CimTrak Integrity Suite 2.0.6.0 F Security Target

EAL 4

augmented ALC_FLR.2

Release Date:	June 9, 2010
Document ID:	07-1501-R-0108
Version:	1.2

Prepared By:	M. McAlister InfoGard Laboratories, Inc.
Prepared For:	CIMCOR® 8252 Virginia Street, Suite C Merrillville, IN 46410

Table of Contents

1 INTRODUCTION 5

1.1 IDENTIFICATION 5
1.2 OVERVIEW 5
1.3 ORGANIZATION 6
1.4 DOCUMENT CONVENTIONS 7
1.5 DOCUMENT TERMINOLOGY 7
1.5.1 ST Specific Terminology 7
1.5.2 Acronyms 9
1.6 COMMON CRITERIA PRODUCT TYPE 10
1.7 ARCHITECTURE OVERVIEW 11
1.8 ARCHITECTURE DESCRIPTION 11
1.8.1 Master Repository 11
1.8.1.1 PostgreSQL (TSF Data) 12
1.8.1.2 CimTrak Database (Raw User Data) 12
1.8.1.3 Cryptographic Subsystem 13
1.8.1.4 CimTrak (proprietary) Communication Protocol 13
1.8.1.5 Heartbeat Detection Subsystem 13
1.8.1.6 Authentication Handler (Subsystem) 14
1.8.1.7 Security Event Processing Subsystem 14
1.8.1.8 Event Notification Subsystem 14
1.8.1.9 Client Messaging Subsystem 15
1.8.1.10 Master Repository Driver/Core Subsystem 15
1.8.2 Agent Component (Windows/Linux) 15
1.8.2.2 Agent Authentication Subsystem 17
1.8.3 CimTrak Management Console 19
1.9 PHYSICAL BOUNDARIES 21
1.9.1 Hardware Components 22
1.9.2 Software Components 23
1.9.3 Guidance Documents 24
1.10 LOGICAL BOUNDARIES 25
1.10.1 Security Audit 25
1.10.2 Identification and Authentication 26
1.10.3 Cryptographic Operations 27
1.10.4 Access Control 28
1.10.5 Change Management 29
1.10.6 Security Management 30
1.10.7 Secure Communications 31
1.10.8 Protection of TOE Functions 32
1.11 FEATURES NOT EVALUATED/EXCLUDED FROM THE CC EVALUATED CONFIGURATION 33

2 CONFORMANCE CLAIMS 34

3 SECURITY PROBLEM DEFINITION 35

3.1 ASSUMPTIONS 35
3.2 THREATS ADDRESSED BY THE TOE 35
3.3 ORGANIZATIONAL SECURITY POLICIES 36

4 SECURITY OBJECTIVES 38

4.1 SECURITY OBJECTIVES FOR THE ENVIRONMENT 39
4.2 MAPPING OF SECURITY ENVIRONMENT TO SECURITY OBJECTIVES 40
4.3 RATIONALE FOR IT SECURITY OBJECTIVES 41

4.4 RATIONALE FOR ASSUMPTION COVERAGE 45

5 EXTENDED COMPONENTS DEFINITION 46

5.1 CLASS FCM: CHANGE MANAGEMENT (EXPLICIT CLASS) 46
5.1.1 FCM_DET_EXP (Detection) 46
5.1.2 FCM_REM_EXP (Remediation) 47
5.2 CLASS FPT: PROTECTION OF THE TSF 48
5.2.1 TSF self test (FPT_TST_EXP) 48

6 SECURITY REQUIREMENTS 49

6.1 TOE SECURITY FUNCTIONAL REQUIREMENTS 51
6.1.1 Class FAU: Security Audit 51
6.1.1.1 FAU_ARP.1 – Security Alarms 51
6.1.1.6 FAU_STG.1 Protected audit trail storage 53
6.1.2 Class FCS: Cryptographic Support 54
6.1.3 Class FDP: User Data Protection 55
6.1.4 Class FIA: Identification and Authentication 57
6.1.5 Class FMT: Security Management 59
6.1.6 Class FPT: Protection of the TSF 62
6.1.7 Class FTA: TOE Access 62
6.1.8 Class FCM: Change Management (Explicit Class) 63
6.1.9 Class FPT: Protection of the TOE 65
6.2 RATIONALE FOR TOE SECURITY REQUIREMENTS 66
6.2.1 TOE Security Functional Requirements Rationale 67
6.3 RATIONALE FOR EXPLICITLY STATED SECURITY REQUIREMENTS 71
6.4 RATIONALE FOR IT SECURITY OBJECTIVES 72
6.5 RATIONALE FOR IT SECURITY REQUIREMENT DEPENDENCIES 72
6.6 RATIONALE FOR IT SECURITY REQUIREMENT DEPENDENCIES NOT SATISFIED 74
6.7 SECURITY ASSURANCE MEASURES 74
6.8 RATIONALE FOR SECURITY ASSURANCE 75
6.9 RATIONALE FOR TOE SECURITY FUNCTIONS 75

7 TOE SUMMARY SPECIFICATION 77

7.1 TOE SECURITY FUNCTIONS 77
7.1.1 Security Audit 77
7.1.2 Identification and Authentication 80
7.1.3 Cryptographic Operations 83
7.1.4 Access Control 87
7.1.5 Change Management 89
7.1.6 Security Management 93
7.1.7 Secure Communications 97
7.1.8 Protection of the TOE 98

List of Tables

Table 1: Hardware Components 23
Table 2: Software Components 24
Table 3: Summary of Mappings Between Threats and IT Security 40
Table 4: Functional Requirements 50
Table 5: Audited Events 52
Table 6: File Properties – Windows 64
Table 7: File Properties – Linux 64
Table 8: Mapping Between Security Functions and IT Security Objectives 67
Table 9: Explicitly Stated SFR Rationale 71
Table 10: Security Objectives to Assurance Requirements 72
Table 12: Security Assurance Measures 75
Table 13: TOE Security Function to SFR Mapping 76
Table 14: Audit Log detail by type 78

List of Figures

Figure 1: TOE network architecture 11
Figure 2: TOE Physical Boundaries 22

Document History

Document Version	Date	Author	Comments
1.0	02/16/10	M. McAlister, IGL	Roll to revision 1.0 – official release for FVOR
1.1	05/11/10	M. McAlister, IGL	Added FIPS 140-2 certificate number for Cimcor Cryptographic Module (FIPS 140-2 Certificate 1315)
1.2	06/09/10	T. Rodriguez/D. Wheeler CIMCOR	Updated versioning

1 Introduction

This section identifies the Security Target (ST), Target of Evaluation (TOE), conformance claims, ST organization, document conventions, and terminology. It also includes an overview of the evaluated product.

