.AUDITS	O.INTEGR	O.EXPORT
FAU_GEN.1-INTERP-202										X		
FAU_SAR.1						X						
FAU_SAR.2							X	X				
FAU_SAR.3						X						
FAU_SEL.1						X				X		
FAU_STG.2 - INTERP-141	X						X	X	X		X	
FAU_STG.4									X	X		
FIA_UAU.1							X	X				
FIA_AFL.1								X				
FIA_ATD.1								X				
FIA_UID.1							X	X				
FMT_MOF.1	X						X	X				
FMT_MTD.1	X						X	X			X	
FMT_SMF.1-INTERP-065							X	X				
FMT_SMR.1								X				
FPT_ITA.1												X

FPT_ITC.1											X	X
FPT_ITI.1											X	X
FPT_RVM.1	X					X		X		X	X	
FPT_SEP.1	X					X		X		X	X	
FPT_STM.1										X		
IDS_SDC.1		X	X									
IDS_ANL.1				X								
IDS_RCT.1					X							
IDS_RDR.1						X	X	X				
IDS_STG.1	X						X	X	X		X	
IDS_STG.2									X			

O.PROTCT The TOE must protect itself from unauthorized modifications and access to its functions and data.

The TOE is required to protect the audit data from deletion as well as guarantee the availability of the audit data in the event of storage exhaustion, failure or attack [FAU_STG.2-INTERP-141]. The System is required to protect the System data from any modification and unauthorized deletion, as well as guarantee the availability of the data in the event of storage exhaustion, failure or attack [IDS_STG.1]. The TOE is required to provide the ability to restrict managing the behavior of functions of the TOE to authorized users of the TOE [FMT_MOF.1]. Only authorized administrators of the System may query and add System and audit data, and authorized administrators of the TOE may query and modify all other TOE data [FMT_MTD.1]. The TOE must ensure that all functions are invoked and succeed before each function may proceed [FPT_RVM.1]. The TSF must be protected from interference that would prevent it from performing its functions [FPT_SEP.1].

O.IDSCAN The Scanner must collect and store static configuration information that might be indicative of the potential for a future intrusion or the occurrence of a past intrusion of an IT System.

A System containing a Scanner is required to collect and store static configuration information of an IT System. The type of configuration information collected must be defined in the ST [IDS_SDC.1].

O.IDSENS The Sensor must collect and store information about all events that are indicative of inappropriate activity that may have resulted from misuse, access, or malicious activity of IT System assets and the IDS.

A System containing a Sensor is required to collect events indicative of inappropriate activity that may have resulted from misuse, access, or malicious activity of IT System assets of an IT System. These events must be defined in the ST [IDS_SDC.1].

O.IDANLZ The Analyzer must accept data from IDS Sensors or IDS Scanners and then apply analytical processes and information to derive conclusions about intrusions (past, present, or future).

The Analyzer is required to perform intrusion analysis and generate conclusions [IDS_ANL.1].

O.RESPON The TOE must respond appropriately to analytical conclusions.

The TOE is required to respond accordingly in the event an intrusion is detected [IDS_RCT.1].

O.EADMIN The TOE must include a set of functions that allow effective management of its functions and data.

The TOE must provide the ability to review and manage the audit trail of the System [FAU_SAR.1, FAU_SAR.3, and FAU_SEL.1]. The System must provide the ability for authorized administrators to view all System data collected and produced [IDS_RDR.1]. The TOE must ensure that all functions are invoked and succeed before each function may proceed [FPT_RVM.1]. The TSF must be protected from interference that would prevent it from performing its functions [FPT_SEP.1].

O.ACCESS The TOE must allow authorized users to access only appropriate TOE functions and data.

