Tripwire, Inc. Tripwire Enterprise Version 5.2 Security Target

Version 1.0

April 2, 2009

Prepared for:

Tripwire, Inc.

One Main Place
101 SW Main Street
Suite 1500
Portland, OR 97204

Prepared By:

Science Applications International Corporation

Common Criteria Testing Laboratory

7125 Columbia Gateway Drive, Suite 300
Columbia, MD 21046

1. SECURITY TARGET INTRODUCTION 4

1.1 SECURITY TARGET, TOE AND CC IDENTIFICATION 4
1.2 CONFORMANCE CLAIMS 4
1.3 CONVENTIONS AND ACRONYMS 5
1.3.1 Conventions 5
1.3.2 Acronyms 5

2. TOE DESCRIPTION 7

2.1 TOE OVERVIEW 7
2.1.1 Tripwire Enterprise Nodes 10
2.1.2 Tripwire Enterprise Agent Node 11
2.1.3 Tripwire Enterprise Agentless Node 11
2.2 TOE ARCHITECTURE 12
2.2.1 Physical Boundaries 13
2.2.2 Logical Boundaries 14
2.3 TOE DOCUMENTATION 16

3. SECURITY ENVIRONMENT 16

3.1 THREATS 16
3.2 ORGANIZATIONAL SECURITY POLICIES 17
3.3 SECURE USAGE ASSUMPTIONS 17
3.3.1 Intended Usage Assumptions 17
3.3.2 Physical Assumptions 17
3.3.3 Personnel Assumptions 17

4. SECURITY OBJECTIVES 18

4.1 SECURITY OBJECTIVES FOR THE TOE 18
4.2 SECURITY OBJECTIVES FOR THE IT ENVIRONMENT 18
4.3 SECURITY OBJECTIVES FOR THE NON-IT ENVIRONMENT 18

5. IT SECURITY REQUIREMENTS 19

5.1 TOE SECURITY FUNCTIONAL REQUIREMENTS 19
5.1.1 Change Audit Assessment (CHG(EX)) 19
5.1.2 Security audit (FAU) 20
5.1.3 User data protection (FDP) 21
5.1.4 Identification and authentication (FIA) 21
5.1.5 Security management (FMT) 21
5.1.6 Protection of the TSF (FPT) 23
5.2 IT ENVIRONMENT SECURITY FUNCTIONAL REQUIREMENTS 23
5.2.1 Protection of the TSF (FPT) 23
5.3 TOE SECURITY ASSURANCE REQUIREMENTS 23
5.3.1 Configuration management (ACM) 24
5.3.2 Delivery and operation (ADO) 25
5.3.3 Development (ADV) 25
5.3.4 Guidance documents (AGD) 26
5.3.5 Life cycle support (ALC) 27
5.3.6 Tests (ATE) 27
5.3.7 Vulnerability assessment (AVA) 28

6. TOE SUMMARY SPECIFICATION 30

6.1 TOE SECURITY FUNCTIONS 30
6.1.1 Change Audit Assessment (EX) 30
6.1.2 Security audit 34
6.1.3 User data protection 34

6.1.4 Identification and authentication *36*
6.1.5 Security management *36*
6.1.6 Protection of the TSF *40*
6.2 TOE SECURITY ASSURANCE MEASURES 41
6.2.1 Configuration management *41*
6.2.2 Delivery and operation *42*
6.2.3 Development *42*
6.2.4 Guidance documents *42*
6.2.5 Life cycle support *43*
6.2.6 Tests *43*
6.2.7 Vulnerability assessment *43*

7. PROTECTION PROFILE CLAIMS 45

8. RATIONALE 46

8.1 SECURITY OBJECTIVES RATIONALE 46
8.1.1 Complete Coverage – Threats *46*
8.1.2 Complete Coverage – Environmental Assumptions *47*
8.2 SECURITY REQUIREMENTS RATIONALE 50
8.2.1 Security Functional Requirements Rationale *50*
8.3 SECURITY ASSURANCE REQUIREMENTS RATIONALE 54
8.4 STRENGTH OF FUNCTIONS RATIONALE 54
8.5 REQUIREMENT DEPENDENCY RATIONALE 54
8.6 EXPLICITLY STATED REQUIREMENTS RATIONALE 55
8.7 TOE SUMMARY SPECIFICATION RATIONALE 55
8.8 PP CLAIMS RATIONALE 56

1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization.

The TOE is Tripwire Enterprise Version 5.2, provided by Tripwire, Inc. The TOE is a change audit assessment product that can monitor the integrity of critical data on a wide variety of servers and network devices (e.g., routers, switches, firewalls, and load balancers) called nodes. It does this by monitoring system status, configuration settings, file content, and file metadata for modification.

The Security Target contains the following sections:

1. Section 1 – Security Target Introduction
   This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization.
2. Section 2 – Target of Evaluation (TOE) Description
   This section gives an overview of the TOE, describes the TOE in terms of its physical and logical boundaries, and states the scope of the TOE.
3. Section 3 – TOE Security Environment
   This section details the expectations of the environment, the threats that are countered by TOE and IT environment, and the organizational policy that TOE must fulfill.
4. Section 4 – TOE Security Objectives
   This section details the security objectives of the TOE and IT environment.
5. Section 5 – IT Security Requirements
   The section presents the security functional requirements (SFR) for TOE and IT Environment that supports the TOE, and details the assurance requirements for EAL3.
6. Section 6 – TOE Summary Specification
   The section describes the security functions represented in the TOE that satisfy the security requirements.
7. Section 7 – Protection Profile Claims
   This section presents any protection profile claims.
8. Section 8 – Rationale
   This section closes the ST with the justifications of the security objectives, requirements and TOE summary specifications as to their consistency, completeness, and suitability.

1.1 Security Target, TOE and CC Identification

ST Title – Tripwire, Inc. Tripwire Enterprise Version 5.2 Security Target

ST Version – Version 1.0

ST Date – April 2, 2009

TOE Identification – Tripwire Enterprise, Version 5.2

CC Identification – Common Criteria for Information Technology Security Evaluation, Version 2.3, August 2005.

1.2 Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Requirements, Version 2.3August 2005.
  - Part 2 Extended

- Common Criteria for Information Technology Security Evaluation Part 3: Security Assurance Requirements, Version 2.3, August 2005.
    - Part 3 Conformant
    - Assurance Level: EAL 3 augmented with ALC_FLR.2

1.3 Conventions and Acronyms

This section specifies the formatting conventions used in the Security Target and provides a glossary of acronyms.

1.3.1 Conventions

The following conventions have been applied in this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
    - Iteration: allows a component to be used more than once with varying operations. In the ST, iteration is indicated by a letter placed at the end of the component. For example FDP_ACC.1a and FDP_ACC.1b indicate that the ST includes two iterations of the FDP_ACC.1 requirement, a and b.
    - Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by bold brackets (e.g., **[assignment]**).
    - Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by bold brackets (e.g., **[*selection*]**). An assignment inside a selection is indicated using bold italics surrounded by bold italics brackets surrounded by bold brackets (e.g., **[*[selection]*]**)
    - Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …").
- Explicitly stated Security Functional Requirements (i.e., those not found in Part 2 of the CC) are identified with "**(EX)**".
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

1.3.2 Acronyms

1.3.2.1 Common Criteria Specific

CC	Common Criteria
EAL	Evaluation Assurance Level
IT	Information Technology
PP	Protection Profile
SF	Security Function
SFP	Security Function Policy
SOF	Strength of Function
ST	Security Target
TOE	Target of Evaluation
TSC	TSF Scope of Control
TSF	TOE Security Functions
TSFI	TSF Interface
TSP	TOE Security Policy

1.3.2.2 TOE Specific

3DES	The NIST Data Encryption Standard block cipher used three times, with either 2 or 3 keys
ACL	Access Control List
ANSI	American National Standards Institute
CLI	Command Line Interface
DAC	Discretionary Access Control
FIPS	Federal Information Processing Standard (NIST)
GUI	Graphical User Interface
HTTPS	Hypertext Transport Protocol Secure
IETF	Internet Engineering Task Force
JDBC	Java Data Base Connectivity
JVM	Java Virtual Machine
MD5	Message Digest 5, a hashing algorithm
NIST	National Institute of Standards and Technology
RAM	Random Access Memory. Non-volatile RAM keeps its data without power.
RMI	Remote Method Invocation
ROM	Read Only Memory. It keeps its data without power.
SHA-1	Secure Hash Algorithm 1 (NIST)
SNMP	Simple Network Management Protocol
SQL	Structured Query Language for data base access
SSL	Secure Sockets Layer
TE	Tripwire Enterprise
TLS	Transport Layer Security
UI	User Interface
VPN	Virtual Private Network

2. TOE Description

The Target of Evaluation (TOE) is Tripwire Enterprise, Version 5.2.

The TOE is a change audit assessment product that can monitor the integrity of critical data on a wide variety of servers and network devices (e.g., routers, switches, firewalls, and load balancers) called nodes. It does this by gathering system status, configuration settings, file content, and file metadata on the nodes and checking gathered node data against previously stored node data to detect modifications.

The TOE consists of a server application component (Tripwire Enterprise Server), a client application component (Tripwire Enterprise Agent), and a client administrative console application component (Tripwire CLI). The product is also bundled with a database application (Firebird Database) to support the product's storage needs. The Firebird Database is considered part of the IT environment. While the product supports using the Firebird Database and the Tripwire Enterprise Server (TE Server) on different machines, they must run on the same machine in an evaluated configuration. The other TOE components can run on different machines in various combinations. The Tripwire Enterprise Server is the only product installed and active on the machine in which it is running.

In addition to these major components, the TOE includes web server functionality that supports HTTPS (HTTP over SSL[1]) connections from a web browser to the TOE's GUI and CLI. The Tripwire Enterprise Server, being a Java application, utilizes the SSL mechanism provided by the JVM in the IT environment as part of the HTTPS communication with the GUI and the CLI. Communications between distributed parts of the TOE are protected from most compromises by the SSL that is provided by the IT environment. In the evaluated configuration, physical protection is assumed for all parts of the TOE, and for remotely monitored nodes.

The TOE also uses RMI (Remote Method Invocation) over mutually authenticated SSL network connections to protect intra-TOE communication between Tripwire Enterprise Server and the Tripwire Enterprise Agents over an untrusted network. RMI is a Java protocol that enables distributed Java-to-Java applications to invoke Java methods on other Java virtual machines, possibly on different hosts. Because RMI is only used after establishing an authenticated SSL connection between the Tripwire Enterprise Server and the Tripwire Enterprise Agents components, the RMI protocol is not available to untrusted callers.

While the TOE can be configured in many ways, the evaluated configuration of the TOE is a specific configuration. The evaluated configuration is shown in Figure 1. It consists of a single Tripwire Enterprise Server and Firebird Database running on the same machine. It includes a Tripwire CLI application running on either the same machine as the Tripwire Enterprise Server or on another machine. Finally, the evaluated configuration includes one or more Tripwire Enterprise Agents running on separate platforms (typically network servers) in the IT environment.

The remainder of this section summarizes the Tripwire Enterprise architecture.

2.1 TOE Overview

The TOE is a change audit assessment product that can monitor the integrity of critical data on a wide variety of servers and network devices (e.g., routers, switches, firewalls, and load balancers) called nodes. It does this by monitoring system status, configuration settings, file content, and file metadata for modification.

There are two classes of nodes that the TOE can monitor, those with built-in external administration interfaces and those without. Examples of the kind of node with built-in administration interfaces are firewalls, routers, switches, load balancers, etc.. Some of these external interfaces use web servers and allow administration via a remote web browser, and others provide command line interfaces or other custom protocols. These nodes are referred to as agentless nodes. Examples of nodes without built-in administration interfaces are Microsoft Windows systems and UNIX systems (Solaris, AIX, HP-UX, etc.) These nodes are referred to as agent nodes, and host an installation of Tripwire Enterprise Agent.

[1] Any reference in the Security Target to SSL being used by the TOE or provided by a JVM is a generalization and actually refers to the TLSv1 protocol being invoked by the TOE and provided by the JVM.

The Tripwire Enterprise Agent provides an interface for Tripwire Enterprise Server where none otherwise exists or to provide a more fully featured interface than an existing one. Tripwire Enterprise Agents are installed on nodes that run server-type operating system.

The TOE may also be used to monitor the configuration of its nodes, thereby identifying changes made by users or other applications, such as software-provisioning and patch-management tools that run independently of Tripwire Enterprise.

A node is represented in the TOE by its network address (hostname or IP address).

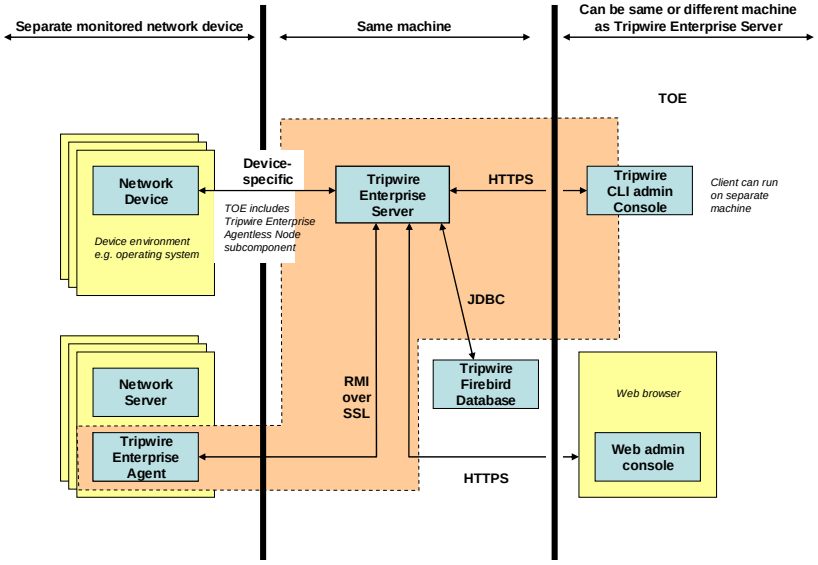


Figure 1: TOE boundary

The Tripwire Enterprise Server component delegates work to Tripwire Enterprise Agents, interacts with agentless nodes, analyzes information, and provides a web-based user interface (or a network interface that provides limited functionality using the Tripwire CLI application component as an alternative to a web browser) for managing the TOE installation. The Tripwire Enterprise Server component is composed of five main subcomponents:

- User Interface (UI)
- Downloadable Agent Code
- Database Mapping Layer
- Remote Method Invocation (RMI) Layer
- Agentless Node device-specific interface

The User Interface (UI) subcomponent provides two interfaces for other programs that provide access to the administrators of the system; a graphical user interface (GUI) called the Tripwire Enterprise Web Admin Console and a command line interface (CLI) called the Tripwire CLI. They will be referred to in this document as the GUI and CLI, respectively.