1.1 Identification

TOE Identification:	CimTrak® Integrity Suite Version 2.0.6.0 F
ST Identification:	CIMCOR® CimTrak® Integrity Suite 2.0.6.0 F Security Target EAL 4 augmented ALC_FLR.2
ST Version:	1.2
ST Publish Date:	June 9, 2010
ST Author:	Mike McAlister (InfoGard)
PP Identification:	Not Applicable

1.2 Overview

The CIMCOR® CimTrak® Integrity Suite application provides a flexible file-based security solution that allows Administrators (Administrator, Standard User roles) to protect selected files from unauthorized changes from a centralized location within the network. CimTrak immediately identifies the change, determines if it is authorized, and optionally institutes corrective action based on application configuration. Since CimTrak maintains a master set of protected files, unauthorized changes can immediately be reversed to mitigate malicious activity or human error. The CimTrak Integrity Suite combines CimTrak for Network Devices and CimTrak for Servers into an integrated application suite.

CimTrak, also referred to in this document as the Target of Evaluation (TOE), presents a multifaceted approach to protecting key network resources and provides comprehensive change control tracking. The application suite consists of 3 major elements: CimTrak Master Repository software acting as a central application server, CimTrak Agent software which is installed on monitored servers within the network, and CimTrak management console software.

The CimTrak Master Repository component maintains a centralized store of protected files and change history within a centralized server. This provides an isolated, encrypted copy of critical files that allows for restoration in the event of unauthorized change, and provides a basis for identifying changes made to protected files within the network. The application also supports a rollback capability which allows previous versions of a protected file to be restored at a later date. The TOE maintains 10 generations of file baselines by default.

Deployment of the TOE includes the installation of a CimTrak Agent component on protected resources within the Operational Environment. The Agent provides real-time or poll based monitoring of protected files and identifies changes made to protected files. When a change is

detected, the Agent communicates with the CimTrak Repository to report change status and/or transfer the master file (authoritative copy) from the Master Repository to the Agent server/Network Host server to overwrite unauthorized changes. The Agent utilizes CimTrak configuration data to determine if the change is allowed based on [Administrator] policy settings for the subject file. The Agent can then institute one of the following actions on the change: Allow the change and log the event, Update the master file baseline stored within the Master Repository, Disallow the change and immediately overwrite the change with the master file copy from the Master Repository, or Prompt the authorized user to either allow or disallow the file change attempt. Communication between the Agent components and the Master Repository is secured using FIPS 140-2 Level 2 validated cryptography using a proprietary CIMCOR communications protocol.

The CimTrak software solution includes a Management Console which features a Graphic User Interface (GUI) that allows Administrators (Administrator, Standard User roles) to manage/configure the application from a separate Administrator management workstation within the network. The management console supports the selection of files on Agent servers/Network Host servers to protect or "lock", configure action to take in the event a change is detected and access a series of reports that detail changes made based on a series of saved baselines stored in the Master Repository. This capability can be used to superimpose changes over the stored baselines to immediately identify what aspects of the "locked" file were changed. In addition, the application logs the identity of the user making the change, when the change was made, and the history of previous changes made to the file. The Management Console communicates with the Master Repository over a cryptographically secured session using FIPS 140-2 Level 2 validated cryptography over a proprietary communication protocol.

1.3 Organization

- **Security Target Introduction (Section 1)** – Provides identification of the TOE and ST, an overview of the TOE, an overview of the content of the ST, document conventions, and relevant terminology. The introduction also provides a description of the TOE security functions as well as the physical and logical boundaries for the TOE, the hardware and software that make up the TOE, and the physical and logical boundaries of the TOE.
- **Conformance Claims (Section 2)** – Provides applicable Common Criteria (CC) conformance claims, Product Profile (PP) conformance claims and Assurance Package conformance claims.
- **Security Problem Definition (Section 3)** – Describes the threats, organizational security policies, and assumptions pertaining to the TOE and the TOE environment.
- **Security Objectives (Section 4)** – Identifies the security objectives for the TOE and its supporting environment as well as a rationale that objectives are sufficient to counter the threats identified for the TOE.
- **Extended Components Definition (Section 5)** – Presents components needed for the ST but not present in Part II or Part III of the Common Criteria Standard.

- **Security Requirements (Section 6)** – Presents the Security Functional Requirements (SFRs) met by the TOE and the security functional requirements rationale. In addition this section presents Security Assurance Requirements (SARs) met by the TOE as well as the assurance requirements rationale. Provides pointers to all other rationale sections, to include the rationale for the selection of IT security objectives, requirements, and the TOE summary specifications as to their consistency, completeness, and suitability
- **Summary Specification (Section 7)** – Describes the security functions provided by the TOE that satisfy the security functional requirements, provides the rationale for the security functions. It also describes the security assurance measures for the TOE as well as the rationales for the assurance measures.

1.4 Document Conventions

The CC defines four operations on security functional and assurance requirements. The conventions below define the conventions used in this ST to identify these operations. When NIAP interpretations are included in requirements, the changes from the interpretations are displayed as refinements.

Assignment:	**indicated with bold text**
Selection:	indicated with underlined text
Refinement:	***additions indicated with bold text and italics***
	deletions indicated with strike-through ~~bold text and italics~~
Iteration:	indicated with typical CC requirement naming followed by a lower case letter for each iteration (e.g., FMT_MSA.1a)

The explicitly stated requirements claimed in this ST are denoted by the "_EXP" extension in the unique short name for the explicit security requirement.

1.5 Document Terminology

Please refer to CC v3.1 Part 1 Section 4 for definitions of commonly used CC terms.

1.5.1 ST Specific Terminology

[Administrators]	Refers in a general sense to a CimTrak application user and therefore a user holding at least one of three available roles: Administrator, Standard user, Auditor.
Administrator	Refers to the Administrator role specifically, as contrasted with the [Administrator(s)] designation which refers to Administrators in a general sense as users of the application.

Agent	Refers to both Filesystem Agent and Network Agent components installed as part of the CimTrak application. Where "Filesystem" or "Network" does not precede an agent description, it denotes that the content relates to both Agent types.
Agent Server	Refers to the server platform upon which a Filesystem Agent is installed.
Authoritative Copy	Refers to a saved copy of "locked" User data stored in the Master Repository for the purpose of restoring files to the last known approved state.
Client	As referenced in the Management Console subsystem, Client Core subsystem, the client in this instance refers to the management console machine.
External Entities	Any entity (human or IT) outside the TOE that interacts (or may interact) with the TOE.
Intrusion	Refers to an event in which there was an unexpected or unauthorized change to a file/object group, monitored by the CimTrak application.
Filesystem Agent	Refers to the component of the CimTrak application which is installed on servers within the network to allow the CimTrak application to monitor selected (locked) files, implement corrective action based on detected changes and collect local audit data on behalf of the Master Repository where audit data is stored.
Heartbeat	Refers to an acknowledgement message sent by the CimTrak Agent to the CimTrak Master Repository at established intervals to assure that the Agent is present and operational.
Locked Files	Refers to files that are protected on Agent server/Network Host server machines by the CimTrak application. This also connotes that changes to these files are detected by the TOE and that remediation measures, including replacement with an authoritative copy, may be taken as configured by the Administrator.
Monitoring Parameters	Refers to the CimTrak feature which allows the monitoring of CPU, Memory and Disk usage of Agent Machines and the triggering of an Alarm if a configured threshold is reached.
Master Repository	Refers to the main CimTrak application which includes server functionality used to communicate with Agent components, security management components for application configuration and databases used for storage of TSF and User Data.
Management Console	Refers to the component of the CimTrak application that is installed on an [Administrator] Workstation to provide access to the Master Repository for the purpose of configuring the