The TOE is required to restrict the review of audit data to those granted with explicit read-access [FAU_SAR.2]. The System is required to restrict the review of System data to those granted with explicit read-access [IDS_RDR.1]. The TOE is required to protect the audit data from deletion as well as guarantee the availability of the audit data in the event of storage exhaustion, failure or attack [FAU_STG.2-INTERP-141]. The System is required to protect the System data from any modification and unauthorized deletion [IDS_STG.1]. Users authorized to access the TOE are defined using an identification and authentication process [FIA_UID.1, FIA_UAU.1]. The TOE is required to provide the ability to restrict managing the behavior of functions of the TOE to authorized users of the TOE [FMT_MOF.1]. Access to the TOE and TOE data is controlled by the authentication and access control mechanisms that the TOE provides and implements. [FMT_SMF.1-INTERP-065]. Only authorized administrators of the System may query and add System and audit data, and authorized administrators of the TOE may query and modify all other TOE data [FMT_MTD.1].

O.IDAUTH The TOE must be able to identify and authenticate users prior to allowing access to TOE functions and data.

The TOE is required to restrict the review of audit data to those granted with explicit read-access [FAU_SAR.2]. The System is required to restrict the review of System data to those granted with explicit read-access [IDS_RDR.1]. The TOE is required to protect the stored audit records from unauthorized deletion [FAU_STG.2-INTERP-141]. The System is required to protect the System data from any modification and unauthorized deletion, as well as guarantee the availability of the data in the event of storage exhaustion, failure or attack [IDS_STG.1]. Security attributes of subjects use to enforce the authentication policy of the TOE must be defined [FIA_ATD.1]. Users authorized to access the TOE are defined using an identification and authentication process [FIA_UID.1, FIA_UAU.1, and FMT_SMF.1-INTERP-065]. The TOE will monitor the number of unsuccessful authentication attempts and, when the defined limit of unsuccessful attempts has been reached, prevent successful authentication for that entity until an authorized administrator takes some action to make authentication possible [FIA_AFL.1]. The authentication failure mechanism is in place for the Sentivist Administration Interface and the Sentivist Server CLI. The number of unsuccessful authentication attempts before a user is locked out is recommended to be preset to three. The same mechanism has not been implemented for the Sentivist Sensor because if a lockout mechanism is in place there is a possibility of the entire system being compromised (e.g. Denial of Service attack) during the time between when the Sensor is locked out and authorized administrator takes action to reset the lockout condition. .The TOE is required to provide the ability to restrict managing the behavior of functions of the TOE to authorized users of the TOE [FMT_MOF.1]. Only authorized administrators

of the System may query and add System and audit data, and authorized administrators of the TOE may query and modify all other TOE data [FMT_MTD.1]. The TOE must be able to recognize the different administrative and user roles that exist for the TOE [FMT_SMR.1]. The TOE must ensure that all functions are invoked and succeed before each function may proceed [FPT_RVM.1]. The TSF must be protected from interference that would prevent it from performing its functions [FPT_SEP.1].

O.OFLOWS The TOE must appropriately handle potential audit and System data storage overflows.

The TOE is required to protect the audit data from deletion as well as guarantee the availability of the audit data in the event of storage exhaustion, failure or attack [FAU_STG.2-INTERP-141]. The TOE must prevent the loss of audit data in the event its audit trail is full [FAU_STG.4]. The System is required to protect the System data from any modification and unauthorized deletion, as well as guarantee the availability of the data in the event of storage exhaustion, failure or attack [IDS_STG.1]. The System must prevent the loss of audit data in the event its audit trail is full [IDS_STG.2].

O.AUDITS The TOE must record audit records for data accesses and use of the System functions.

Security-relevant events must be defined and auditable for the TOE [FAU_GEN.1-INTERP-202]. The TOE must provide the capability to select which security-relevant events to audit [FAU.SEL.1]. The TOE must prevent the loss of collected data in the event its audit trail is full [FAU_STG.4]. The TOE must ensure that all functions are invoked and succeed before each function may proceed [FPT_RVM.1]. The TSF must be protected form interference that would prevent it from performing its functions [FPT_SEP.1]. Time stamps associated with an audit record must be reliable [FPT_STM.1].