The UI's GUI is a web server running in the TOE for use by an external web browser. The web browser is not part of the TOE. The connection between the web browser and the GUI uses HTTPS to protect the integrity of the connection. The GUI provides an administrator the ability to perform such functions as add users, configure and schedule integrity checks[2], manage nodes, and view reports. User identification and authentication is handled through the GUI.

[2] The more general term 'integrity check' is used in this Security Target, but is intended to have the same meaning as the term 'version check' which is used in Tripwire guidance documentation.

The UI's CLI provides an interface for scripts to perform a limited number of operations on the TOE. Its functionality is a subset of the GUI's and is insufficient to fully administer the TOE. For example, there are no CLI commands for adding or deleting users or changing passwords. The CLI provides administrator access to the Tripwire Enterprise Server. Like the GUI interface, the CLI connects to the Tripwire Enterprise Server using HTTPS[3].

Commands available through the CLI include the following:

baseline	Baselines sets of monitored elements (node groups, a single node, or elements within a node).
check	Checks monitored elements for change. (Also known as an integrity check)
help	Provides assistance in using the CLI.
import	Imports an XML node file or an XML rule file into the Tripwire Enterprise configuration.
licurl	Generates a Launch-in-Context URL.
promote	Promotes the latest version of an element to the baseline.
report	Generates Tripwire Enterprise reports, by specific nodes or rules.
set	Sets default arguments for common options for other CLI commands.
variable	Defines or updates a Tripwire Enterprise variable.
version	Reports the current version of the CLI.

Tripwire Enterprise Server user identification and authentication credentials must be included as arguments to the **baseline**, **check**, **import**, **licurl**, **promote**, **report** and **variable** commands every time they are invoked.[4] The other commands are processed locally by the CLI and are not passed to the Tripwire Enterprise Server. The **set** command can be used to specify the default userid and password during a CLI session, after which arguments with the default userid and password will be automatically added to each of the commands sent to the Tripwire Enterprise Server. Use of this feature is discouraged by Tripwire CC guidance documents (i.e., the Tripwire Enterprise 5.2 Release Notes Addendum). Including user identification and authentication credentials as arguments to each command allows the Tripwire Enterprise Server to process commands from multiple users without having to maintain separate user sessions. Every command sent to Tripwire Enterprise Server stands alone and executes without a session context.

The Downloadable Agent Code is the portion of the Tripwire Enterprise Server that provides instructions to the Tripwire Enterprise Agent component about what types of operations to perform on a particular agent's host machine. This code is a Java object that is executed by the Tripwire Enterprise Agent by means of Remote Method Invocation (RMI). This code can contain instructions to perform any of the following activities:

- Create a baseline (a known good state)
- Run an integrity check
- Execute an action

The Database Mapping Layer provides a JDBC interface to a SQL database. The default database provided with Tripwire Enterprise, and the only one included in the evaluated configuration, is the Firebird Database. The Firebird Database must be installed and run on the same machine as the Tripwire Enterprise Server. During the Tripwire Enterprise Server installation a single account is created in the Firebird Database. This account is used by the Tripwire Enterprise Server when storing and accessing configuration information about monitored objects[5], log messages and report data in the database. The Firebird Database is relied upon to store, retrieve and protect data which it handles such that only the Tripwire Enterprise Server can access its own data.

[3] The Tripwire Enterprise Server uses the SSL provided by the IT Environment for HTTPS communications.
[4] If authentication credentials (i.e., userids and passwords) are saved in scripts on the machine being used an administrative workstation, protection of these scripts is outside the scope of the TOE.
[5] This includes user names and passwords used to establish a connection to agentless nodes. These passwords are stored within the database in an hashed form.

The Remote Method Invocation (RMI) Layer allows Tripwire Enterprise Server to execute Java byte code on Tripwire Enterprise Agents. RMI runs over a mutually authenticated SSL connection[6]. Since the protocol transporting the RMI protocol is authenticated, RMI need not perform additional security checks. However, as a precaution, RMI messages are executed using downloaded bytecode encapsulated in JAR files which are signed by another certificate. These JAR files are signed by a certificate, in which the private signing half is only held in-house at Tripwire headquarters. Any communications based on unsigned or improperly signed JARs are rejected at the start of the RMI communication.

The Agentless Node device-specific interface provides a custom interface to the management interfaces of a specific list of supported devices. The device-specific interfaces utilize protocols such as SSH, telnet, and ftp . Agentless Nodes are described in Section 2.1.3.

The Tripwire Enterprise Server can be installed on the following platforms:

- Windows 2000, XP Professional, 2003
- Solaris 7, 8, or 9
- Red Hat Enterprise Linux 3 & 4

Tripwire Enterprise installs the following JVM for use by the TOE on these platforms.

- Sun Microsystems JVM 1.4.2_08

The table below shows the version of the Firebird Database that is installed with the Tripwire Enterprise Server for each supported platform.

Platform	**Firebird Database Version Number**
Windows platforms:	Version 1.0.3
Solaris 7, 8, or 9	Version 1.0.0
Red Hat Enterprise Linux 3 & 4	Version 1.0.3

2.1.1 Tripwire Enterprise Nodes

A node is a server or network device to be monitored. There are two types of nodes – Agent nodes and Agentless nodes. Agent nodes have code running on the target system, a Tripwire Enterprise Agent, and agentless nodes communicate via standard network protocols (e.g., SSH, Telnet) between the Tripwire Enterprise Server and the target network device. Both types of nodes are presented in this discussion.

The Tripwire Enterprise agents can be installed on the following platforms:

- Windows 2000, XP Professional, 2003
- Solaris 8, 9, 10
- Red Hat Enterprise Linux 3 & 4, SUSE Enterprise Server 9
- HP-UX 11.0, 11i v1, 11i v2
- AIX 5.1, 5.2, 5.3

All current patches and security fixes must be installed upon these operating systems before installing the Tripwire Enterprise agents.

The Tripwire Enterprise agentless configuration can target the following network devices:

- Alcatel OmniSwitch
- Cisco Catalyst Routers & Switches
- Cisco IOS Routers & Switches
- Cisco PIX Firewall
- Cisco VPN Concentrator

[6] SSL is provided by the JVM in the IT environment

- Extreme
- F5 BigIP
- Foundry
- HP ProCurve M & XL
- Juniper M/T Series
- Marconi ASX
- NetScreen
- Nokia IPSO Firewall
- Nortel Passport & Alteon
- POSIX compliant UNIX systems

All current patches and security fixes must be installed upon these network devices before allowing a Tripwire Enterprise Server to target the network device.

2.1.2 Tripwire Enterprise Agent Node

The Tripwire Enterprise Agent portion of the TOE is installed on nodes that do not provide external management interfaces of their own. It is the agent of the Tripwire Enterprise Server on that node. It is always running and ready to receive instructions from the Tripwire Enterprise Server.

The Tripwire Enterprise Agent is executes baselines and integrity checks on its node and communicates the results from those operations to the Tripwire Enterprise Server for reporting and for integration with system-wide results. The Tripwire Enterprise Agent is composed of three parts.

The Bootstrap layer portion of the Agent is a basic Java Virtual Machine (JVM). When it starts, it registers itself with a single Tripwire Enterprise Server.

The Downloaded Code portion of the Tripwire Enterprise Agent is a set of directives it receives from the Tripwire Enterprise Server after registration. The Tripwire Enterprise Server uses the directives to customize the Agent to provide the services required by its configuration. The Tripwire Enterprise Agent is always running and ready to receive instructions from the Tripwire Enterprise Server.

Tripwire Enterprise Server receives attribute values and baseline from the Tripwire Enterprise Agent (the Tripwire Enterprise Server is responsible for storing these values). The most recent set of harvested attributes is cached on the Tripwire Enterprise Agent until it can be sent to the Tripwire Enterprise Server.

The Tripwire Enterprise Agent uses a Remote Method Invocation (RMI) layer over SSL to communicate securely with the Tripwire Enterprise Server.

2.1.3 Tripwire Enterprise Agentless Node

An Agentless node is a node monitored by Tripwire Enterprise Server without the need of a Tripwire Enterprise Agent. Agentless nodes are most often network devices that provide their own built-in external management interfaces and do not have operating systems that could support a Tripwire Enerprise Agent. For these devices, Tripwire Enterprise Server uses the built-in interfaces to monitor the configurations..

The configuration data for an agentless node is specific to each agentless node type and is not provided by the TOE. The Tripwire Enterprise Server obtains information about the current configuration from the agentless node as needed and compares that information to saved baseline configuration information.

Agentless nodes are monitored from the Tripwire Enterprise Server without installing software on the node itself. For each Agentless node, the Tripwire Enterprise Server uses the following access information:

- Target IP address (or hostname)
- Communications protocols
- Authentication credentials

Tripwire Enterprise Server uses the above information to establish a connection and authenticate itself to the agentless node, as if it were an administrator. It uses its device-specific knowledge of the format and structure of the management interface to collect configuration information. When verifying the integrity of the configuration on an

agentless node, Tripwire Enterprise Server compares newly collected information with baseline configuration information.

2.2 TOE Architecture

The components that make up the TOE are:

- Tripwire Enterprise Server – Analyzes collected information from Tripwire Enterprise Agents and Agentless nodes. Tripwire Enterprise Server includes a User Interface (UI) subcomponent that provides interfaces to both web-based (GUI) and command line (CLI) administrative interface applications.
- Tripwire Enterprise Agent – Collects information from monitored servers for the Tripwire Enterprise Server.
- Tripwire CLI – The command-line administrative interface to the UI.

The TOE relies upon the following software in the local[7] IT environment of the Tripwire Enterprise Server or the Tripwire Enterprise Agent.

- Firebird Database – Stores data for Tripwire Enterprise Server.
- Java Virtual Machine – provides a runtime environment for the TOE.

The TOE assumes the following network IT entities are in the operating network environment.

- SMTP Server – An email server is used to facilitate delivery of integrity check results to administrators when the TOE is so configured.
- SNMP recipient -- A network management device is used to facilitate delivery of integrity check results to administrators when the TOE is so configured.
- Syslog Server – A destination for the collection of log messages sent by the TOE.
- Workstation providing a web browser for access to the GUI
- Servers requiring Tripwire Enterprise Agents
- Agentless Nodes

Notification mechanisms such as email, SNMP, and syslog server are outside of the TOE boundary. The TOE implements only client-side protocols for these protocols. The TOE utilizes external (non-TOE) mechanisms for delivery of notifications, thus the TOE cannot guarantee delivery of notifications. Web browsers used with the web-based GUI are not part of the TOE.

The Tripwire Enterprise Server uses various network protocols to communicate with other parts of the TOE and with the IT environment. Depending upon the communication pathway the Tripwire Enterprise Server acts either as a server or as a client on each pathway. The following summarize the network communication pathways that exist.

- Tripwire Enterprise Server – Tripwire Enterprise Agent communication.

  The Tripwire Enterprise Server and Tripwire Enterprise Agent are peers, with either able to initiate communication to accomplish the task being performed. Both the Tripwire Enterprise Server and Tripwire Enterprise Agent components of the TOE configure and use the SSL provided to them by the JVM upon which they are running. A mutually authenticated SSL connection is established that allows these components of the TOE to use the RMI protocol to exchange services.

- Tripwire Enterprise Server – Firebird Database

  The Tripwire Enterprise Server is the only component of the TOE that communicates with the Firebird Database. It uses the JDBC protocol to connect to the database on the same host which is running the Tripwire Enterprise Server component of the TOE.

[7] Local IT environment refers to software running on the same host as either the server or agent components of the TOE.

- Tripwire Enterprise Server – Agentless nodes

  The Tripwire Enterprise Server initiates connections to agentless nodes to obtain information made available by protocols supported by the node (e.g., FTP, Telnet, SSH). For protocols requiring user authentication, the Tripwire Enterprise Server provides login data for the specific node being accessed, then gathers information from the node as determined by rules established for that network device (e.g., a specific Cisco PIX Firewall or Netscreen device).

- Tripwire Enterprise Server – CLI & GUI

  The Tripwire Enterprise Server includes web server functionality that supports HTTP over the SSL provided by the JVM in the IT environment. Thus, the communication between the Tripwire Enterprise Server and the Tripwire CLI uses the HTTPS protocol. Similarly, communication between the Tripwire Enterprise Server and the GUI is also over the HTTPS protocol.

- Tripwire Enterprise Server – SMTP/SNMP/Syslog server

  The Tripwire Enterprise Server is a client to SMTP/SNMP/Syslog servers. The Tripwire Enterprise Server uses these servers as configurable delivery mechanisms for TOE generated messages.

Tripwire CLI does not offer any inbound network communication pathways..

The only network protocol accepted by Tripwire Enterprise Agents is RMI, which is received over this authenticated SSL connection established with the Tripwire Enterprise Server. Both Tripwire Enterprise Server and Tripwire Enterprise Agents configure the JVM on which they are running to reject connections over SSL that fail authentication.

2.2.1 Physical Boundaries

The Evaluated Configuration includes components running in the TOE boundary and components running outside the TOE boundary. Inside the TOE boundary are Tripwire Enterprise Server, and the Tripwire CLI running on a single computer, and one or more Tripwire Enterprise Agents running on remote servers.

The Tripwire Enterprise Server component of the TOE can operate on several supported operating systems:

- Windows 2000, XP Professional, 2003
- Solaris 7, 8, or 9
- Red Hat Enterprise Linux 3 & 4

The host operating system for Tripwire Enterprise Server has no impact on the supported list of Tripwire Enterprise Agent or Agentless nodes that can be monitored.

The TOE configuration also includes a web browser and a network connecting all of the other components into a single LAN.

The TOE relies upon the following software in the IT environment.

- Firebird Database – Stores data for Tripwire Enterprise Server.
- Java Virtual Machine – provides a runtime environment for the TOE.
- Host Operating System – provides process-related (e.g., time) and network-related (e.g., name resolution) services for the JVM.

The TOE assumes the following network IT entities are in the operating environment.

- SMTP Server – An email server is used to facilitate delivery of integrity check results to administrators when the TOE is so configured.
- SNMP recipient -- A network management device is used to facilitate delivery of integrity check results to administrators when the TOE is so configured.
- Syslog Server – A destination for the collection of log messages sent by the TOE.