	application and reviewing detected security events and remediation taken by the TOE application.
Network Agent	Refers to the component of the CimTrak application which is installed on dedicated Host Servers within the network to allow the CimTrak application to monitor selected (locked) configuration data from Network Devices, implement corrective action based on detected changes and collect local audit data on behalf of the Master Repository where audit data is stored.
Network Agent Server	Refers to the dedicated Network Host platform upon which the Network Agent is installed.
"Private Key" feature	The "Private Key" feature allows an Administrator to apply a secondary encryption key to specific Agents or Database Object, thus requiring a passphrase be entered prior to granted access to the Agent data or object. This secures data from other [Administrators] who do not know the passphrase required to view or access the data. This is either applied at Installation time (Agent based) or during Management Console sessions (Object based).
Polling monitor	Refers to the detection mechanism employed by CimTrak network agents to detect changes made to network devices.
User Data	Refers to data stored on Agent server machines within the Operational Environment which is protected by the TOE's security functionality and, when so configured, is stored as an authoritative copy within the TOE Master Repository.

1.5.2 Acronyms

3DES	Triple Data Encryption Standard
AES	Advanced Encryption Standard
ASA	(Cisco) Adaptive Security Appliance
CC	Common Criteria
DLL	Dynamic Link Library
FIPS	Federal Information Processing Standard
FOS	(Cisco) Firewall Operating System
GUI	Graphic User Interface
HMAC	Hashed Message Authenticated SHA1 hash
ICANN	Internet Corporation for Assigned Names and Numbers
IOS	(Cisco) Internetwork Operating System
JUNOS	Juniper Operating System

NDIM	Network Device Interaction Module
ODBC	Open DataBase Connectivity
RNG	Random Number Generator
SQL	Structured Query Language
TFTP	Trivial File Transfer Protocol
TLS	Transport Layer Security
TOE	Target of Evaluation
TSC	TOE Scope of Control
TSF	TOE Security Functionality
VB	Visual Basic
VOIP	Voice Over Internet Protocol

1.6 Common Criteria Product type

The TOE is a network appliance classified as a **Sensitive Data Protection** application for Common Criteria purposes. The TOE is made up of software components.

1.7 Architecture Overview

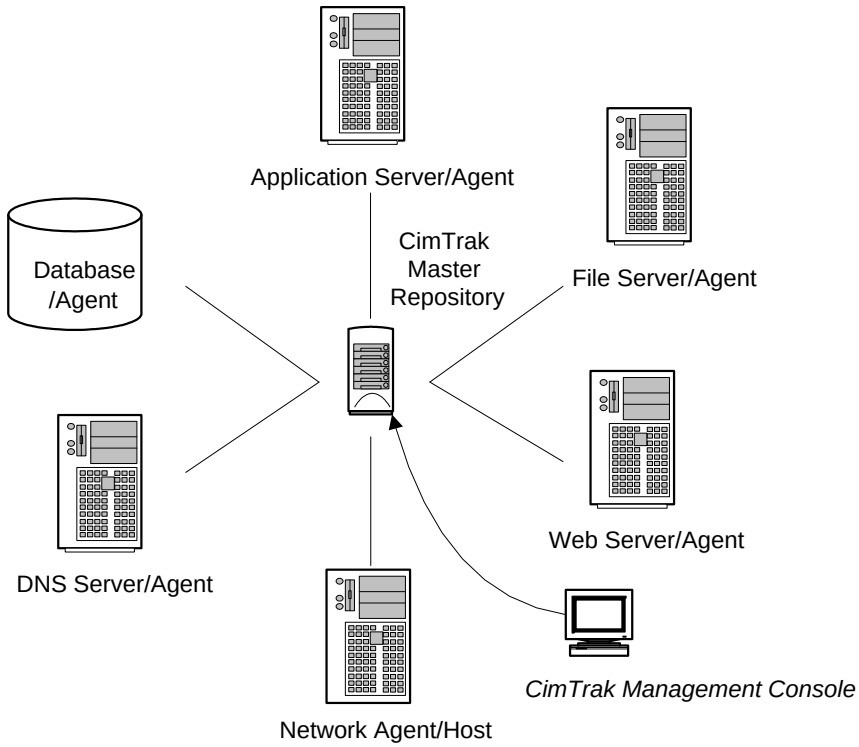


Figure 1: TOE network architecture

1.8 Architecture Description

The CimTrak application architecture is divided into the following sections in this ST.

1.8.1 Master Repository

The Master Repository is a centralized server/storage machine which hosts the Master Repository application and provides storage resources for: master copies, baseline user data, and configuration data. The Master Repository is maintained in a server room environment and local access is restricted to authorized [Administrators].

The Master Repository consists of the following subsystems:

- PostgreSQL
- CimTrak Database
- Cryptographic Subsystem
- CimTrak Communication Protocol
- Heartbeat Detection Subsystem

- Authentication Handler
- Security Event Processing Subsystem
- Event Notification Subsystem
- Client Messaging Subsystem

  Driver/Core Subsystem

1.8.1.1 PostgreSQL (TSF Data)

The PostgreSQL subsystem is a CimTrak implementation of the PostgreSQL Open Source database. This subsystem provides a relational database used to store all TSF data including application configuration data, audit logs, application account data, locked filenames, file SHA1 hash and metadata from Agent servers/Network devices monitored by the CimTrak application. This database is responsible for storage of all data except for the actual copies of user data maintained in the Master Repository for file rollback and restoration functions. Additional information about the PostgreSQL database can be viewed at: http://www.postgresql.org/about/.

1.8.1.2 CimTrak Database (Raw User Data)

The CimTrak Database is a proprietary database implementation developed by Cimcor for the CimTrak product. This database is designed to allow for the storage of raw User Data in its native format without typical database format restrictions. This allows the CimTrak Master Repository to store actual copies of User Data in exactly the same format as maintained by the source within the CimTrak managed Agent servers/Network Host servers. Within the CimTrak proprietary database the following information is maintained related to the raw user data stored there:

a) The filename/configuration file ID

b) Attributes associated with the file/configuration

c) Based on the policy selected, the actual contents of the file/configuration will be stored

d) Generation ID

All information stored in the proprietary CimTrak database is encrypted and compressed. The combination of CimTrak Server ID, CimTrak Agent ID, CimTrak Object Group, Generation ID, and Filename, are used to correlate information between the PostgreSQL database, and the proprietary CimTrak proprietary database.

This CimTrak database subsystem is written is the C^{++} programming language.

1.8.1.3 Cryptographic Subsystem

The Cryptographic subsystem is a Cimcor Cryptographic Module FIPS 140-2 Level 2 validated software object (FIPS 140-2 Certificate #1315) used by the TOE for the purpose of securing communications between TOE components using TLS, encryption of User Data stored within the Master Repository, File Comparison (Hashing), and creation of digital signatures for specified data managed by the TSF. The subsystem itself is written in the "C" language with additional TSF interface code and associated .DLLs written in C^{++}. These added portions allow the CimTrak application to access the Cimcor Cryptographic Module and utilize its functionality in support of the TSF.