O.INTEGR The TOE must ensure the integrity of all audit and System data.

The TOE is required to protect the audit data from deletion as well as guarantee the availability of the audit data in the event of storage exhaustion, failure or attack [FAU_STG.2-INTERP-141]. The System is required to protect the System data from any modification and unauthorized deletion [IDS_STG.1]. Only authorized administrators of the System may query or add audit and System data [FMT_MTD.1]. The System must protect the collected data from modification and ensure its integrity when the data is transmitted to another IT product [FPT_ITC.1, FPT_ITI.1]. The TOE must ensure that all functions to protect the data are not bypassed [FPT_RVM.1]. The TSF must be protected form interference that would prevent it from performing its functions [FPT_SEP.1].

O.EXPORT When any IDS component makes its data available to other IDS components, the TOE will ensure the confidentiality of the System data.

The TOE must make the collected data available to other IT products [FPT_ITA.1]. The TOE must protect all data from modification and ensure its integrity when the data is transmitted to another IT product [FPT_ITC.1, FPT_ITI.1].

10.4 Rationale for Assurance Requirements

EAL2 was chosen to provide a low to moderate level of assurance that is consistent with good commercial practices. As such minimal additional tasks are placed upon the vendor assuming the vendor follows reasonable software engineering practices and can provide support to the evaluation for design and testing efforts.

The chosen assurance level is appropriate with the threats defined for the environment. While the System may monitor a hostile environment, it is expected to be in a non-hostile position and embedded in or protected by other products designed to address threats that correspond with the intended environment. At EAL2, the System will have incurred a search for obvious flaws to support its introduction into the non-hostile environment. This specific ST chose EAL 2 in order to conform to the Assurance Requirements levied in the Intrusion Detection System System Protection Profile Version 1.4, February 4, 2002. In the TOE, there are no non-administrative user interfaces i.e. the user is not provided with a direct interface to the product or an account to use to log into the product. Hence, the requirement for user guidance (AGD_USR.1) does not apply since the TSF does not provide any interfaces for direct use by non-administrative users. Thus the AGD_USR.1 requirement is vacuously satisfied. (PD-0106: Situations where AGD_USR may be vacuously satisfied)

10.5 Rationale for Explicitly Stated Requirements

A family of IDS requirements was created to specifically address the data collected and analyzed by an IDS. The audit family of the CC (FAU) was used as a model for creating these requirements. The purpose of this family of requirements is to address the unique nature of IDS data and provide for requirements about collecting, reviewing and managing the data. These requirements have no dependencies since the stated requirements embody all the necessary security functions.

10.6 Rationale for Strength of Function

The TOE minimum strength of function is SOF-basic. The evaluated TOE is intended to operate in commercial and DoD low robustness environments processing unclassified information. This security function is in turn consistent with the security objectives described in section as required by the Intrusion Detection System System Protection Profile Version 1.4, February 4, 2002.

10.7 Rationale for Satisfying All Dependencies

The Intrusion Detection System System Protection Profile does satisfy all the requirement dependencies of the Common Criteria. Table below Requirement Dependencies lists each requirement from the Intrusion Detection System System Protection Profile with a dependency and indicates whether the dependent requirement was included. As the table indicates, all dependencies have been met.

Table 8 - Requirement Dependencies

Functional Component	Dependency	Included
FAU_GEN.1- INTERP-202	FPT_STM.1	Yes
FAU_SAR.1	FAU_GEN.1- INTERP-202	Yes
FAU_SAR.2	FAU_SAR.1	Yes
FAU_SAR.3	FAU_SAR.1	Yes
FAU_SEL.1	FAU_GEN.1- INTERP-202 and FMT_MTD.1	Yes
FAU_STG.2	FAU_GEN.1	Yes
FAU_STG.4	FAU_STG.2- INTERP-141	Yes
FIA_UAU.1	FIA_UID.1	Yes