- Servers requiring Tripwire Enterprise Agents
- Agentless Nodes

The Evaluated Configuration of the TOE does not include Web Services Integration, the Remedy AR System tickets Plug-in, the HP Openview Plug-in, or the AAA Monitoring Tool. The Evaluated Configuration of the TOE also does not include authentication using external servers.

2.2.2 Logical Boundaries

This section identifies the security functions that the TSF provides**.**

- Change Audit Assessment
- Security Audit
- User Data Protection
- Identification and Authentication
- Security Management
- Protection of the TSF

2.2.2.1 Change Audit Assessment (EX)

The Tripwire Enterprise Agent component of the TOE can collect object attribute information for files, directories, registry keys and registry key values. By comparing collected information against saved values, the agent monitors these resources to detect changes. Once detected, the agent reports the detected change to the Tripwire Enterprise Server to allow administrator specified actions to occur.

For Agentless nodes, the Tripwire Enterprise Server component collects attribute information, compares the information to baselines and initiates administrator specified actions. The Tripwire Enterprise Server can monitor files on Agentless nodes, command output from agentless nodes, and network availability using interfaces that each node provides.

The Tripwire Enterprise Server component can perform actions in response to object attribute comparisons, specifically: display integrity check results to the console, send integrity check results to administrators using email, send integrity check results to administrators using SNMP, send a log message to a Syslog server, execute a command on the Tripwire Enterprise Server host operating system, execute a command on the Tripwire Enterprise Agent host operating system, and promote new element versions to be a baseline. For some Agentless node types, Tripwire Enterprise Server can also restore a changed element to its baseline state on the Agentless node.

The syslog server, SMTP server, and recipient of SNMP messages are all external IT entities residing in the environment. It is the responsibility of these IT entities to complete the delivery of such communications. The TOE provides the functionality to send communications to these IT entities in the environment. The TOE does not rely upon these external IT entities to provide security for the TOE.

2.2.2.2 Security Audit

The TOE provides its own audit mechanism, with its own audit trail, that can generate audit records containing integrity check results and TOE management actions. The TOE refers to audit records as ‘log messages” and to the audit trail as the “message log”. These terms are interchangeable. The TOE stores the “message log” in the Firebird Database.

The TOE provides administrators the ability to manage the Tripwire Enterprise Server audit trail using administrator console interfaces. Administrators can read, search and sort log messages in the audit trail based on date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event.

The security audit function relies on the TOE’s environment to supply a reliable date and time stamp which the TOE includes in each audit record. The security audit function also relies upon the TOE’s environment to store and protect audit data.

2.2.2.3 User Data Protection

The TOE implements three security objects: user sessions, nodes, and node groups. The user session is the only object that is also a subject. Nodes and node groups are definitions of network entities upon which some integrity check operation is to be performed. Examples of the information the Tripwire Enterprise Server retains about a node or node group are a name, the type of node(s), a description, the number of elements being checked, and last check date/time.

Access to subjects and objects is controlled by the Discretionary Access Control (DAC) policy, as defined in FDP_ACF.1 in Section 5.1.3.2, for all available operations on nodes and node groups (and their contents). Node objects have ACLs that can specify user, user group, and role access permissions. These attributes are compared against user identities and roles of subjects in order to determine whether requested operations should be allowed. If the access checks fail, access will be refused. Only the administrator role can access the user session subject.

2.2.2.4 Identification and Authentication

The TOE defines user identities, authentication data, user groups, and role information. The TOE offers no TSF-mediated functions until the user is authenticated. The TOE offers no TSF-mediated functions until the user is identified.

2.2.2.5 Security Management

Tripwire Enterprise Server offers a graphical user interface (GUI), and a command line interface (CLI). The TOE restricts the ability to execute commands by restricting access to these user interfaces, by enforcing user permissions, and by assigning roles to users.

A user permission is a system authorization which enables a user with that permission to view, add, change, or delete data in Tripwire Enterprise. The following list defines common types of user permissions.

- **Load permissions** provide read-only access to a class of Tripwire Enterprise objects and groups. For instance, the load rules permission grants access to the Rule Manager. In the Rule Manager, users can review all rules and rule groups.
- **Create permissions** authorize users to create a class of Tripwire Enterprise objects and groups. For example, the create nodes permission authorizes users to create nodes and node groups.
- **Delete permissions** authorize users to permanently remove objects or groups from the system. For instance, with the delete nodes permission, users can delete both nodes and node groups.
- **Update permissions** enable users to modify the properties of a class of Tripwire Enterprise objects and groups. For example, users can change the properties of nodes and node groups with the update nodes permission.
- **Manage permissions** authorize users to modify Tripwire Enterprise settings or parameters.

Tripwire Enterprise includes five default user roles all of which are considered to be trusted accounts. A user role is a collection of user permissions that may be assigned to a user. The default user roles are: Administrator, Power User, Regular User, Monitor User, and User Administrator. Four of these default user roles (Administrator, Power User, Regular User, and Monitor User) are organized hierarchically from most capabilities to least capabilities being Administrator, Power User, Regular User and Monitor User. The User Administrator role is orthogonal to the other roles and has permissions to manipulate user accounts.

2.2.2.6 Protection of the TSF

The TOE relies on the underlying operating system to provide the abstraction of a process. The TOE uses one process for the execution of the Tripwire Enterprise Server. The Firebird database (in the IT environment) operates in its own process. Additional subordinate processes may be created by the Tripwire Enterprise Server to perform independent tasks, however, this process structure is not related to Tripwire Enterprise Server enforcement of internal user roles. It is expected that the IT environment provides protections such that the communications between the Tripwire Enterprise Server and the Firebird database cannot be spoofed. It is also expected that the

Firebird database be protected such that only the TOE can access the database. This is accomplished by requiring the database to be on the same host as the Tripwire Enterprise Server. The Firebird Database is relied upon to store, retrieve and protect data which it handles such that only the Tripwire Enterprise Server can access TOE data in the database.

The Tripwire Enterprise Server is a Java program that runs on its own JVM. The JVM is provided as part of the TOE installation process, as a distinct product, but is not part of the TOE. The TOE itself distinguishes actions of TOE users within the TOE by associating users with threads running within the JVM. The TOE does not provide a general programming interface to TOE users. The user community of the TOE has no relationship to the users of the underlying operating system. The JVM also provides the actual implementation of SSL (i.e., TLSv1). References to the TOE using SSL mean that the TOE is using the JVM SSL implementation.

Agentless nodes provide an interface conformant with their security model for external access to the data objects that the TOE monitors. The TOE complies with that security model in accessing the objects (e.g., by providing login credentials required by the agentless nodes using supported network protocols such as SSH or telnet). For agentless nodes that do not support SSH based protocols for login, it is expected that those responsible for managing the agentless nodes have taken steps to secure the communication pathways between the TOE and the agentless nodes per their security environment. The TOE does not rely upon the security of these communication pathways to agentless nodes for TOE's self protection.

The Tripwire Enterprise Server and Tripwire Enterprise Agent components of the TOE configure and use the SSL provided to them by the JVM upon which they are running. Both the server and agent configure their JVM to require a mutually authenticated SSL connection be established for all communications between the server and agent. This allows these components of the TOE to use the RMI protocol to exchange services.

The TOE also uses the JVM provided SSL to protect the confidentiality and integrity of its communication with the users of the GUI and CLI. This protects the data, including TOE user names and passwords, from manipulation and observation.

2.3 TOE Documentation

Tripwire offers a series of documents that describe the installation process for Tripwire Enterprise as well as guidance for subsequent use and administration of the applicable security features. Refer to Section 6 for information about these and other documentation associated with Tripwire Enterprise.

3. Security Environment

This section summarizes the threats addressed by the TOE (often with help from its environment) and assumptions about the intended environment of the TOE. Note that while the identified threats are mitigated by the security functions implemented in the TOE, the overall assurance level (EAL 3 augmented with ALC_FLR.2) also serves as an indicator of whether the TOE would be suitable for a given environment.

3.1 Threats

T.AUTHENT	An authorized user may be unaware of an inadvertent change TOE data or functions they are authorized to modify.
T.COLLECT	An attacker may be able to inappropriately change attribute information for targeted objects without being detected.
T.MANAGE	An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.
T.PROTECT	An attacker may be able to gain unauthorized access to data collected from targeted objects.

3.2 Organizational Security Policies

There are no organization security policies.

3.3 Secure Usage Assumptions

3.3.1 Intended Usage Assumptions

A.ACCESS The TOE has access to all of the IT Systems (nodes) and data within those IT Systems that it is configured to monitor.

A.ASCOPE The TOE is appropriately scalable[8] to the IT Systems it is configured to monitor.

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

A.DEDICATE The host on which the TE server component of the TOE resides does not provide a general purpose computing environment to untrusted users.

3.3.2 Physical Assumptions

A.LOCATE The processing resources of the TOE and IT environment platform upon which it executes will be located within controlled access facilities, which will prevent unauthorized physical access.

A.PROTCT The TOE software critical to security policy enforcement and IT environment platform upon which it executes will be protected from unauthorized physical modification.

A.NETOK Those responsible for managing the agentless nodes have taken steps to secure the communication pathways between the TOE and the agentless nodes per their security environment.

3.3.3 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 willfully negligent or hostile, and will follow and abide by the instructions provided by the TOE documentation.

A.NOTRST Only authorized users can access the TOE.

[8] Appropriately scalable refers to the TOE being able to handle the volume of processing or traffic flow for systems which it is monitoring.

4. Security Objectives

This section summarizes the security objectives for the TOE and its environment.

4.1 Security Objectives for the TOE

O.AUDITING	The TOE shall provide the capability to create records containing integrity check results and security-relevant events associated with users.
O.AUTHENT	The TOE shall verify the claimed identity of users.
O.COLLECT	The TOE shall collect attribute information for targeted objects and maintain a baseline of attributes for each.
O.COMPARE	The TOE shall perform integrity checks on targeted objects by comparing collected attributes of each object against its stored baseline and generating a report containing integrity check results.
O.DAC	The TOE shall control access to resources based upon the identity of users, groups of users, and roles.
O.MANAGE	The TOE shall provide functions such that authorized users can manage it and provide user and administrator guidance showing how to use the interfaces to install and configure the system securely.
O.PROTECT	The TOE shall protect itself and its assets from external interference or tampering.

4.2 Security Objectives for the IT Environment

OE.TIME	The IT environment will provide a time source that provides reliable time stamps.
OE.PROTECT	The IT environment shall protect the TOE, TOE assets and intra-TOE communications from disclosure, interference, and tampering.
OE.DBCOMM	The IT environment protects the external database such that communications between the TOE and the external database is protected from disclosure and modification.
OE.PROTDB	The IT environment must ensure that the external database is protected such that only the TOE can access the external database.

4.3 Security Objectives for the Non-IT Environment

OE.INSTALL	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.
OE.PHYCAL	Those responsible for the TOE must ensure that those parts of the TOE critical to security policy are protected from any physical attack.
OE.CREDEN	Those responsible for the TOE must ensure that passwords are protected in a manner that is consistent with IT security.
OE.PERSON	Personnel working as authorized administrators shall be carefully selected and trained for proper operation of the System.
OE.INTROP	The TOE is interoperable with the IT System it monitors.
OE.COMSEC	Those responsible for managing the agentless nodes have taken steps to secure the communication pathways between the TOE and the agentless nodes per their security environment.

OE.DEDICATED Those responsible for managing the TOE will ensure that the host on which the TE server component of the TOE resides does not provide a general purpose computing environment to untrusted users.

5. IT Security Requirements

This section defines the security functional and security assurance requirements for the TOE and associated IT environment components. Note that in addition to these requirements, the TOE also satisfies a minimum strength of function 'SOF-basic. The only applicable (i.e., probabilistic or permutational) security functions are FIA_UAU.2, which is levied on the TOE.

5.1 TOE Security Functional Requirements

The following table describes the SFRs that are candidates to be satisfied by Tripwire Enterprise.

Requirement Class	Requirement Component
CHG(EX): Change audit assessment	CHG_COL.1(EX): Change Audit Collection
	CHG_ASM.1(EX): Change Audit Assessment
	CHG_REP.1(EX): Change Audit Reporting
FAU: Security audit	FAU_GEN.1(EX): Audit data generation
	FAU_SAR.1: Audit review
	FAU_SAR.3: Selectable audit review
FDP: User data protection	FDP_ACC.2: Complete access control
	FDP_ACF.1: Security attribute based access control
FIA: Identification and authentication	FIA_ATD.1: User attribute definition
	FIA_UAU.2: User authentication before any action
	FIA_UID.2: User identification before any action
FMT: Security management	FMT_MOF.1: Management of security functions behaviour
	FMT_MSA.1a: Management of object security attributes
	FMT_MSA.1b: Management of account security attributes
	FMT_MSA.3: Static attribute initialization
	FMT_MTD.1a: Management of TSF data
	FMT_MTD.1b: Management of TSF data
	FMT_MTD.1c: Management of TSF data
	FMT_MTD.1d: Management of TSF data
	FMT_SMF.1: Specification of Management Functions
	FMT_SMR.1: Security roles
FPT: Protection of the TSF	FPT_ITT.1a: Basic internal TSF data transfer protection
	FPT_RVM.1: Non-bypassability of the TSP

Table 1 TOE Security Functional Components

5.1.1 Change Audit Assessment (CHG(EX))

5.1.1.1 Change Audit Collect (CHG_COL.1 (EX))

CHG_COL.1.1 The TSF shall be able to collect the following information from the targeted IT system resource(s):

a.) Configuration information for the establishment of baselines in the TOE and for comparison with previously established baselines, and

b.) Reports of changes to configuration information.

CHG_COL.1.2 The TSF shall collect and record the following information from targeted IT system resource(s):

a.) Date and time of the collection,
b.) Type of information collected,
c.) Identity of the IT system resource from which the information was collected,
d.) Administratively configurable, rule-defined information that is specific to each targeted IT system resource type.