Symmetric cryptographic session keys are generated by the Cimcor Cryptographic Module software Random Number Generator (RNG) within the Cryptographic Subsystem using either the AES or 3DES algorithm. Asymmetric keys can also be generated by the Cimcor Cryptographic Module software RNG to support key establishment through Diffie-Hellman or RSA.

TLS based session encryption supported by the Cryptographic subsystem includes communications between the CimTrak Master Repository and deployed CimTrak Agents and CimTrak management sessions between the CimTrak management console and the CimTrak Master Repository. These sessions are conducted using TLS and either the AES or 3DES algorithm.

1.8.1.4 CimTrak (proprietary) Communication Protocol

The CimTrak proprietary communication protocol is a TCP/IP based protocol operating over an IPv4 or IPv6 network protocol connection. This subsystem is written in the C^{++} programming language. The protocol interacts with the underlying operating system, Network Interface Card (NIC) and resident cryptographic subsystem to orchestrate communications between Agent machines/Network host machines, management console and the Master Repository. These sessions are conducted over TLS as supported by the Cimcor Cryptographic Module within the Cryptographic subsystem as described above.

1.8.1.5 Heartbeat Detection Subsystem

The Heartbeat Detection Subsystem is provided within the application to monitor deployed CimTrak Agents with the deployed network. The CimTrak application maintains a heartbeat with Agents to verify their online status. By default, heartbeat is verified at 30 second intervals; however, this value may be configured by the Administrator role. In the event heartbeat verification goes unanswered, a security event is processed, the configured corrective action is executed and an email is sent to the configured address for the Administrator role based on application settings.

1.8.1.6 Authentication Handler (Subsystem)

The Authentication Handler is a software subsystem written in C++ which interfaces to the PostgreSQL database for accessing authentication credentials during the login authentication process. When credentials are passed to the Authentication Handler from a deployed Agent or an [Administrator] requesting a management session, the Authentication Subsystem passes the credentials to the PostgreSQL database where they are compared against stored hashed password values. Upon successful validation of credentials, the Agent or [Administrator] session is allowed to proceed.

1.8.1.7 Security Event Processing Subsystem

The Security Event Processing Subsystem is a C++ subsystem which orchestrates the transfer of data to and from CimTrak Agents in the deployed network environment and the Master Repository. When a file change is detected by a CimTrak Agent and the settings dictate the new baseline should be saved to the CimTrak database within the Master Repository, the Security Event Processing Subsystem receives the changed file data from Agents and routes this data to the CimTrak database. This subsystem also participates in CimTrak Agent startup processes where file metadata*, Agent initialization instructions and data change/remediation policies are passed from the Master Repository machine to the Agent component for local enforcement of CimTrak policies. Agents do not retain this information locally upon shutdown; therefore, this transfer of configuration/policy data is executed following authentication during every Agent server/Network Host server startup cycle.

*metadata refers to information on how the object group should be protected. The metadata contains all of the object group settings as specified on the object group screen, directories and filenames to protect, and their attributes. These attributes are listed under File Object Group Properties in Section 6.1.5.3.

1.8.1.8 Event Notification Subsystem

The Event Notification Subsystem is a C++ subsystem within the CimTrak application which orchestrates the transfer of events and audit data to internal data stores within the PostgreSQL Database. In addition, this information may also be routed to external resources via SNMP (for administrative messaging), SMTP (email notifications to admin), the WebTrends Extended Logging Format (for use with the WebTrends firewall reporting tool), and Syslog for external storage of audit log records. This subsystem works in conjunction with the Security Event Processing Subsystem, Authentication subsystem and Client Message Processing subsystem to assure that security events related to those subsystem functions are detected and reported. The [Administrator] is notified of security events based on configured email notification and audit logs which produce a detailed audit trail for later review.

1.8.1.9 Client Messaging Subsystem

The Client Message Processing Subsystem is a C++ based object subsystem which is responsible for the passing of data objects from the CimTrak Application, installed on the Master Repository machine to the CimTrak Management Console software running on the [Administrator] Workstation. The CimTrak GUI is loaded locally on the [Administrator] Workstation using the CimTrak Management Console software and data used to populate the GUI pages is requested from the Master Repository via a secure session. Upon verification that the access privileges allow access to the requested data objects, the data objects are passed to the Management Console via the Client Message Processing Subsystem and are rendered within the applicable GUI page to the CimTrak [Administrator].

1.8.1.10 Master Repository Driver/Core Subsystem

The Master Repository Core subsystem is a centralized processing resource that serves as a Master Control object that mediates incoming connection requests and routes them to the Authentication Handler subsystem. This core subsystem executes initial loading of setup values during Master Repository startup. General support activities are also handled by this subsystem such as log cleanup of old entries, CimTrak Master Repository email functions and checking for inactive management console sessions. In addition, this subsystem interfaces with the CimTrak communication protocol to accept message requests and route them to the applicable subsystem to process the requested action. The Master Repository Core subsystem is written in C++.

1.8.2 Agent Component (Windows/Linux)

The Agent application of the CimTrak TOE is installed on each Server for which application functionality is desired. The Network Agent component resides on a dedicated Host machine serving as a platform for the Agent in supporting network devices such as routers, switches or firewall devices. The Agent is manually installed using an installation CD on each Agent server/Network Host server. The Agent component implements the protection or "locking" of selected files on the Agent server/Network Host server. This component detects when changes occur to locked files and immediately executes the configured corrective action. The Agent interacts with the Master Repository to download local policy enforcement settings during each Agent server/Network Host server startup process. This approach is required during each startup cycle as no TSF data is stored within Agent server/Network Host server hard drive resources after Agent shutdown.

Audit data is not stored on local Agent server/Network Host server machine hard drive or cache resources. It is maintained in volatile memory until it is transferred securely in its raw form to the Master Repository for inclusion in application audit logs within the PostgreSQL database. Intrusion type logs are immediately transferred to the Master Repository whereas, all other Agent server/Network Host server events are batched at the Agent and are sent to the Master

Repository every 2 minutes by default. Further information regarding Agent audit support is described in Section 1.10.1, Security Audit security function.

File data transfers occur between the Agent and Master Repository when new baselines are established and when file restoration is required due to an unauthorized file change. Audit event info is also transferred between the Agent & Master Repository to communicate action taken by the Agent, applicable security notification and actions taken to the Master Repository, where the audit data is centralized for [Administrator] review. Filesystem Agent transfers are made over TLS encrypted sessions. Network Agents transfers uses secure SSHv2 protocol sessions.

CimTrak Agent components are identical among supported server types except for the following:

- Network Agents include an Network Device Interaction Module (NDIM)
- Filesystem Agents include a Disk Filter Driver module

Network Agents support network devices that support remote configuration and invokes the Network Device Interaction Module (NDIM) to coordinate communication to these devices. Network Agents require dedicated Host Server hardware to provide a platform for the Agent to execute on; however, a single Agent host machine can support multiple network components. As with the non-Network Agents installed directly on server machines, no TSF data such as policy settings or audit logs is stored local to the Host Server running the Network Agent.