Functional Component	Dependency	Included
FIA_AFL.1	FIA_UAU.1[6]	Yes
FMT_MOF.1	FMT_SMF.1-INTERP-065 and FMT_SMR.1	Yes
FMT_MTD.1	FMT_SMF.1-INTERP-065 and FMT_SMR.1	Yes
FMT_SMR.1	FIA_UID.1	Yes

11 GLOSSARY OF TERMS

CC Common Criteria

CD Compact Disc

CGI Common Gateway Interface

CLI Command Line Interface

CPU Central Processing Unit

DNS Domain Name Service

DoD Department of Defense

FTP File Transfer Protocol

HDD Hard Disk Drive

ICMP Internet Control Message Protocol

IDS Intrusion Detection System

IMAP Internet Message Access Protocol

IP Internet Protocol

IRC Internet Relay Chat

ISO International Standards Organization

IT Information Technology

MAC Message Authentication Code

NSA National Security Agency

[6] FIA_UAU.1 (timing of authentication) is dependent on FIA_UID.1 (timing of identification). Both requirements are met.

NTP	Network Time Protocol
OS	Operating System
POP	Post Office Protocol
PP	Protection Profile
RAM	Random Access Memory
RAID	Redundant Array of Independent (or Inexpensive) Disks
RPC	Remote Procedure Call
RRT	Rapid Response Team
SAI	Sentivist Administration Interface
SAR	Security Assurance Requirement
SCSI	Small Computer System Interface
SFR	Security Functional Requirement
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SRP	Secure Remote Password
ST	Security Target
TCP	Transmission Control Protocol
TFTP	Trivial File Transfer Protocol
TOE	Target of Evaluation
TSF	Target of Evaluation (TOE) Security Function
TSP	Target of Evaluation (TOE) Security Policy
UDP	User Datagram Protocol
WWW	World Wide Web

12 Appendix A: List of RFCs[7] used for Protocol Analysis and Anomaly detection in the Sentivist

Package	RFCs (or other as noted)
DHCP	951, 1533, 2131, 2132, 3396
DNS	1034, 1035, 1183, 1536, 1886, 1995, 1996, 2065, 2535, 2671, 2845, 2915, 2916, 2929, 2930, 2931, 3007, 3008, 3071, 3090, 3110, 3121, 3130, 3225, 3226, 3425
finger	1288
FTP	959,1123,1639,2428,2640
ICMP	792, 950, 1256
IMAP	1731, 2060, 3501
IP	791, 1112
IRC	1459, 2810, 2811, 2812, 2813
LPD	1179
MSRPC	CAE C706
Policy	IANA Internet Protocol V4 Address Space (http://www.iana.org/assignments/ipv4-address-space)
POP	1082, 1225, 1734, 1939, 2449, 2595,
rcommands	930, 1282
RPC	1094, 1813, 1831, 1832
Scanner	
SIP	3261, 3262, 3265
SMB	1001, 1002
SMTP	821, 822, 1869, 2045, 2476
SNMP	1157
SSH	draft-ylonen-ssh-protocol (http://www.free.lp.se/bamse/draft-ylonen-ssh-protocol-00.txt) draft-ietf-secsh-architecture (http://www.ietf.org/internet-drafts/draft-ietf-secsh-architecture-16.txt) draft-ietf-secsh-transport (http://www.ietf.org/internet-drafts/draft-ietf-secsh-transport-18.txt)
SSL	SSL 2.0 specification (http://wp.netscape.com/eng/security/SSL_2.html), SSL 3.0 Specification (http://wp.netscape.com/eng/ssl3/), TLS Specification (http://www.ietf.org/html.charters/tls-charter.html)
TCP	793, 2481
telnet	854, 855
TFTP	1350

[7] All the RFCs mentioned in this section are available at the authoritative source page, RFC Editor, http://www.rfc-editor.org/

UDP	768
www	1945, 2616