5.1.1.2 Change Audit Assessment (CHG_ASM.1(EX))

CHG_ASM.1.1 The TSF shall compare current and saved attributes of targeted IT system resources and detect changes to these resources as indicated by changes to the resources attributes. (EX)

5.1.1.3 Change Audit Reporting (CHG_REP.1(EX))

CHG_REP.1.1 The TSF shall take one or more of the following actions if a change is detected to the attributes of a targeted IT system resource(s):
a.) Display integrity check results to the console,
b.) Send integrity check results to administrators using email,
c.) Send integrity check results to administrators using SNMP,
d.) Send a log message to a Syslog server,
e.) Execute a command on the TE Server host operating system,
f.) Execute a command on the TE Agent host operating system,
g.) Promote new resource version to be a baseline, or
h.) Restore a changed resource to its baseline. (EX)

Application Note: *It may not be possible to restore every type of targeted IT system resource due to the nature of the resource or the capabilities of the interface offered by the network device providing the targeted IT system resource.*

5.1.2 Security audit (FAU)

5.1.2.1 Audit data generation (FAU_GEN.1(EX))

FAU_GEN.1.1(EX) The TSF shall be able to generate an audit record of the following auditable events:
a.) All auditable events for the **[*minimum*]** level of audit; and
b.) **[TOE integrity checks on monitored nodes, TOE management actions, successful and unsuccessful use of identification and authentication mechanisms].**

FAU_GEN.1.2(EX) 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, **[no other minimum audit relevant information]**.

5.1.2.2 Audit review (FAU_SAR.1)

FAU_SAR.1.1 The TSF shall provide **[administrators]** with the capability to read **[all audit information]** 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.

5.1.2.3 Selectable audit review (FAU_SAR.3)

FAU_SAR.3.1 The TSF shall provide the ability to perform **[*searches, sorting*]** of audit data based on **[the following criteria:**
a.) **date and time of the event,**
b.) **type of event,**
c.) **subject identity, and**
d.) **the outcome (success or failure) of the event]**

5.1.3 User data protection (FDP)

5.1.3.1 Complete access control (FDP_ACC.2)

FDP_ACC.2.1 The TSF shall enforce the **[Discretionary Access Control Policy]** on **[**
a.) **All server users;**
b.) **The following server objects: nodes and node groups]**
and all operations among subjects and objects covered by the SFP.

FDP_ACC.2.2 The TSF shall ensure that all operations between any subject in the TSC and any object within the TSC are covered by an access control SFP.

5.1.3.2 Security attribute based access control (FDP_ACF.1)

FDP_ACF.1.1 The TSF shall enforce the **[Discretionary Access Control Policy]** on objects based on the following: **[**
a.) **Server subject attributes: user identity, group memberships and roles**
b.) **Server object attributes: access control lists (ACLs)]**

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **[**
a.) **If the ACL grants the requesting user identity the requested access, the requested access is allowed;**
b.) **If the user identity is a member of a group and the ACL grants the group the requested access, the requested access is allowed;**
c.) **Otherwise access is denied, unless access is explicitly authorized in accordance with the rules specified in FDP_ACF.1.3]**.

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **[**
a.) **If the server subject is the default administrator account, the requested access is allowed.]**.

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the **[none]**.

5.1.4 Identification and authentication (FIA)

5.1.4.1 User attribute definition (FIA_ATD.1)

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.) **Group memberships**
d.) **Role]**.

5.1.4.2 User authentication before any action (FIA_UAU.2)

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

5.1.4.3 User identification before any action (FIA_UID.2)

FIA_UID.2.1 The TSF shall require each user to identify itself before allowing any other TSF-mediated actions on behalf of that user.

5.1.5 Security management (FMT)

5.1.5.1 Management of security functions behaviour (FMT_MOF.1)

FMT_MOF.1.1 The TSF shall restrict the ability to **[*disable and enable*]** the functions **[**
a.) **Related to the specification of integrity check rules**
b.) **Related to the specification of integrity check actions]**

to **[Administrator and Power User]**.

5.1.5.2 Management of security attributes (FMT_MSA.1a)

FMT_MSA.1a.1 The TSF shall enforce the **[Discretionary Access Control Policy]** to restrict the ability to **[*query, modify, delete*]** the security attributes **[of objects]** to **[Administrator]**.

5.1.5.3 Management of security attributes (FMT_MSA.1b)

FMT_MSA.1b.1 The TSF shall enforce the **[Discretionary Access Control Policy]** to restrict the ability to **[*query, modify, add, delete*]** the security attributes **[of user accounts]** to **[Administrator and User Administrator]**.

5.1.5.4 Static attribute initialization (FMT_MSA.3)

FMT_MSA.3.1 The TSF shall enforce the **[Discretionary Access Control Policy]** to provide **[*restrictive*]** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the **[no user role]** to specify alternative initial values to override the default values when an object or information is created.

5.1.5.5 Management of TSF data (FMT_MTD.1a)

FMT_MTD.1a.1 The TSF shall restrict the ability to **[*[include* or *exclude]*]** the **[integrity check rules]** to **[Administrator, Power User, and Regular User]**.

5.1.5.6 Management of TSF data (FMT_MTD.1b)

FMT_MTD.1b.1 The TSF shall restrict the ability to **[*[include* or *exclude]*]** the **[integrity check actions]** to **[Administrator**, **Power User, Regular User and Monitor User**].

5.1.5.7 Management of TSF data (FMT_MTD.1c)

FMT_MTD.1c.1 The TSF shall restrict the ability to **[*[promote to baseline]*]** the **[current collected object attributes]** to **[Administrator and Power User]**.

5.1.5.8 Management of TSF data (FMT_MTD.1d)

FMT_MTD.1d.1 The TSF shall restrict the ability to **[*query* or *clear*]** the **[integrity check reports stored on the server]** to **[Administrator]**.

5.1.5.9 Specification of Management Functions (FMT_SMF.1)

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

a.) **Specification of integrity check rules**
b.) **Specification of integrity check actions**
c.) **Promotion of collected object attributes to baselines**
d.) **Manage integrity check reports**
e.) **Schedule integrity checks**
f.) **Manage the security attributes of objects and user accounts]**

5.1.5.10 Security roles (FMT_SMR.1)

FMT_SMR.1.1 The TSF shall maintain the roles **[**Administrator, Power User, Regular User, Monitor User, and User Administrator **]**.

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

5.1.6 Protection of the TSF (FPT)

5.1.6.1 Basic internal TSF data transfer protection (FPT_ITT.1a)

FPT_ITT.1.1a The TSF shall **use mechanisms in the IT Environment to** protect TSF data from **[*disclosure, modification*]** when it is transmitted between separate parts of the TOE.

5.1.6.2 Non-bypassability of the TSP (FPT_RVM.1)

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.

5.2 IT Environment Security Functional Requirements

The following table describes the SFRs that are candidates to be satisfied by the IT environment.

Requirement Class	Requirement Component
FPT: Protection of the TSF	FPT_SEP.1: TSF domain separation
	FAU_STG.1: Protected Storage
	FPT_STM.1: Reliable Time Stamps
	FPT_ITT.1b: Basic internal TSF data transfer protection

Table 2: IT Environment Security Functional Components

5.2.1 Protection of the TSF (FPT)

5.2.1.1 TSF domain separation (FPT_SEP.1)

FPT_SEP.1.1 The ~~TSF~~ IT Environment shall maintain a security domain for ~~its own~~ execution of the TOE that protects ~~it~~ the TOE from interference and tampering by untrusted subjects.

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

5.2.1.2 Protected storage (FAU_STG.1)

FAU_STG.1.1 The ~~TSF~~ IT Environment shall protect the stored **TOE data** ~~audit records~~ from unauthorised **access** ~~deletion~~.

FAU_STG.1.2 The ~~TSF~~ IT Environment shall be able to [***prevent***] unauthorised modifications to **and disclosure of** the stored **TOE data** ~~audit records~~ in the **external database** ~~audit trail~~.

5.2.1.3 Reliable time stamps (FPT_STM.1)

FPT_STM.1.1 The ~~TSF~~ IT Environment shall be able to provide reliable time stamps for ~~its own~~ use by the TOE.

5.2.1.4 Basic internal TSF data transfer protection (FPT_ITT.1b)

FPT_ITT.1.1b The ~~TSF~~ IT Environment shall **provide mechanisms to allow the TSF to** protect TSF data from **[*disclosure, modification and ensure mutual authentication*]** when it is transmitted between separate parts of the TOE.

5.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are the EAL3 augmented with ALC_FLR.2 components as specified in Part 3 of the Common Criteria. No operations are applied to the assurance components.

Requirement Class	Requirement Component
ACM: Configuration management	ACM_CAP.3: Authorization controls
	ACM_SCP.1: TOE CM coverage
ADO: Delivery and operation	ADO_DEL.1: Delivery procedures
	ADO_IGS.1: Installation, generation, and start-up procedures
ADV: Development	ADV_FSP.1: Informal functional specification
	ADV_HLD.2: Security enforcing high-level design
	ADV_RCR.1: Informal correspondence demonstration
AGD: Guidance documents	AGD_ADM.1: Administrator guidance
	AGD_USR.1: User guidance
ALC: Life cycle support	ALC_DVS.1: Identification of security measures
	ALC_FLR.2: Flaw reporting procedures
ATE: Tests	ATE_COV.2: Analysis of coverage
	ATE_DPT.1: Testing: high-level design
	ATE_FUN.1: Functional testing
	ATE_IND.2: Independent testing - sample
AVA: Vulnerability assessment	AVA_MSU.1: Examination of guidance
	AVA_SOF.1: Strength of TOE security function evaluation
	AVA_VLA.1: Developer vulnerability analysis

Table 3 EAL 3 augmented with ALC_FLR.2 Assurance Components

5.3.1 Configuration management (ACM)

5.3.1.1 Authorization controls (ACM_CAP.3)

ACM_CAP.3.1d The developer shall provide a reference for the TOE.
ACM_CAP.3.2d The developer shall use a CM system.
ACM_CAP.3.3d The developer shall provide CM documentation.
ACM_CAP.3.1c The reference for the TOE shall be unique to each version of the TOE.
ACM_CAP.3.2c The TOE shall be labeled with its reference.
ACM_CAP.3.3c The CM documentation shall include a configuration list and a CM plan.
ACM_CAP.3.4c The configuration list shall uniquely identify all configuration items that comprise the TOE.
ACM_CAP.3.5c The configuration list shall describe the configuration items that comprise the TOE.
ACM_CAP.3.6c The CM documentation shall describe the method used to uniquely identify the configuration items.
ACM_CAP.3.7c The CM system shall uniquely identify all configuration items.
ACM_CAP.3.8c The CM plan shall describe how the CM system is used.
ACM_CAP.3.9c The evidence shall demonstrate that the CM system is operating in accordance with the CM plan.
ACM_CAP.3.10c The CM documentation shall provide evidence that all configuration items have been and are being effectively maintained under the CM system.
ACM_CAP.3.11c The CM system shall provide measures such that only authorized changes are made to the configuration items.
ACM_CAP.3.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.1.2 TOE CM coverage (ACM_SCP.1)

ACM_SCP.1.1d The developer shall provide a list of configuration items for the TOE.
ACM_SCP.1.1c The list of configuration items shall include the following: implementation representation and the evaluation evidence required by the assurance components in the ST.
ACM_SCP.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.2 Delivery and operation (ADO)

5.3.2.1 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.

5.3.2.2 Installation, generation, and start-up procedures (ADO_IGS.1)

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.

5.3.3 Development (ADV)

5.3.3.1 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.

5.3.3.2 Security enforcing high-level design (ADV_HLD.2)

ADV_HLD.2.1d The developer shall provide the high-level design of the TSF.
ADV_HLD.2.1c The presentation of the high-level design shall be informal.
ADV_HLD.2.2c The high-level design shall be internally consistent.
ADV_HLD.2.3c The high-level design shall describe the structure of the TSF in terms of subsystems.
ADV_HLD.2.4c The high-level design shall describe the security functionality provided by each subsystem of the TSF.
ADV_HLD.2.5c 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.2.6c The high-level design shall identify all interfaces to the subsystems of the TSF.
ADV_HLD.2.7c The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.
ADV_HLD.2.8c The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing details of effects, exceptions and error messages, as appropriate.
ADV_HLD.2.9c The high-level design shall describe the separation of the TOE into TSP-enforcing and other subsystems.

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

ADV_HLD.2.2e The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

5.3.3.3 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.

5.3.4 Guidance documents (AGD)

5.3.4.1 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 behaviour 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.

5.3.4.2 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 behaviour 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 are 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.

5.3.5 Life cycle support (ALC)

5.3.5.1 Identification of security measures (ALC_DVS.1)

ALC_DVS.1.1d The developer shall produce development security documentation.

ALC_DVS.1.1c The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

ALC_DVS.1.2c The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.

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

ALC_DVS.1.2e The evaluator shall confirm that the security measures are being applied.

5.3.5.2 Flaw reporting procedures (ALC_FLR.2)

ALC_FLR.2.1d The developer shall provide flaw remediation procedures addressed to TOE developers.

ALC_FLR.2.2d The developer shall establish a procedure for accepting and acting upon all reports of security flaws and requests for corrections to those flaws.

ALC_FLR.2.3d The developer shall provide flaw remediation guidance addressed to TOE users.

ALC_FLR.2.1c The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

ALC_FLR.2.2c The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC_FLR.2.3c The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC_FLR.2.4c The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.

ALC_FLR.2.5c The flaw remediation procedures documentation shall describe a means by which the developer receives from TOE users reports and enquiries of suspected security flaws in the TOE.

ALC_FLR.2.6c The procedures for processing reported security flaws shall ensure that any reported flaws are corrected and the correction issued to TOE users.

ALC_FLR.2.7c The procedures for processing reported security flaws shall provide safeguards that any corrections to these security flaws do not introduce any new flaws.

ALC_FLR.2.8c The flaw remediation guidance shall describe a means by which TOE users report to the developer any suspected security flaws in the TOE.

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

5.3.6 Tests (ATE)

5.3.6.1 Analysis of coverage (ATE_COV.2)

ATE_COV.2.1d The developer shall provide an analysis of the test coverage.

ATE_COV.2.1c The analysis of the test coverage shall demonstrate the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.

ATE_COV.2.2c The analysis of the test coverage shall demonstrate that the correspondence between the TSF as described in the functional specification and the tests identified in the test documentation is complete.

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

5.3.6.2 Testing: high-level design (ATE_DPT.1)

ATE_DPT.1.1d The developer shall provide the analysis of the depth of testing.

ATE_DPT.1.1c The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design.

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

5.3.6.3 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.