Filesystem Agent include a Disk Filter Driver module for interacting with the disk for monitoring file change events on the installed platform; since Network Agents monitor and remediate configuration changes remotely, this subsystem is not required.

Windows based file servers, Linux based file servers and Network devices all have different CimTrak Agents and all types are included in the same Agent installation package for the Common Criteria Evaluated configuration. The difference between these Agent components is related to the Operating System interface and the file characteristics monitored for change detection.

The Agent Component consists of the following subsystems:

- Agent Core Subsystem
- Agent Authentication Subsystem
- CimTrak Communication Protocol
- Agent Security Event Logic
- Cryptographic Subsystem
- Agent Filter Driver subsystem (Filesystem Agents only)
- Network Device Interaction Module (Network Agents only)

1.8.2.1 Agent Core Subsystem

The Agent Core Subsystem provides the base Agent software infrastructure for local implementation of CimTrak security features. The core subsystem, written in C++, communicates with the Master Repository to download local file names, policy enforcement instructions and maintains a heartbeat monitoring arrangement with the Master Repository to verify continued operations. The Agent File Monitor and the Agent Core Subsystem interact with the Agent server/Network Host server file system (OS) and with the Security Event logic subsystem to monitor the status of locked objects and facilitate action when required. The Agent Core subsystem runs as a service on the Agent server/Network Host server.

For Filesystem Agents, the Agent Core Subsystem (service) monitors locked file status using a privileged user status within the OS.

For Network Agents, the Agent Core (service) works with the Network Device Interaction Module to monitor configuration status on focused network devices.

Also included within this subsystem are C++ based DLL files which facilitate interaction with the underlying operating system and the cryptographic subsystem as part of the Agent software component.

1.8.2.2 Agent Authentication Subsystem

The Agent Authentication Subsystem is a C++ based subsystem which manages identification and authentication functions, on behalf of the Agent, to the Master Repository. During establishment of the Agent, a username and password is established for the purpose of authenticating the Agent to the Master Repository prior to allowing the download of TSF data (file names, policies etc.). The password created for the Agent is maintained internally at the software level and is not accessible by any human user by design. The password is stored in the form of a one-way hash within the Agent Authentication subsystem.

A Diffie-Hellman or RSA key exchange is conducted between the Agent Authentication subsystem and the Master Repository and the TLS session is established. The username/password pair is passed from the Agent to the Master Repository and is validated by the Master Repository. Following successful authentication, configuration data is passed securely over the TLS encrypted channel to the CimTrak Agent for implementation and the selected (locked) files are immediately monitored for changes.

1.8.2.3 CimTrak (proprietary) Communication protocol

The CimTrak Communication protocol is fully described in Section 1.8.1.4 as it is also used within the Master Repository. CimTrak Protocol implementation within the Agent Component facilitates the communication between the Agent Subsystem and the Master Repository for the transfer of TSF and User data.

1.8.2.4 Agent Security Event Logic

The Agent Security Event Logic subsystem is a C^{++} based subsystem which interacts with the file system monitor to trigger security events upon detecting a change attempt to a locked file. When Filesystem Agents detect a change to a locked file, this subsystem implements specific actions configured within CimTrak associated with that particular file. Upon identifying an unauthorized file change requiring immediate restoration, the Agent Security Event Logic subsystem returns a command to the underlying file system to replace the unauthorized file changes with files obtained from the Master Repository.

For Network Agents, this subsystem works with a polling monitor within the Network Device Interaction Module (NDIM) to identify network device configuration changes to a locked attribute and trigger a security event. Based on the configured Corrective Action, the Agent Security Event logic may orchestrate the replacement of the changed configuration attributes with baseline copies (authoritative copy) obtained from the Master Repository.

For both Agent types, this subsystem passes security notifications to the Master Repository, which creates the associated audit log entries within the PostgreSQL database and triggers a notification email to the configured email address when applicable.

1.8.2.5 Cryptographic Subsystem

The Cryptographic Subsystem within the Agent Component is identical to the Cryptographic Subsystem described as part of the Master Repository in Section 1.8.1.3.

All communication between the CimTrak Agent and the Master Repository is conducted over TLS encrypted sessions. As part of the TLS session, message integrity is verified to establish that no changes have occurred in transit during these sessions by using a hashed message authenticated SHA1 hash (HMAC).

The Cryptographic Subsystem within the Agent Component provides Cryptographic support for securing communications between the Agent Component and the Master Repository which includes Session Key Generation, Session Encryption/Decryption and destruction of cryptographic keys used for these sessions. This subsystem implements a Cimcor Cryptographic Module which provides support for the establishment of TLS sessions between the Agent Machine and Master Repository.

1.8.2.6 Agent Filter Driver Subsystem (Windows Filesystem Agents)

The Agent Filter Driver Subsystem pertains only to Windows based Filesystem Agents and allows the TOE [Administrator] to monitor file change events using an additional CimTrak component as opposed to the standard CimTrak approach of monitoring via the underlying OS. This represents an option that provides more detailed analysis of Agent file changes monitored by CimTrak. This alternative file monitor mechanism is loaded dynamically by the Agent Security Event subsystem and allows for the monitoring of additional details about particular file changes such as process name, process ID, thread ID and user. The driver then processes all File IO, it determines the path and filename for the event, then if the filename is a file or directory

being watched, it then determines whether this is an event that needs reporting. When events are detected the driver buffers them until the Agent Security Event subsystem pulls them out to process them.

CimTrak can use this information to determine network connections by user and where those connections are mounted for forensic investigation purposes. This subsystem provides a driver mechanism, which read/write processes on the Agent machine must pass through, prior to the request being passed to the next level driver. The use of the Agent Filter driver mechanism is more intrusive than just hooking to a windows event (as the standard CimTrak approach), but provides more details about file change and user activity when desired by the [Administrator].

1.8.2.7 Network Device Interaction Module (Network Agents)

The Network Device Interaction Module applies only to CimTrak Network Agent installations. The Network Device Interaction Module coordinates the communication process for the CimTrak Agent installed on the dedicated Host Server and the supported network device, typically a router or firewall. Communication between the CimTrak Agent and the network device is conducted over the SSHv2 protocol. This subsystem includes LibSSH "C" libraries used to establish SSHv2 connections with Network devices. Trivial File Transfer Protocol (TFTP) is used to transfer configuration files to Cisco network devices. Other devices, those developed by Juniper, support configuration file remediation through SSHv2.