5.3.6.4 Independent testing - sample (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.

5.3.7 Vulnerability assessment (AVA)

5.3.7.1 Examination of guidance (AVA_MSU.1)

AVA_MSU.1.1d The developer shall provide guidance documentation.
AVA_MSU.1.1c The guidance documentation shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.
AVA_MSU.1.2c The guidance documentation shall be complete, clear, consistent and reasonable.
AVA_MSU.1.3c The guidance documentation shall list all assumptions about the intended environment.
AVA_MSU.1.4c The guidance documentation shall list all requirements for external security measures (including external procedural, physical and personnel controls).
AVA_MSU.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AVA_MSU.1.2e The evaluator shall repeat all configuration and installation procedures to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.
AVA_MSU.1.3e The evaluator shall determine that the use of the guidance documentation allows all insecure states to be detected.

5.3.7.2 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 defined in the PP/ST.

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 defined in the PP/ST.

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.

5.3.7.3 Developer vulnerability analysis (AVA_VLA.1)

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.

6. TOE Summary Specification

This chapter describes the security functions and associated assurance measures.

6.1 TOE Security Functions

6.1.1 Change Audit Assessment (EX)

The TOE manages and monitors the configuration and status of a set of external networked components called nodes. Discovered unauthorized or unexpected modifications to nodes are reported using one or more of the available notification mechanisms. Users can configure other actions to be performed automatically when changes are detected, including for some node types, restoring the baseline configuration.

Monitoring the integrity of a node consists of taking a node snapshot (a collection of the current values of a specific set of node elements) and comparing it against the node's baseline (a previously stored set of presumed good values for those node elements). If the resource's attributes differ between the snapshot and the baseline, the resource is determined to have changed and one or more of the available notification mechanisms may be invoked depending upon the administrator's configuration choices. Administrators define the frequency of monitoring specific nodes. Frequency can be any interval of N minutes, hourly, daily, weekly, monthly, or just once.[9]

For agentless nodes, the Tripwire Enterprise Server performs the necessary steps to monitor the node using the communication protocols offered by the specific node. For agent nodes, those nodes where Tripwire Enterprise Agent has been installed, the agent performs the integrity check and passes the results to the Tripwire Enterprise Server for auditing, notification and reporting. Node elements are specific to the type of node and how the monitoring is configured but can include the content and attributes of files, registry values, software or firmware version numbers, and other node configuration parameters.

Node snapshots are saved as a record of a node's configuration at a specific point in time. .Each node may have an unlimited number of snapshots, but only one is flagged as the current baseline. Integrity checks are performed as follows:

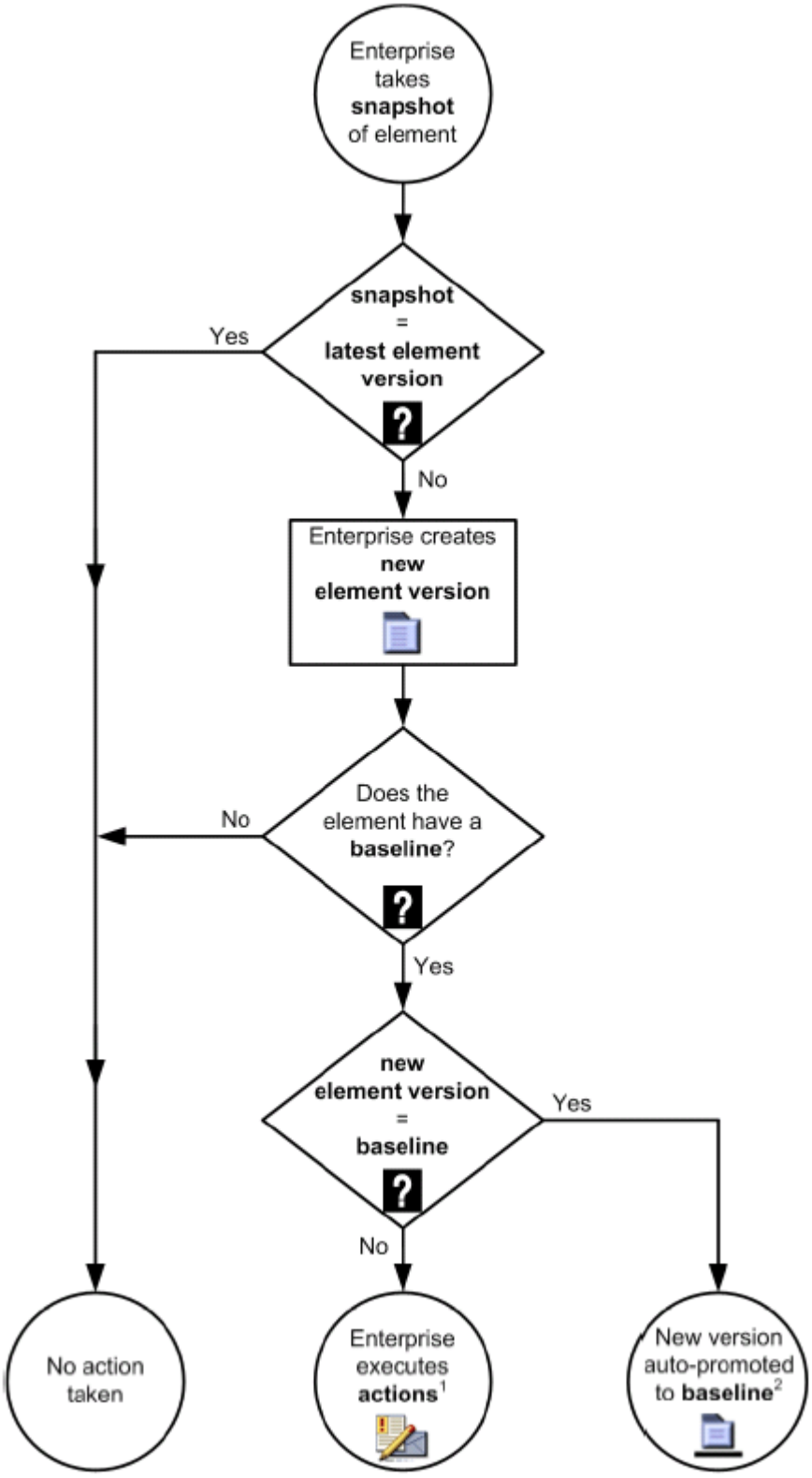
- The TOE creates a temporary record of a node's snapshot.
- To determine if the node's configuration has changed, the TOE compares the snapshot with the latest saved snapshot.
    - If a change is detected, the application saves the new snapshot.
    - If no change is detected, no further action is taken.
- If a new snapshot was created, the application determines if the node has an existing baseline. If the node lacks a baseline, no further action is taken.
- If the node has a baseline, the TOE determines whether to promote the new snapshot to the baseline by comparing it to the node's current baseline.
    - If the new snapshot is identical to the baseline, the TOE auto-promotes the new version to the baseline, updating the baseline date and time.
    - If the new element version differs from the baseline, the application executes all actions associated with the applicable rule (such as notifying users, restoring the configuration of the node, taking the node offline, etc.).

Authorized administrators use the TOE's Graphical User Interface to configure the integrity checking mechanism, creating integrity check rules that specify node elements and corresponding attributes to monitor. Note that the

[9] The time between snapshots is a window of time in which an attribute could be changed and then changed back with no detection.

administrator can make changes to node baselines in addition to using the TOE to perform integrity checks at regular intervals.

The above operations are depicted in the figure below.



The TOE can monitor various types of nodes and node elements. While some types of nodes require the installation of a Tripwire Enterprise Agent on the node, other node types provide suitable interfaces and do not require the installation of a Tripwire Enterprise Agent. In the latter case, the monitoring is handled by the Tripwire Enterprise Agentless Node subcomponent. The following node types and elements can by monitored:

- Agent Nodes requiring Tripwire Enterprise Agents (mostly operating systems)
    - Files (the file element attributes monitored are listed below)
    - Directories (the directory element attributes monitored are listed below)
    - Registry keys and values (for the Windows operating system only, the registry element attributes monitored are listed below)
- Agentless Nodes
    - Files content attributes
    - Command output (run command and capture output to check node settings or parameters)
    - Availability (network connectivity)

In the case of nodes hosting Tripwire Enterprise Agents, the TOE can monitor file, directory, and registry keys/values as follows:

- UNIX file element attributes monitored:
    - The access control list for a file or directory
    - The last date and time when a file or directory was accessed
    - The last date and time when file or directory metadata was modified (or created)
    - The UNIX user group that owns a file or directory
    - The MD5[10] hash for a file
    - The last time file or directory content was changed by a user
    - A hash that associates a file with a software-installation package
    - Permission and file mode bits
    - The SHA-1 hash for a file
    - The size of a file
    - The owner of the file or directory
- Windows file element attributes monitored:
    - The last time a file or directory was accessed by a user
    - Archive flag
    - A flag that indicates whether the file or directory is compressed
    - The date and time when a file or directory was created
    - A list that specifies the level of file or directory access granted to Windows users or user groups
    - The Windows user group that owns a file or directory
    - Hide flag
    - The MD5 hash of a file
    - Offline flag
    - The owner of the file
    - A hash or version string that associates a file with a software-installation package

[10] The cryptography used in this product has not been FIPS certified nor has it been analyzed or tested to conform to cryptographic standards during this evaluation. All cryptography has only been asserted as tested by the vendor.

  - Read-only flag
  - A list that controls the generation of audit log entries for attempts to access a securable object.
  - The SHA-1[11] hash of a file
  - The size of a file
  - The number of alternate data streams on a file or directory
  - The MD5 hash for the file or directory alternate data stream(s)
  - The SHA-1 hash for the file or directory alternate data stream(s)
  - System flag
  - Temp flag
  - The date and time when file or directory content was last changed
- Windows registry key element attributes monitored:
  - A list that specifies the level of access granted to Windows users or user groups
  - Indicates the type of data in a value
  - The Windows user or user group that owns a registry key
  - The MD5 hash of data in a registry value
  - The owner of a registry key
  - A string that associates a registry key or value with a software-installation package
  - A list that controls the generation of audit log entries for attempts to access a registry key
  - The SHA-1 hash of data in a value
  - The size of data in a value
  - The date and time when a key was last changed

Authorized administrators configure the integrity checking mechanism by specifying actions to take in response to integrity checks. For actions relying upon an external IT entity (i.e., email, SNMP, syslog), the TOE is only capable of sending the integrity check results or log message. The TOE relies upon the IT environment to complete delivery. The TOE is capable of the following actions when a change is detected.

- Display integrity check results to the console
- Send integrity check results to administrators using email
- Send integrity check results to administrators using SNMP
- Send a log message to a Syslog server
- Execute a command on either the Tripwire Enterprise Server or a node using an agent
- Promote new element versions to baseline
- Restore a changed element to its baseline state (although not all types of elements can be restored).

Note that the TOE provides a command line interface (CLI) that can be used instead of the administrative graphical user interface (GUI) to perform some administration functions on the TOE. However, the CLI is a subset of the GUI and cannot perform all administration functions.

For more information about TOE administration (including more information about integrity check policies), see the user data protection and security management descriptions below.

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

- CHG_COL.1(EX): The Tripwire Enterprise Agents component of the TOE contributes to the monitoring of files, directories, and registry keys and values of targeted IT system resource(s) by collecting object

[11] The cryptography used in this product has not been FIPS certified nor has it been analyzed or tested to conform to cryptographic standards during this evaluation. All cryptography has only been asserted as tested by the vendor.

attribute information. The collected information is stored in none and node group objects in the Firebird database. The Tripwire Enterprise Agentless Node subcomponent of the Tripwire Enterprise Server component can monitor files, command output, and availability of targeted IT system resource(s) by collecting object attribute information.

- CHG_ASM.1(EX): The Tripwire Enterprise Server component or the Tripwire Enterprise Agent can compare collected attribute information from monitored IT system resources using administrator-configured rules. These rules determine the frequency of the monitoring activity and the action taken by the TOE when a change is detected.
- CHG_REP.1(EX): The Tripwire Enterprise Server component can perform actions in response to element attribute comparisons, specifically: display integrity check results to the console, send integrity check results to administrators using email, send integrity check results to administrators using SNMP, send a log message to a Syslog server, execute a command on either the Tripwire Enterprise Server host operating system or on the Tripwire Enterprise Agent host operating system, promote new element versions to baseline, restore a changed element to its baseline state (note that not all types of elements can be restored).

6.1.2 Security audit

The TOE provides its own audit mechanism that can generate audit records containing monitored node integrity check results and TOE management actions. The TOE stores its audit records in the Firebird Database.

Each audit record includes date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event. The auditable events include:

- Results of an integrity check on a monitored node.
- Successful requests to perform an operation on an object covered by the SFP
- Use of the management functions:
  - Specification of integrity check rules,
  - Specification of integrity check actions,
  - Promotion of collected object attributes to baselines,
  - Review of integrity check reports.
- Successful and unsuccessful user identification and authentication.

The audit function always runs when the TOE is running. It cannot be stopped or (re)started. This prevents the audit trail from having gaps when the TOE is running.

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

- FAU_GEN.1(EX): The TOE generates audit records for TOE management events and for unsuccessful user identification and authentication events.
- FAU_SAR.1: The TOE provides administrators the ability to read from the audit trail using administrator console interfaces.
- FAU_SAR.3: The TOE provides administrators the ability to search and sort audit data using administrator console interfaces based on date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event.

6.1.3 User data protection

The TOE implements a discretionary access control (DAC) policy for object access based on:

- user identities,
- user group memberships,

- user roles, and
- Access Control Lists (ACLs).

The TOE objects directly subject to this policy are nodes and node groups. There are no other objects. A node is represented by the network address of a server, router, switch, firewall, or load balancer that contains objects that Tripwire Enterprise may monitor. A node group is an object containing a collection of nodes. Access controls on individual nodes always overrides access controls on node groups.

The TOE stores access permissions for applicable objects in access control lists (ACLs) and provides the ability to grant and revoke permissions. The user identity, group and role associated with the user identity, are used by the DAC mechanism to validate the user's permission to access the applicable objects. Users may be granted the following permissions:

- Load permissions – provide read-only access to a class of Tripwire Enterprise objects and groups. For instance, the load rules permission grants access to the Rule Manager. In the Rule Manager, users can review all rules and rule groups.
- Create permissions – authorize users to create a class of Tripwire Enterprise objects and groups. For example, the create nodes permission authorizes users to create nodes and node groups.
- Delete permissions – authorize users to permanently remove objects or groups from the system. For instance, with the delete nodes permission, one can delete both nodes and node groups.
- Update permissions – enable users to modify the properties of a class of Tripwire Enterprise objects and groups. For example, one can change the properties of nodes and node groups with the update nodes permission.
- Manage permissions – authorize users to modify Tripwire Enterprise settings or parameters..

While user identities can be used in ACLs to assigned specific access permissions to specific users, the TOE also supports a 'group' feature. A group is a special identity that is allowed to have members. Note that both users and groups can be members of groups and each user or group can be a member of multiple groups. Authorized system or group administrators may grant group membership to a user.

User groups provide a convenient way to grant and revoke permissions to more than one user in a single statement. If a user is a member of a user group that does not have access to an object, but the user has been explicitly granted access to the object, the user will be able to access the object. Permissions granted to specific users override permissions granted to user groups.

Users acquire permission based upon the role assigned to their user account. An access control associates a role with a user (or user group) and a node (or node group).

The following process is used to determine permissions.

1. If a user is identified in an access control for a specific node, the role associated with that access control becomes the user’s permission to that node, otherwise.
2. If a user group is identified in an access control for a specific node, the role associated with that access control becomes the user’s permission to that node, otherwise.
3. If a user is identified in an access control for a specific node group, the role associated with that access control becomes the user’s permission to that node, otherwise.
4. If a user group is identified in an access control for a specific node group, the role associated with that access control becomes the user’s permission to that node, otherwise.
5. Repeat the previous 2 steps for each containing node group.
6. If no permissions are found apply permissions based upon the user’s role.

Note: If a user accesses a node contained within multiple, nested node groups with access controls, the applicable access control is determined by proximity to the node. In other words, the access control of the lowest node group in the node hierarchy determines if the user has permission to the node and, if so, what permission is granted the user.

The default administrator account has a privilege associated with it that allows discretionary access override, meaning that it can access any object even if the default administration account does not otherwise have access to the object. This makes some configuration errors correctable which would not be correctable otherwise. The default administrator account is the one created when the TOE is installed.

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

- FDP_ACC.2: All server subjects are subject to the DAC policy for all available operations on nodes and node groups (and their contents).
- FDP_ACF.1: Server objects have ACLs and can define groups and these attributes are compared against user identities in order to determine whether the request operation should be allowed. Alternately, a user may have a role that explicitly grants the requested access regardless of the normal access check. If both of these checks fail, access will be refused.

6.1.4 Identification and authentication

The TOE provides its own identification and authentication mechanism. In order to access the Tripwire Enterprise Server, a login account, including a login name and password, must be created for the user. User accounts can be established as either regular user accounts or administrator accounts via the assignment of roles, as described in the Security Management function (below). User login names and hashed passwords are stored as part of the TOE's configuration data.

To login to the TOE using the graphical user interface (GUI), the user provides the login name and password at the prompt. The TOE hashes the password and compares the resulting value to that stored in the TOE configuration data. If either the login name or password is incorrect the login request will fail and no functions will be made available. If a user should execute 3 failed login attempts within a single session, the user will be delayed for 30 seconds. The user's account will not be locked out or disabled. As a result of a successful login, a subject is created on behalf of the client.

To access the TOE using the command line interface (CLI), the user provides the login name and password as arguments to every CLI command that accesses a TOE object. The TOE's CLI interface provides a shortcut, however, letting the user set the login name and password, after which the CLI will automatically add these values as userid and password arguments to every command. Guidance recommends that users provide their userid and password manually with each command they enter.

In addition to user name and password, any groups and authorities assigned to the user are also stored as a part of the TOE configuration data. Note that groups are used to simplify access control management. Authorities define the roles and permissions available to the user.

User passwords must be at least six characters[12] and less than 24 characters. There are 94 possible characters that may be used in a user password (i.e., 26 alphabetic characters, 10 numeric characters, and 32 non alpha-numeric characters)

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

- FIA_ATD.1: The TOE defines user identities, authentication data, groups, and role information.
- FIA_UAU.2: The TOE offers no TSF-mediated functions until the user is authenticated.
- FIA_UID.2: The TOE offers no TSF-mediated functions until the user is identified.

6.1.5 Security management

The TOE provides two sets of interfaces to control how it operates: a graphical user interface (GUI), provided by a built-in web server, and a command-line interface. The GUI is a full-functioned interface from which a user with appropriate permissions can completely administer the TOE. The CLI provides a subset of the GUI functionality, which is insufficient to completely administer the TOE. The CLI, for example, provides no functionality to add or

[12] Note that Tripwire provides additional recommendations on a stronger password, but these recommendations are not required to provide the required strength of function for this mechanism.

delete users or to change user passwords (these functions are available only through the GUI). When each interface provides access to the same administrative functions they have the same restrictions. The CLI is provided to support the execution of remote scripts.

A user permission is a system authorization which enables a user to view, add, change, or delete data in Tripwire Enterprise. A user role is a collection of user permissions that may be assigned to a user. Tripwire Enterprise includes five default user roles:

- Administrator
- Power User
- Regular User
- Monitor User
- User Administrator

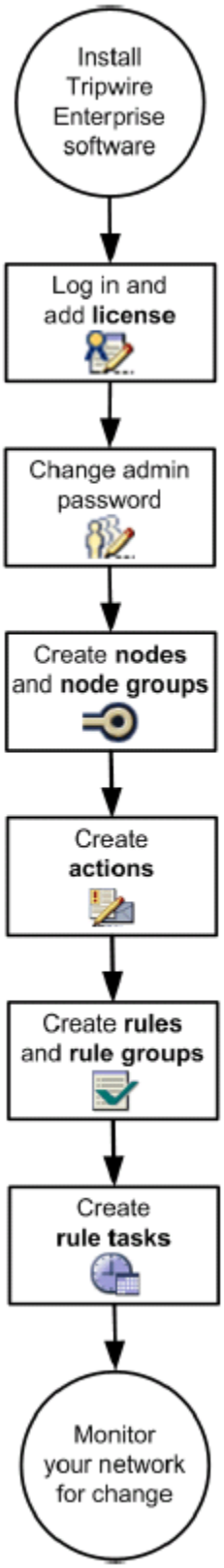
Four of the default user roles (Administrator, Power User, Regular User, and Monitor User) are organized hierarchically. In other words, each role possesses the permissions granted to lesser roles, as well as an additional set of permissions. The Administrator role has the most permissions, followed by the Power User role, the Regular User role and the Monitor User role has the least permissions. The User Administrator role is orthogonal to the other roles and has permissions to manipulate user accounts.

When the TOE is installed, the nodes to be monitored must be added to the TOE configuration. Either Agentless or Agent nodes can be added. After nodes are added, rules are created to specify which elements on each node are to be monitored. Actions can then be created to cause the TOE to take remedial measures in response to changes detected by integrity checks. After actions are added, rules are defined that specify how the TOE will check selected elements for changes. Integrity checks of selected nodes can then be scheduled by creating one or more Rule Tasks.

Rule Tasks are used to schedule integrity checks of nodes and/or node groups. An integrity check starts by taking a snapshot of (collecting a set of object attributes from) a node. The snapshot is then compared to a previous snapshot that has been saved as a baseline, using snapshot check rules established by the administrator.

Administrators can create baselines when a rule task is created (initialize baselines) or after the previous steps have been completed. Administrators can also promote a collected set of object attributes (snapshot) to baseline status at any time. Both of these actions require the Administrator role.

The above operations are depicted in the figure below.


Install Tripwire Enterprise software
Log in and add **license**
Change admin password
Create **nodes** and **node groups**
Create **actions**
Create **rules** and **rule groups**
Create **rule tasks**
Monitor your network for change

Objects do not have an ACL assigned to them when they are created. This prevents them from being accessed by any subject. Access can be granted subsequently to specific users. The first such access specified creates an ACL for the object.

When the TOE reporting mechanism is configured, the Administrator role can define reports with varying levels of detail about the results of integrity checks as follows:

- Change Process Compliance - This report identifies authorized and unauthorized changes to specified nodes over a period of time. An authorized change is associated with a valid change request ticket ID.
- Change Rate - This report shows the total number of changes (additions, removals, and modifications) detected on specified nodes over a period of time. Within the selected time period, the report displays the number of detected changes at a regular interval (or 'frequency'); for instance, daily, weekly, or monthly.
- Change Variance - This report shows the total number of rules and elements associated with detected changes on specified nodes. As appropriate, you can limit report output to specific nodes, rules, and/or element names. Typically, this report is executed immediately after deployment of a patch or other software package. To determine which new element versions should be promoted, you may review the report for inconsistencies across the updated systems.
- Changed Elements - This report lists all changed elements identified by the specified criteria. Report output specifies exactly which attributes changed for each element.
- Changes by Node or Group - This report displays the number of changes detected on one or more nodes (or node groups). The change comparison calculates the total number of changes for each node, as well as the totals for each type of change (added, removed, or modified).
- Changes by Severity - This report shows the total number of changes detected on one or more nodes (or node groups) that fall within a specified range of severity levels.
- Detailed Changes - This report compiles comprehensive change information for elements on specified nodes.
- Device Inventory - This report identifies the make, model, and version of specified nodes.
- Frequently Changed Nodes - This report ranks the most frequently changed nodes that meet the specified criteria. The report includes the total number of changes for each node, as well as the totals for each type of change (added, removed, or modified).
- Inventory Changes - For your Tripwire Enterprise implementation, this report calculates the number of nodes that have been added, modified, and deleted over a specified period of time.
- Monitoring Policy - This report identifies the criteria set associated with one or more file system rules or Windows registry rules and, optionally, the times when those rules should apply.
- Nodes with Changes - For the specified criteria, this report identifies the number of nodes that have changed over a given period of time.
- Reference Node Variance - This report identifies all elements that differ between one node (the reference node) and another (the compare node). In a single report, the reference node may be compared with one or more compare nodes.
- System Access Control - This report provides security-related information on specified user accounts, user roles, user groups, and/or access controls.

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

- FMT_MOF.1: The TOE restricts the ability to enable and disable integrity check rules and reporting options, to the Administrator and Power User roles.
- FMT_MSA.1a: The ability to manage object attributes is restricted by enforcing the permissions of the Administrator role.

- FMT_MSA.1b: The ability to manage user account attributes is restricted by enforcing the permissions of the Administrator and a User Administrator role.
- FMT_MSA.3: By default every object is created without an ACL, which prevents access by all users. Subsequently, permissions can be explicitly added to the ACL to grant access to specific users.
- FMT_MTD.1a: The TOE restricts the ability to include or exclude integrity check rules in the active configuration of the TOE by enforcing the permissions of the Administrator role, the Power User role, and the Regular User role.. Rules that have been excluded are not executed. Rules that have been included will be executed.
- FMT_MTD.1b: The TOE restricts the ability to include or exclude integrity check actions in the active configuration of the TOE by enforcing the permissions of the Administrator role, the Power User role, the Regular User role and the Monitor User role.. Integrity check actions that have been excluded are not executed. Integrity check actions that have been included will be executed.
- FMT_MTD.1c: The TOE restricts the ability to promote object attribute snapshots by enforcing the permissions of the Administrator role, and the Power User role.
- FMT_MTD.1d: The TOE restricts the ability to query or clear integrity check reports to authorized administrators by enforcing the permissions of the Administrator role.
- FMT_SMF.1: The TOE provides administrators with the ability to perform all management functions, including: specification of integrity check rules and actions, scheduling of integrity checks, promoting object attribute snapshots to baselines, and querying or clearing integrity check reports stored in the database.
- FMT_SMR.1: The TOE defines user accounts that can be assigned a system-defined role. The user roles provided by the TOE are Administrator, Power User, Regular User, Monitor User, and User Administrator.

6.1.6 Protection of the TSF

The TOE is an application that executes within a dedicated machine running a host operating system. The TOE is instantiated as services on Windows platforms and as daemons on UNIX-based platforms. The remainder of this discussion will refer to “process” as a non-platform specific term for “service” and “daemon”. The Tripwire Enterprise Server runs on a JVM in a host operating system provided process The host operating system is expected to provide process isolation to each process with its own unique address space and separation from all other processes.

Tripwire Enterprise Server is a JAVA program that runs in a Tripwire supplied JVM. The Tripwire Enterprise Server manages its users internally by adding them as a Principal to the Java Virtual Machine's AccessControlContext for the thread and then using that information when determining a user's permission to perform an operation in the system. Each thread has a structure that tracks security information, Tripwire Enterprise Server populates this structure with the correct user identity and uses that identity when determining a user's permission to perform an operation in the system.[13]

The Tripwire Enterprise Server separates user network connections based on individual administrative GUI and CLI connections. The TOE uses HTTPS to protect TSF and user data transmitted between the GUI and the user’s browser. It also uses HTTPS to protect TSF and user data transmitted between the CLI and the user’s shell. The TOE includes web server functionality and configures and uses the SSL functionality provided by the JVM.

The TOE uses SSL (provided by the JVM) to protect TSF data when it is being transmitted between Tripwire Enterprise Server and Tripwire Enterprise Agent components. The TSF Protection security function is supported by a mutually authenticated SSL connection between the Tripwire Enterprise Agent and the Tripwire Enterprise Server. Authentication is accomplished using X509 certificates for both the Tripwire Enterprise Agent and the Tripwire Enterprise Server.

[13] The TOE does not rely upon the JVM to enforce access controls, but merely utilizes JVM thread support data structures.

During installation of either the Tripwire Enterprise Server or a Tripwire Enterprise Agent, a *services passphrase* is collected from the installer. This *services passphrase* is used during the generation of an X.509 signing certificate[14] which is used to sign another X.509 certificate that is host specific. These certificates are created using 1024-bit DSA. The private key of the signing certificate is destroyed immediately after it has been used to sign the host-specific certificate. The public key of the signing certificate, the public key of the host-specific certificate, and the private key of the host-specific certificate are then stored in Sun JKS (JAVA Key Store) format and made available to the JVM. When the Tripwire Enterprise Server or Tripwire Enterprise Agent start, they ensure the JVM is using these certificates, and configure the JVM to require mutual authentication on SSL connections.

If the same *services passphrase* is used during installation of a Tripwire Enterprise Server and a Tripwire Enterprise Agent, then both will generate the same signing certificate, and both will recognize the signature authority on the other's host-specific certificate.

For SSL communications, Tripwire allows the JVM that is part of the Tripwire Enterprise Agent to use it's built-in SSL layer to auto-negotiate the connection's protocol and cipher-strength. Tripwire does however apply the following criteria for narrowing down the list of ciphers which the JVM enables by default. The two filters are:

1) The symmetric algorithm must use a key of at least 128 bits.
2) Tripwire does not allow anonymous/unauthenticated connections (which would allow a man-in-the-middle attack if not screened)

The TOE provides identification and authentication on both of its admin interfaces, thereby preventing circumvention of the access control mechanism.