Connection to these network devices requires a script that contains connection string information. During the Network Agent installation process, these scripts, as part of the Network Device Interaction Module, are used to determine which network devices are available and are supported by CimTrak. Following installation, these scripts are used to communicate with the respective Network devices to read/write configuration data monitored by the CimTrak Network Agent. The NDIM module includes configuration scripts for Juniper devices running JUNOS, Cisco® devices running IOS/FOS/ASA for communication with Windows and Linux based network device deployments. CimTrak Network Agents supports Cisco and Juniper network devices running the following Operating Systems. Devices must support SSHv2:

- Juniper- JunOS version 6.0 and above
- Cisco- IOS version 11.2 and above
- Cisco- FOS version 6.0 and above
- Cisco-ASA version 7.0 and above

As with the Filesystem Agent type, the Network Device Interaction Module interfaces with the Agent Core Subsystem, Agent Authentication Subsystem, CimTrak Communication Protocol Subsystem, Agent Security Event Logic, and the Cryptographic Subsystem

1.8.3 CimTrak Management Console

The CimTrak Management Console is a software subsystem loaded on the [Administrator] Workstation for GUI based access to the CimTrak application hosted on the Master Repository

machine. Access to the CimTrak application is only allowed through use of this management console software. This Management Console renders the GUI pages local to the Management Console machine and only passes object access requests & object data across the secure channel established with the Master Repository. Upon receiving the requested objects, this subsystem populates the applicable GUI form and renders the information to the [Administrator]. The Management Console only retains registry & preferred language settings after a session is closed, therefore, all additional TSF or User Data must be downloaded during each session from the Master Repository following [Administrator] authentication. The CimTrak Management Console component consists of the following subsystems:

- Client Core Subsystem
- Authentication Subsystem
- CimTrak Communication Protocol
- Cryptographic Subsystem

1.8.3.1 Client Core Subsystem

The Client Core Subsystem is a software subsystem written in the (visual basic) VB.net object-oriented programming language which provides the security management application for the CimTrak TOE. This subsystem includes the GUI pages making up the set of management interfaces and populates the GUI with data objects imported from the Master Repository.

The Client Core subsystem communicates through a C^{++} Active X software component to facilitate object requests/transfers to and from the Master Repository. Access to TSF or User Data is controlled by object access control mechanisms provided by the authentication subsystem within the Master Repository. When a data object is requested by the Client Core Subsystem (as part of a report or GUI form page), the Authentication Subsystem within the Master Repository application verifies that the [Administrator] is allowed access to that object and then the object is passed to the Client Core Subsystem for rendering on the GUI page. In the event the object is not authorized for access by the requesting user, the field will appear grayed out or the GUI page will not be shown and an "Access Denied" page will be rendered on the CimTrak Management Console.

1.8.3.2 Authentication Subsystem

The Authentication Subsystem with the CimTrak Management Console software provides the mechanism for passing authentication credentials from the Management Console machine to the Master Repository for validation. This subsystem is identical to the Authentication Subsystem contained within the Agent Component except that it supports authentication of [Administrators] requesting appliance security management functions vs. supporting authentication on behalf of the Agent component. The (Agent) Authentication Subsystem description may be reviewed in Section 1.8.2.2.

1.8.3.3 CimTrak (proprietary) Communication Protocol

The CimTrak Communication protocol is fully described in Section 1.8.1.4 as part of the Master Repository architecture description. Implementation within the CimTrak Management Console facilitates the communications between the CimTrak Management Console and the Master Repository for use in management of the TOE application.

1.8.3.4 Cryptographic Subsystem

The Cryptographic Subsystem within the Agent Component is identical to the Cryptographic Subsystem described as part of the Master Repository in Section 1.8.1.3.

All communication between the CimTrak Management Console and the Master Repository is conducted over symmetric key encrypted sessions over TLS. Message integrity is verified using the SHA1 hashing algorithm to establish that no changes have occurred in transit during these sessions by using a hashed message authentication code - SHA1 (HMAC).

The Cryptographic Subsystem within the CimTrak Management Console Component provides Cryptographic support for securing communications between the Management Console and the Master Repository, which includes Session Key Generation, Session Encryption and destruction of cryptographic keys used for these sessions.

1.9 Physical Boundaries

This section lists the hardware and software components of the product and denotes which are in the TOE and which are in the environment.

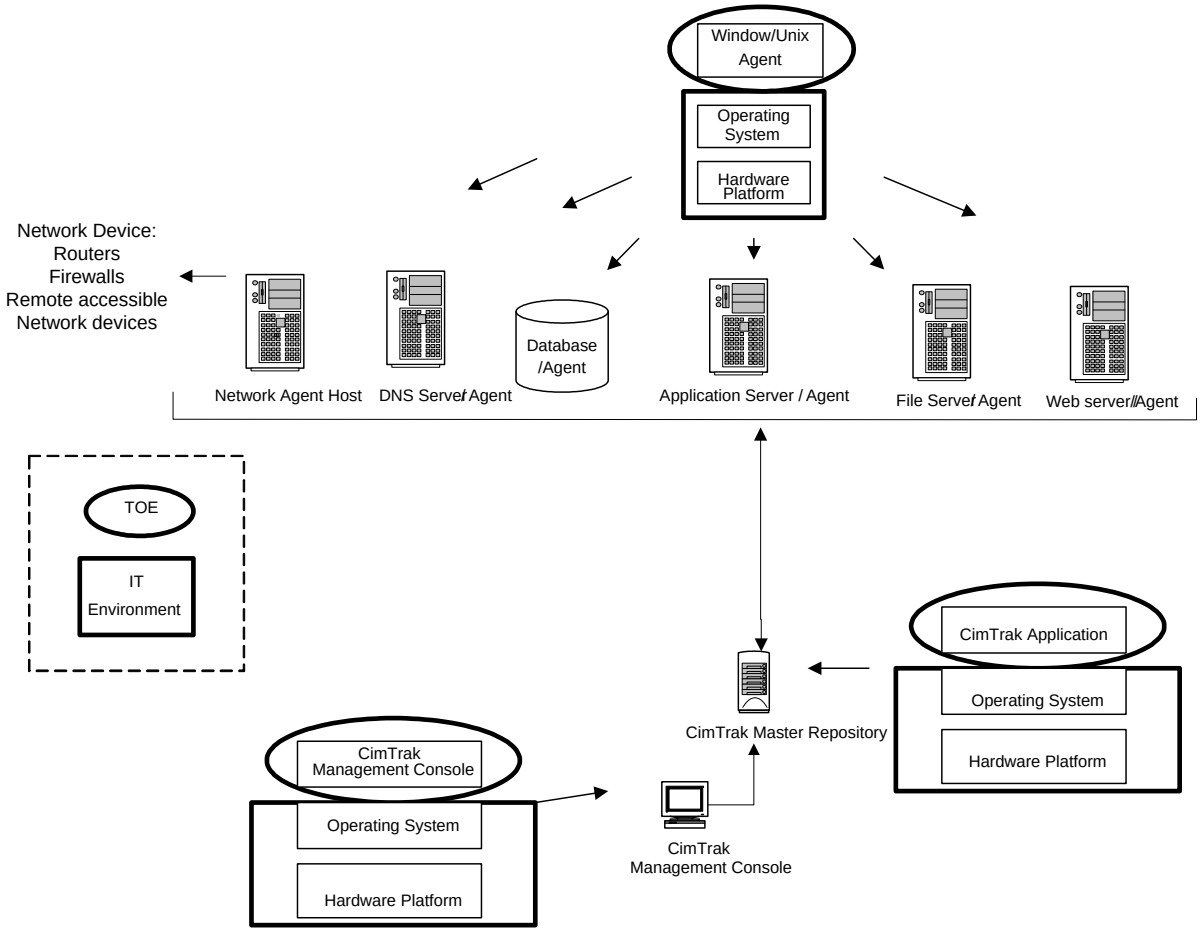


Figure 2: TOE Physical Boundaries

1.9.1 Hardware Components

This table identifies hardware components and indicates whether or not each component is in the TOE.