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

- FPT_ITT.1a: The TOE configures the SSL provided by the JVM to ensure connections between Tripwire agents and the Tripwire Server use the filters listed above.
- FPT_RVM.1: Users of the TOE can access TOE commands only through one of the two administration interfaces provided. The TOE enforces Commands issued by a user are processed within the TE Server such that the TOE's enforcement of access control cannot be bypassed.

6.2 TOE Security Assurance Measures

6.2.1 Configuration management

The configuration management measures applied by Tripwire ensure that configuration items are uniquely identified, and that documented procedures are used to control and track changes that are made to the TOE. Tripwire ensures changes to the implementation representation are controlled. Tripwire performs configuration management on the TOE implementation representation, design documentation, tests and test documentation, user and administrator guidance, delivery and operation documentation, life-cycle documentation, vulnerability analysis documentation, and configuration management documentation.

These activities are documented in:

- Tripwire, Inc. Tripwire Enterprise- 5.2, Tripwire for Servers 4.6, Tripwire Manager 4.6 Configuration Management Plan

The Configuration management assurance measure satisfies the following EAL 3 augmented with ALC_FLR.2 assurance requirements:

- ACM_CAP.3
- ACM_SCP.1

[14] A signing certificate is a certificate that will be used to sign another certificate.

6.2.2 Delivery and operation

Tripwire provides delivery documentation and procedures to identify the TOE, secure the TOE during delivery, and provide necessary installation and generation instructions. Tripwire's delivery procedures describe all applicable procedures to be used to prevent in appropriate access to the TOE. Tripwire also provides documentation that describes the steps necessary to install Tripwire Enterprise in accordance with the evaluated configuration.

These activities are documented in:

- Tripwire Manager and Tripwire for Servers Delivery Procedures Delivery Procedures [15]
- Tripwire Enterprise Installation Guide 5.2
- Tripwire Enterprise 5.2 Release Notes Addendum

The Delivery and operation assurance measure satisfies the following EAL 3 augmented with ALC_FLR.2 assurance requirements:

- ADO_DEL.1
- ADO_IGS.1

6.2.3 Development

Tripwire has numerous documents describing all facets of the design of the TOE. In particular, they have a functional specification that describes the accessible TOE interfaces; a high-level design that decomposes the TOE architecture into subsystems and describes each subsystem and their interfaces; and, correspondence documentation that explains how each of the design abstractions correspond from the TOE summary specification in the Security Target to the subsystems.

These activities are documented in:

- Tripwire Enterprise- v5.2 Design Document (HLD, FSP, and RCR)
- Tripwire Enterprise- 5.2 Reference Guide
- Tripwire Enterprise- 5.2 User Guide

The Development assurance measure satisfies the following EAL 3 augmented with ALC_FLR.2 assurance requirements:

- ADV_FSP.1
- ADV_HLD.2
- ADV_RCR.1

6.2.4 Guidance documents

Tripwire provides administrator and user guidance on how to utilize the TOE security functions and warnings to administrators and users about actions that can compromise the security of the TOE. The Tripwire Enterprise 5.2 Release Notes Addendum contains details regarding the common criteria specific instructions and warning provided to administrators.

These activities are documented in:

- Tripwire Enterprise- 5.2 Reference Guide
- Tripwire Enterprise- 5.2 User Guide
- Tripwire Enterprise 5.2 Release Notes Addendum
- Tripwire Enterprise Installation Guide 5.2

[15] The delivery procedures in this document apply to the product defined by this security target.

The Guidance documents assurance measure satisfies the following EAL 3 augmented with ALC_FLR.2 assurance requirements:

- AGD_ADM.1
- AGD_USR.1

6.2.5 Life cycle support

Tripwire ensures the adequacy of the procedures used during the development and maintenance of the TOE through its life-cycle. Tripwire includes security controls on the development environment that are adequate to provide the confidentiality and integrity of the TOE design and implementation that is necessary to ensure the secure operation of the TOE. In addition, Tripwire identifies and tracks reported flaws, ensuring that they are addressed and corrections and corrective measures are made available as applicable.

These activities are documented in:

- Tripwire, Inc. Tripwire Enterprise 5.2, Tripwire for Servers 4.6, Tripwire Manager 4.6 Lifecycle

The Life cycle support assurance measure satisfies the following EAL 3 augmented with ALC_FLR.2 assurance requirements:

- ALC_DVS.1
- ALC_FLR.2

6.2.6 Tests

Tripwire has a test plan that describes how each of the necessary security functions is tested, along with the expected test results. Tripwire has documented each test as well as an analysis of test coverage and depth demonstrating that the security aspects of the design evident from the functional specification and high-level design are appropriately tested. Actual test results are created on a regular basis to demonstrate that the tests have been applied and that the TOE operates as designed.

These activities are documented in:

- Tripwire Inc. Tripwire Enterprise- 5.2 common Criteria Test Plan
- Tripwire Enterprise- Test Results (spreadsheets and HTML pages)

The Tests assurance measure satisfies the following EAL 3 augmented with ALC_FLR.2 assurance requirements:

- ATE_COV.2
- ATE_DPT.1
- ATE_FUN.1
- ATE_IND.2

6.2.7 Vulnerability assessment

The TOE administrator and user guidance documents describe the operation of Tripwire Enterprise and how to maintain a secure state. These guides also describe all necessary operating assumptions and security requirements outside the scope of control of the TOE. They have been developed to serve as complete, clear, consistent, and reasonable administrator and user references.

Tripwire has conducted a strength of function analysis wherein all permutational or probabilistic security mechanisms have been identified and analyzed resulting in a demonstration that all of the relevant mechanisms fulfill the minimum strength of function claim, SOF-Basic.

Tripwire performs regular vulnerability analyses of the entire TOE (including documentation) to identify obvious weaknesses that can be exploited in the TOE.

These activities are documented in:

- Tripwire Enterprise- Version 5.2 Vulnerability Assessment
- Tripwire, Inc. Tripwire Enterprise Strength of Function Analysis

The Vulnerability assessment assurance measure satisfies the following EAL 3 augmented with ALC_FLR.2 assurance requirements:

- AVA_MSU.1
- AVA_SOF.1
- AVA_VLA.1

7. Protection Profile Claims

There is no Protection Profile claim in this Security Target.

8. Rationale

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

- Security Objectives;
- Security Functional Requirements;
- Security Assurance Requirements;
- Strength of Functions;
- Requirement Dependencies;
- TOE Summary Specification; and,
- PP Claims.

8.1 Security Objectives Rationale

This section shows that all secure usage assumptions, organizational security policies, and threats are completely covered by security objectives. In addition, each objective counters or addresses at least one assumption, organizational security policy, or threat.

8.1.1 Complete Coverage – Threats

This section shows that all secure usage assumptions, organizational security policies, and threats are completely covered by security objectives. In addition, each objective counters or addresses at least one assumption, organizational security policy, or threat.

	O.AUDITING	O.AUTHENT	O.COLLECT	O.COMPARE	O.DAC	O.MANAGE	O.PROTECT	OE.PROTECT	OE.TIME	OE.DBCOMM	OE.PROTDB
T.AUTHENT	x	x							x	x	x
T.COLLECT			x	x							
T.MANAGE						x					
T.PROTECT					x		x	x			

8.1.1.1 T.AUTHENT

An authorized user may be unaware of an inadvertent change TOE data or functions they are authorized to modify.

This Threat is satisfied by ensuring that:

- O.AUDITING: The TOE audits security-relevant events.
- O.AUTHENT: The TOE identifies users.
- OE.TIME: The IT environment provides reliable time for use in audit records.

- OE.DBCOMM The IT environment provides protection of the communications between the TOE and the external database from disclosure and modification
- OE.PROTDB The IT environment provides protection of external database such that only the TOE can access the information stored by the TOE in the external database.

Audit records with reliable time stamps and user identification can prevent authorized users from making changes to the system that reduce security by allowing them to be held accountable for their actions. Auditing also helps users and administrators identify and correct inadvertent changes to the system, so an insecure state does not persist.

8.1.1.2 T.COLLECT

An attacker may be able to inappropriately change attribute information for targeted objects without being detected.

This Threat is satisfied by ensuring that:

- O.COLLECT: The TOE collects attribute information for targeted objects.
- O.COMPARE: The TOE compares collected attribute information for targeted objects against stored baselines.

The TOE regularly collects configuration information from the nodes it protects (targeted objects) and compares it to the baseline configuration information for each node that it has previously collected and stored. The baseline information is assumed to represent an authorized configuration because it was asserted to be such when the node was first put under TOE management and the TOE has managed the configuration since then, allowing only authorized changes. As a result, an unauthorized change to the configuration of a node by an attacker will be recognized, (optionally) repaired, and an alert generated, allowing the node's configuration to be returned to a secure state and the method by which it was changed without authorization to be investigated.

8.1.1.3 T.MANAGE

An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.

This Threat is satisfied by ensuring that:

- O.MANAGE: The TOE provides administrative interfaces that can be used to administer its security functions, and user and administrator guidance showing how to use the interfaces to install and configure the system securely.

8.1.1.4 T.PROTECT

An attacker may be able to gain unauthorized access to data collected from targeted objects.

This Threat is satisfied by ensuring that:

- O.PROTECT: The TOE protects collected attribute information for targeted objects.
- O.DAC: The TOE protects targeted objects configuration information.
- OE.PROTECT: The IT environment protects the TOE, TOE assets and inter-TOE communications from disclosure, interference, and tampering.

8.1.2 Complete Coverage – Environmental Assumptions

This section provides evidence demonstrating coverage of the Non-IT security objectives by the environmental assumptions. The following table shows this assumption to objective mapping.

		OE.COMSEC	OE.DEDICATED	OE.INSTALL	OE.PHYCAL	OE.CREDEN	OE.PERSON	OE.INTROP
Intended usage assumptions	**A.ACCESS**							X
	A.ASCOPE							X
	A.DYNMIC						X	X
	A.DEDICATE		X					
Physical assumptions	**A.LOCATE**				X			
	A.PROTCT				X			
	A.NETOK	X						
	A.MANAGE						X	
Personnel assumptions	**A.NOEVIL**			X	X	X	X	
	A.NOTRST				X	X		

8.1.2.1 A.ACCESS

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

This Assumption is satisfied by ensuring that:

- OE.INTROP: The OE.INTROP objective ensures the TOE has the needed access.

8.1.2.2 A.ASCOPE

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

This Assumption its corresponding non-IT objective, OE.INTROP, are taken from a validated PP (IDS). The purpose of this is to state the TOE can handle the traffic it is monitoring. This Assumption is satisfied by ensuring that:

- OE.INTROP: The OE.INTROP objective ensures the TOE is capable of handling the necessary interactions with the IT System it monitors.

8.1.2.3 A.DYNIMC

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

This Assumption is satisfied by ensuring that:

- OE.INTROP: The OE.INTROP objective ensures the TOE has the proper access to the IT System.
- OE.PERSON: The OE.PERSON objective ensures that the TOE will be managed appropriately.

8.1.2.4 A.DEDICATE

The host on which the TE server component of the TOE resides does not provide a general purpose computing environment to untrusted users.

This Assumption is satisfied by ensuring that:

- OE.DEDICATED: Those responsible for managing the TOE will ensure that the host on which the Tripwire Enterprise Server resides does not provide a general purpose computing environment to untrusted users..

8.1.2.5 A.LOCATE

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

This Assumption is satisfied by ensuring that:

- OE.PHYCAL: The TOE is protected from any physical attack.

8.1.2.6 A.PROTCT

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

This Assumption is satisfied by ensuring that:

- OE.PHYCAL: The OE.PHYCAL provides for the physical protection of the TOE.

8.1.2.7 A.NETOK

Those responsible for managing the agentless nodes have taken steps to secure the communication pathways between the TOE and the agentless nodes per their security environment.

This Assumption is satisfied by ensuring that:

- OE.COMSEC: Those responsible for managing the agentless nodes have taken steps to secure the communication pathways between the TOE and the agentless nodes per their security environment.

8.1.2.8 A.MANAGE

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

This Assumption is satisfied by ensuring that:

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

8.1.2.9 A.NOEVIL

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

This Assumption is satisfied by ensuring that:

- OE.INSTALL: The OE.INSTAL objective ensures that the TOE is properly installed and operated.
- OE.PHYCAL: The OE.PHYCAL objective provides for physical protection of the TOE by authorized administrators.
- OE.CREDEN: The OE.CREDEN objective supports this assumption by requiring protection of all authentication data
- OE.PERSON: Personnel working as administrators will be carefully selected and trained.

This assumption requires the TOE to prevent attackers from becoming authorized administrators by obtaining and using administrator credentials. This requires the TOE to be physically protected, to be properly installed and operated, and that the administration credentials be adequately protected. This assumption also requires administrators who might be willfully negligent or hostile to be avoided and that administrators be trained and motivated to manage the TOE according to the TOE documentation.

8.1.2.10 A.NOTRST

The TOE can only be accessed by authorized users.

This Assumption is satisfied by ensuring that:

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

8.2 Security Requirements Rationale

This section provides evidence supporting the internal consistency and completeness of the components (requirements) in the Security Target. Note that
Table 4 indicates the requirements that effectively satisfy the individual objectives.

8.2.1 Security Functional Requirements Rationale

All Security Functional Requirements (SFR) identified in this Security Target are fully addressed in this section and each SFR is mapped to the objective(s) that it is intended to satisfy.

	O.AUDITING	O.AUTHENT	O.COLLECT	O.COMPARE	O.DAC	O.MANAGE	O.PROTECT	OE.PROTECT	OE.TIME	OE.DBCOMM	OE.PROTDB
CHG_COL.1(EX)			X								
CHG_ASM.1(EX)				X							
CHG_REP.1(EX)				X							
FAU_GEN.1(EX)	X					X					
FAU_SAR.1	X					X					
FAU_SAR.3	X					X					
FDP_ACC.2					X		X				
FDP_ACF.1					X		X				
FIA_ATD.1					X						
FIA_UAU.2		X			X						
FIA_UID.2		X			X						
FMT_MOF.1						X					
FMT_MSA.1					X	X					
FMT_MSA.3					X	X					
FMT_MTD.1a						X					
FMT_MTD.1b						X					
FMT_MTD.1c						X					
FMT_MTD.1d						X					
FMT_SMF.1						X					
FMT_SMR.1						X					
FPT_ITT.1a							X				
FPT_RVM.1							X				
FPT_SEP.1								X			
FAU_STG.1										X	X
FPT_STM.1									X		
FPT_ITT.1b								X			

Table 4 Objective to Requirement Correspondence

8.2.1.1 O.AUDITING

The TOE shall provide the capability to create records containing integrity check results and security-relevant events associated with users.

This TOE Security Objective is satisfied by ensuring that:

- FAU_GEN.1(EX): The TOE generates audit events for integrity check results and TOE management events.
- FAU_SAR.1: The TOE allows administrators to read and interpret all of the audit records.
- FAU_SAR.3: The TOE provides the ability to perform audit record searching and sorting on date and time, type, subject identity, and outcome (success or failure) of an event.

8.2.1.2 O.AUTHENT

The TOE shall verify the claimed identity of users.

This TOE Security Objective is satisfied by ensuring that:

- FIA_UAU.2: The TOE offers no TSF-mediated functions until the user is authenticated.
- FIA_UID.2: The TOE offers no TSF-mediated functions until the user is identified.

8.2.1.3 O.COLLECT

The TOE shall collect attribute information for targeted objects and maintain a baseline of attributes for each.

This TOE Security Objective is satisfied by ensuring that:

- CHG_COL.1(EX): The TOE can collect data indicating changes to files, directories, registry keys, command output, and availability of targeted IT system resource(s).

8.2.1.4 O.COMPARE

The TOE shall perform integrity checks on targeted objects by comparing collected attributes of each object against its stored baseline and generating a report containing integrity check results.

This TOE Security Objective is satisfied by ensuring that:

- CHG_ASM.1(EX): The Tripwire Enterprise Server component can compare collected attribute information from monitored IT system resources using administrator-configured rules.
- CHG_REP.1(EX): The Tripwire Enterprise Server component can perform actions in response to object attribute comparisons, specifically: display integrity check results to the console, send integrity check results to administrators using email, send integrity check results to administrators using SNMP, send a log message to a Syslog server, execute a command on the Tripwire Server using a rule (e.g. generate operating system audit events containing integrity check results), execute a command on either the Tripwire Enterprise Server or a node using an agent, promote new element versions to baseline, restore a changed element to its baseline state (note that not all types of elements can be restored).

8.2.1.5 O.DAC

The TOE shall control access to resources based upon the identity of users or groups of users.

This TOE Security Objective is satisfied by ensuring that:

- FDP_ACC.2: All server subjects are subject to the DAC policy for all available operations on nodes and node groups (and their contents).
- FDP_ACF.1: Server objects have ACLs and can define groups and these attributes are compared against user identities in order to determine whether the request operation should be allowed. Alternately, a user may have a role that explicitly grants the requested access regardless of the normal access check. If both of these checks fail, access will be refused.
- FIA_ATD.1: The TOE maintains security attributes for each user, including user identity, authentication data, group memberships, and roles.
- FIA_UAU.2: The TOE successfully authenticates users before allowing any other TSF-mediated action for that user.
- FIA_UID.2: The TOE identifies users before allowing any other TSF-mediated actions for that user.
- FMT_MSA.1: The ability to manage subject attributes is restricted to an administrator.
- FMT_MSA.3: By default every server object is created without an ACL. Subsequently, access can be granted to other users, but there is no method to specify access other than the default during creation.

8.2.1.6 O.MANAGE

The TOE shall provide functions such that it can be managed by authorized users.

This TOE Security Objective is satisfied by ensuring that:

- FAU_GEN.1(EX): The TOE generates audit records for the necessary events and containing sufficient information to allow administrators to effectively manage the security of the TOE.
- FAU_SAR.1: The TOE provides administrators the ability to read from the audit trail using administrator console interfaces.
- FAU_SAR.3: The TOE provides administrators the ability to search and sort audit data using administrator console interfaces based on date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event.
- FMT_MOF.1: The TOE restricts the ability to enable and disable integrity check rules and reporting options, by restricting access to the administrative graphical user interface.
- FMT_MSA.1: The ability to manage subject attributes is restricted to an administrator.
- FMT_MSA.3: By default every server object is created without an ACL. Subsequently, access can be granted to other users by the default administrator account, but there is no method to specify access other than the default during creation.
- FMT_MTD.1a: The TOE restricts the ability to include or exclude integrity check rules to authorized administrators by restricting access to the administrative graphical user interface.
- FMT_MTD.1b: The TOE restricts the ability to include or exclude integrity check reporting actions to authorized administrators by restricting access to the administrative graphical user interface.
- FMT_MTD.1c: The TOE restricts the ability to promote object attribute snapshots by restricting access to the administrative graphical user interface.
- FMT_MTD.1d: The TOE restricts the ability to query or clear integrity check reports to authorized administrators by restricting access to the administrative graphical user interface.
- FMT_SMF.1: The TOE provides administrators with the ability to perform all management functions, including: specification of integrity check rules and actions, promoting object attribute snapshots to baselines, and querying or clearing integrity check reports stored in the database.

- FMT_SMR.1: The TOE defines user accounts that can be assigned a system-defined role. The user roles Administrator, Power User, Monitor User, and User Administrator are considered "authorized administrator" roles and any other user accounts are considered simply "users".

8.2.1.7 O.PROTECT

The TOE shall protect itself and its assets from external interference or tampering.

This TOE Security Objective is satisfied by ensuring that:

- FDP_ACC.2: All server subjects are subject to the DAC policy for all available operations on nodes and node groups (and their contents).
- FDP_ACF.1: Server objects have ACLs and can define groups and these attributes are compared against user identities in order to determine whether the request operation should be allowed. Alternately, a user may have a role that explicitly grants the requested access regardless of the normal access check. If both of these checks fail, access will be refused.
- FPT_ITT.1a: The TOE uses the mechanisms provided by the IT environment to protect communications between the Tripwire agents and the Tripwire Server
- FPT_RVM.1: The access control enforcing component of the TOE cannot be bypassed.

8.2.1.8 OE.PROTECT

The IT environment shall protect the TOE, TOE assets and intra-TOE communications from disclosure, interference, and tampering.

This IT Environment Security Objective is satisfied by ensuring that:

- FPT_ITT.1b: The IT environment provides the TOE with an SSL mechanism that the TOE configures to protect communications between the Tripwire Agents and the Tripwire Server
- FPT_SEP.1: The IT environment provides mechanisms that the TOE uses to implement and configure processes that are protected from inappropriate access.

8.2.1.9 OE.TIME

The IT environment will provide a time source that provides reliable time stamps.

This IT Environment Security Objective is satisfied by ensuring that:

- FPT_STM.1: The operating system provides reliable time stamps to the TOE.

8.2.1.10 OE.DBCOMM

The IT environment protects the external database such that communications between the TOE and the external database is protected from disclosure and modification.

This IT Environment Security Objective is satisfied by ensuring that:

- FAU_STG.1: The IT environment will provide disclosure and modification protection of the communications between the TOE and the external database.

8.2.1.11 OE.PROTDB

The IT environment must ensure that the external database is protected such that only the TOE can access the external database.

This IT Environment Security Objective is satisfied by ensuring that:

- FAU_STG.1: The network environment will be designed such that only the TOE can access the external database.

8.3 Security Assurance Requirements Rationale

The selected security assurance level is EAL3 augmented with ALC_FLR.2.

EAL3 was selected as the base assurance level because the TOE is a commercial product whose users require a commercial product with a moderate level of independently assured security. Tripwire Enterprise is targeted at a relatively benign environment with good physical access security and competent administrators. Within such environments it is assumed that attackers will have little attack potential. As such, EAL3 is appropriate to provide the assurance necessary to counter the limited potential for attack.

The base assurance level was augmented with ALC_FLR.2 because flaw remediation procedures provide greater assurance that security-related bugs will be fixed. This is an important assurance measure for a widely distributed commercial product.

8.4 Strength of Functions Rationale

The TOE is targeted at a generalized IT environment with good physical access security and competent administrators. Within such environments it is assumed that attackers will have a moderate attack potential. As such, a strength of function of 'medium' is appropriate for the intended environment. The only mechanism with a probabilistic or permutational component is user authentication (FIA_UAU.2).

8.5 Requirement Dependency Rationale

The following table demonstrates that all dependencies among the claimed security requirements are satisfied and therefore the requirements work together to accomplish the overall objectives defined for the TOE.

ST Requirement	CC Dependencies	ST Dependencies
CHG_COL.1(EX)	none	none
CHG_ASM.1(EX)	none	none
CHG_REP.1(EX)	none	none
FAU_GEN.1(EX)	FPT_STM.1	FPT_STM.1 (IT environment)
FAU_SAR.1	FAU_GEN.1	FAU_GEN.1(EX)
FAU_SAR.3	FAU_SAR.1	FAU_SAR.1
FDP_ACC.2	FDP_ACF.1	FDP_ACF.1
FDP_ACF.1	FDP_ACC.1 and FDP_MSA.3	FDP_ACC.2 (hierarchical) and FDP_MSA.3
FIA_ATD.1	none	none
FIA_UAU.2	FIA_UID.1	FIA_UID.2
FIA_UID.2	none	none
FMT_MOF.1	FMT_SMF.1 and FMT_SMR.1	FMT_SMR.1 and FMT_SMF.1
FMT_MSA.1a	(FDP_ACC.1 or FDP_IFC.1) and FMT_SMF.1 and FMT_SMR.1	FDP_ACC.2 (hierarchical) and FMT_SMF.1 and FMT_SMR.1
FMT_MSA.1b	(FDP_ACC.1 or FDP_IFC.1) and FMT_SMF.1 and FMT_SMR.1	FDP_ACC.2 (hierarchical) and FMT_SMF.1 and FMT_SMR.1
FMT_MSA.3	FMT_MSA.1 and FMT_SMR.1	FMT_MSA.1 and FMT_SMR.1
FMT_MTD.1a	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_MTD.1b	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_MTD.1c	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_MTD.1d	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_SMF.1	none	none

ST Requirement	CC Dependencies	ST Dependencies
FMT_SMR.1	FIA_UID.1	FIA_UID.2
FPT_ITT.1a	none	none
FPT_RVM.1	none	none
FPT_SEP.1	none	none
FAU_STG.1	FAU_GEN.1	FAU_GEN.1(EX)
FPT_STM.1	none	none
FPT_ITT.1b	none	none
ACM_CAP.3	ALC_DVS.1	ALC_DVS.1
ACM_SCP.1	ACM_CAP.3	ACM_CAP.3
ADO_DEL.1	none	none
ADO_IGS.1	AGD_ADM.1	AGD_ADM.1
ADV_FSP.1	ADV_RCR.1	ADV_RCR.1
ADV_HLD.2	ADV_FSP.1 and ADV_RCR.1	ADV_FSP.1 and ADV_RCR.1
ADV_RCR.1	none	none
AGD_ADM.1	ADV_FSP.1	ADV_FSP.1
AGD_USR.1	ADV_FSP.1	ADV_FSP.1
ALC_DVS.1	none	none
ALC_FLR.2	none	none
ATE_COV.2	ADV_FSP.1 and ATE_FUN.1	ADV_FSP.1 and ATE_FUN.1
ATE_DPT.1	ADV_HLD.1 and ATE_FUN.1	ADV_HLD.2 and ATE_FUN.1
ATE_FUN.1	none	none
ATE_IND.2	ADV_FSP.1 and AGD_ADM.1 and AGD_USR.1 and ATE_FUN.1	ADV_FSP.1 and AGD_ADM.1 and AGD_USR.1 and ATE_FUN.1
AVA_MSU.1	ADO_IGS.1 and ADV_FSP.1 and AGD_ADM.1 and AGD_USR.1	ADO_IGS.1 and ADV_FSP.1 and AGD_ADM.1 and AGD_USR.1
AVA_SOF.1	ADV_FSP.1 and ADV_HLD.1	ADV_FSP.1 and ADV_HLD.2
AVA_VLA.1	ADV_FSP.1 and ADV_HLD.1 and AGD_ADM.1 and AGD_USR.1	ADV_FSP.1 and ADV_HLD.2 and AGD_ADM.1 and AGD_USR.1

8.6 Explicitly Stated Requirements Rationale

The CHG class of explicitly stated security functional requirements captures the TOE's basic functionality for monitoring the security state of a set of external computing and network devices (called nodes) and reporting anomalies in that state. The CHG_COL.1(EX) component specifies requirements for the collection of security state from the monitored nodes. The CHG_REP.1(EX) component specifies requirements for the reporting of anomalous changes in that state. The CHG_ASM.1(EX) component specifies requirements for the assessment of changes to the collected state. This class was modeled on the CC FAU (Security Audit) class. These SFRs depend on FPT_STM.1 when time stamping collected security state and when generating reports.

The requirement FAU_GEN.1(EX) is a modification of the FAU_GEN.1 requirement from the CC. This explicitly stated requirement is used in place of the original CC requirement because the TOE always performs auditing, thus making it impossible for the TOE to generate audit records when the audit mechanism is started and stopped.

8.7 TOE Summary Specification Rationale

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

This Section in conjunction with Section 6, the TOE Summary Specification, provides evidence that the security functions are suitable to meet the TOE security requirements. The collection of security functions work together to

provide all of the security requirements. The security functions described in the TOE summary specification are all necessary for the required security functionality in the TSF. **Table 5 Security Functions vs. Requirements Mapping** demonstrates the relationship between security requirements and security functions.

	Change audit assessment (EX)	**Security audit**	**User data protection**	**Identification and authentication**	**Security management**	**Protection of the TSF**
CHG_COL.1(EX)	X					
CHG_ASM.1(EX)	X					
CHG_REP.1(EX)	X					
FAU_GEN.1(EX)		X				
FAU_SAR.1		X				
FAU_SAR.3		X				
FDP_ACC.2			X			
FDP_ACF.1			X			
FIA_ATD.1				X		
FIA_UAU.2				X		
FIA_UID.2				X		
FMT_MOF.1					X	
FMT_MSA.1					X	
FMT_MSA.3					X	
FMT_MTD.1a					X	
FMT_MTD.1b					X	
FMT_MTD.1c					X	
FMT_MTD.1d					X	
FMT_SMF.1					X	
FMT_SMR.1					X	
FPT_ITT.1a						X
FPT_RVM.1						X

Table 5 Security Functions vs. Requirements Mapping

8.8 PP Claims Rationale

See Section 7, Protection Profile Claims.