TOE or Environment	Component	Description
Environment	Server Hardware – Master Repository	Minimum: Pentium III (Pentium IV, 2.6 GHz or better recommended) 1Gig RAM 220MB of available hard drive space

Environment	CimTrak Management Console	Minimum: Pentium 500 MHz (Pentium II or better recommended) 512MB RAM 200MB of available hard drive space
Environment	CimTrak Filesystem Agent	Minimum: Pentium III processor 512MB RAM 200MB of available hard drive space
Environment	CimTrak Network Device Agent	Minimum: Pentium III processor 512MB RAM 150MB of available hard drive space
Environment	Firewall/ Router/Switch (Network devices with remote configuration capabilities)	Network Architecture item(s) As required based on network topology Provided scripts support: Cisco and Juniper network devices running the following Operating Systems (OS): Juniper- JunOS version 6.0 and above* Cisco- IOS version 11.2 and above* Cisco- FOS version 6.0 and above* Cisco-ASA version 7.0 and above* *must support SSHv2

Table 1: Hardware Components

1.9.2 Software Components

This table identifies software components and indicates whether or not each component is in the TOE.

TOE or Environment	Component	Description
TOE	CimTrak Application Release 2.0.6.0 F	CimTrak Application loaded on the Master Repository

Environment	Master Repository Operating System	Windows 2000 SP4, XP SP3, 2003, Vista, or 2008
TOE	CimTrak Agent Release 2.0.6.0 F	CimTrak Windows/Linux/Network Agent installed on network machines or network agent hosts
Environment	Agent server Operating System	Windows 2000 SP4, XP SP3, 2003, Vista, or 2008 Linux Kernel 2.6 (Ubuntu, Redhat, CentOS, SuSE)
TOE	CimTrak Management Console Release 2.0.6.0 F	CimTrak Management Console component
Environment	Management Console Operating System	Windows 2000 SP4, XP SP3, 2003, Vista, or 2008
Environment	DNS Application(s)	Network applications
Environment	Database Server Application(s)	Network applications
Environment	Web Server Application(s)	Network applications
Environment	File Server Application(s)	Network applications

Table 2: Software Components

1.9.3 Guidance Documents

The following guidance documents are provided with the TOE upon delivery in accordance with EAL 4 requirements:

1) Cimcor® CimTrak Integrity Suite Common Criteria Supplement EAL4
Document #: 2010-03-01-2060F

2) Cimcor® CimTrak User Guide Version 2.0.6.0

CimTrak Server / Repository
CimTrak File System Agent
CimTrak Management Console
CimTrak Network Device Agent

3) Cimcor® CimTrak Installation Guide Version 2.0.6.0

CimTrak Server / Repository
CimTrak File System Agent
CimTrak Management Console
CimTrak Network Device Agent

All documentation delivered with the product is germane to and within the scope of the TOE.

1.10 Logical Boundaries

This section contains the product features and denotes which are within the logical boundaries of the TOE. The following Security Functions are supported by the CimTrak TOE:

- Security Audit
- Identification and Authentication
- Cryptographic Operations
- Access Control
- Change Management
- Security Management
- Secure Communications
- Protection of the TOE

1.10.1 Security Audit

The Security Audit security function within the CimTrak application generates audit records for security related events including: application configuration, file change detection & remediation, and [Administrator] access to TSF/User Data.

The Security Audit security function is supported by the Security Event Logic within the CimTrak Agent Component which communicates Agent events to the Master Repository for inclusion in audit records. Within the Master Repository, the Event Notification Subsystem and Security Event Processing subsystems participate in generating Events for audit logs in conjunction with audit storage facilities within the PostgreSQL database.

Security Audit events generated from the Agent Components include the success/failure of Agent login/initialization to the Master Repository, synchronization of file data between the Agent and the Master Repository, success/failure of file locking operations and file change/remediation action taken upon detection of changes to locked files. Changes made to locked files generate a series of event logs entries to detail the event. Once a change is made, a "File modified" log entry is made detailing the date/time of the event, the file path of the changed file on the Agent server/Network device, the identity of the Agent detecting the change and the remediation action taken. Also, in the event files are saved to locations that are locked, (i.e., a worm or Trojan program) an entry is made that indicates the added file was removed and saved to the Master Repository for later review. Audit events are not stored on the Agent server/Network Host server machine storage resources. They are queued within volatile memory. Intrusion related logs are immediately uploaded to the Master Repository and all other logs types and are passed to the Master Repository at two minute intervals.

The Master Repository, in conjunction with the Management Console, generates security audit records for application configuration actions taken by [Administrators] such as installation processes, account management configuration, management of Agent server/Network Host server accounts & object groups, cryptographic algorithm/key size selection, watch list creation defining file objects to be protected and configuration of watch properties (action to be taken upon detection of a Agent file change). This information is maintained under the Event Log tab when the Master Repository machine is accessed for viewing of audit data.

Audit records may only be accessed by authorized [Administrators] through properly authenticated CimTrak Management Console sessions and may not be modified by any user. Agent servers/Network Host servers do not feature a user interface and therefore cannot access audit records maintained in the Master Repository.

Disk storage space for the Master Repository is monitored to assure storage resources are not depleted during operation. A storage threshold is set and upon reaching the specified storage amount used (%), an email notification is sent to the configured [Administrator] email address advising of limited remaining audit storage remaining. This protects audit records in that these are stored upon the same disk resources, therefore, audit records are protected from full storage resources through this disk storage level monitoring function.

Audit logs may be processed in parallel to a syslog server within the Operational Environment for external storage of CimTrak audit records.

1.10.2 Identification and Authentication

The Identification and Authentication security function provides the method by which the CimTrak TOE assures that entities communicating with the Master Repository are identified and authenticated prior to being granted access to TSF resources. This includes authentication prior to access of User Data stored within the TOE's Master Repository component. ID & Authentication polices are also enforced against External entities attempting to communicate with the Master Repository (i.e., Agents) as well as human users accessing the CimTrak TOE via the CimTrak Management Console ([Administrators]).

Agents maintain User Accounts within the Master Repository as much as a human user does. The Agent account includes a Username/Password combination that must be provided following the initial key exchange and establishment of the secure session. The password for the Agent is generated during Agent installation using the Random Number Generation (RNG) within the Cryptographic Subsystem. The manner in which the password is generated and stored assures that it is only known & accessible to the Agent and Master Repository applications and is not accessible to a human user. When an Agent server/Network Host server is booted and the Agent service is started, it attempts to contact the Master Repository. Following the Diffie-Hellman or RSA based key exchange and establishment of a secure TLS session, the Agent presents its Username and Password to the Master Repository for validation. The Authentication Subsystem takes these credentials, hashes them using the Cryptographic Subsystem and compares the hashed value against the hashed credentials in the PostgreSQL database. Once validated, the Agent downloads configuration data from the Master Repository and begins monitoring locked

files. Passwords are stored within the Master Repository: PostgreSQL Database in the form of a one way cryptographic hash using SHA1.

[Administrator] sessions are initiated from the CimTrak Management Console machine within the deployed network. The [Administrator] initiates a management session with the Master Repository by launching the Management Console application and entering the applicable username/password on the provided log-on page. Prior to validation of credentials, the Management Console only allows access to the Help screens and Language selection options within the log-on page. The Management Console application initiates a secure session using the same Diffie-Hellman exchange described above and validates authentication credentials in the same manner as noted above for CimTrak Agents. Upon successful validation of credentials, the Master Repository Client Messaging Processing subsystem passes the requested data objects to the Management Console application for rendering on the GUI pages as the validated role allows.

CimTrak supports various password enforcement measures including strong password guidelines, password expiration after a specified period of time and locking of accounts upon meeting a maximum number failed logins. The TOE supports using user entered or RNG created passwords and custom settings that allow the Administrator to require minimum number of characters, upper/lowercase/special characters, perform dictionary checks to avoid use of dictionary words for passwords, and disallow the use of previously used passwords.

The CimTrak application manages user roles by group memberships. The Common Criteria Evaluated configuration stipulates the creation of the Administrator, Standard User and Auditor groups during installation. During the login process, the TOE authenticates [Administrators] to one of three available roles for management of the application: Administrator, Standard User and Auditor. These roles are further described in Section 1.10.6 Security Management.

1.10.3 Cryptographic Operations

The Cryptographic Operations security function provides cryptographic support for securing communications, data transfer and data storage within the CimTrak TOE. The TOE utilizes a FIPS 140-2 Level 2 validated Cimcor Cryptographic Module (FIPS 140-2 Certificate #1315).

To initiate communications between the CimTrak management console/Server Agent and the Master Repository, the applicable Agent contacts the Master Repository to establish a TLS secure session. A Diffie-Hellman or RSA based key exchange is conducted using an asymmetric key size of 512, 1024 or 2048 bits. Subsequent to the successful key exchange, a private key is created using the Cimcor Cryptographic Module software RNG.

The CimTrak TOE supports encrypted sessions using SSHv2 between the Agent component installed on the Network host computer and network devices managed by CimTrak. These sessions leverage LibSSH “C” libraries within the Network Device Interaction Module that supports SSH session establishment using the SSHv2 protocol and AES or 3DES encryption algorithms.

Symmetric cryptographic keys are generated using a software random number generator using the AES cryptographic algorithm with key sizes of 128, 192 or 256 bits. Alternatively, the 3DES

algorithm may be used with a key size of 192 bits. This key is used to encrypt/decrypt data using the TLS secure communications protocol.

This same secret key is used to key the HMAC used for integrity checking of data transferred through these TLS based sessions. The HMAC used in this processes leverages the SHA1 algorithm using a block size of 512 bits.

Encryption and decryption operations of session traffic and data at rest use either the AES or 3DES symmetric key algorithms. This includes securing Management Console sessions over TLS with the Master Repository, Agent sessions with the Master Repository over TLS and encryption of User Data at rest within the CimTrak database inside the Master Repository. Key exchanges used to initiate secure TLS based sessions between Agents or the Management Console with the Master Repository is managed using the Diffie-Hellman or RSA key exchange protocol.

CimTrak also includes an option for applying a secondary layer of encryption to data on an Agent or Object basis. During Agent installation, a passphrase may be entered that results in the generation of a symmetric (private) key associated with that Agent; supporting the "Private Key" encryption feature. From that point forward, all Master Repository encryption actions for that Agent utilize that key for 1st stage encryption and then perform a secondary encryption function with a CimTrak symmetric key assigned to the Master Repository. Alternatively, this secondary ("Private Key") encryption can be enabled on an Object basis from the Management Console. This feature allows certain Agent data or Database objects to be access controlled by the applicable [Administrator], so the CimTrak [Administrator] won't necessarily have access to the entire database. Separate [Administrators] may be assigned to particular Agents or database Objects to allow for a further level of access control.

Hashing is used within the CimTrak TOE for comparison of files to determine if changes have occurred. The SHA1 algorithm is used to produce a message digest of 160 bits long which serves as cryptographic hash of User Data stored within the Master Repository.

Mechanisms within the Administrative interface of the CimTrak TOE enforce that only FIPS 140-2 Level 2 validated cryptography is used and that all options available for use meet FIPS 140-2 operational requirements.

1.10.4 Access Control

The Access Control security function enforces the CimTrak Access Control security function policy (SFP) which defines rules for access to user data managed by the TOE. Write access is effectively restricted to any files designated as "locked" by the CimTrak application.

The CimTrak application directly accesses files on machines in the Operational Environment using CimTrak Filesystem Agents installed directly on the server's file system. Network device configuration is monitored through an Agent installed on a Host server monitoring device remotely. This allows the Agent to monitor change status for selected data objects, send a copy of the file to the Master Repository, and/or overwrite the file on the Agent server/Network device with an authoritative copy imported from the Master Repository. The CimTrak TOE enforces

the CimTrak Access Control security function policy for all access to user data through the TOE application.

Access to User Data once stored in the Master Repository is limited to CimTrak [Administrators] through the Management Console and the applicable Agent from the original file system/network device. File or configuration objects are tied to their associated Agent accounts with respect to TOE operations. This assures that only the Agent associated with a particular file stored on the Master Repository can only be accessed by that Agent. For Network devices, this is managed by discrete object groups so only the single member of a specific object group can access associated configuration data stored on the Master Repository.

Access Control is supported by the authentication mechanisms described in Section 1.10.2 Identification and Authentication. The Authentication Subsystem within the Master Repository supports access to the CimTrak application and the CimTrak Database stores user data and restricts access to only properly authenticated user entities.

Access to the CimTrak Database is coordinated and performed solely by the CimTrak Master Repository. Upon a request for data from the client/agent, the Master Repository Authentication Handler validates the request for security access and that the client/agent is requested data valid for its security level. If the requested request is approved, then the Master Driver/Core subsystem retrieves and, leveraging the Cryptographic Module, decrypts the information from the CimTrak Database and returns it to the requesting TOE component.

1.10.5 Change Management

The Change Management security function provides the ability for the TOE to detect changes made to files/configuration data set as locked through the CimTrak application. Once such changes are identified, CimTrak implements the configured corrective action (remediation) and provides a comprehensive mechanism to document and track file/configuration changes for Agent servers/Network devices deployed within the network.

The Agent component of the CimTrak application is installed on selected machines within the network environment. During application configuration, a series of file objects or object groups are selected for each machine or network device and are thereby designated for monitoring. This information is downloaded by the Agent from the Master Repository during each Agent server/Network Host startup and initialization sequence.

For Filesystem Agents, the Agent Core subsystem works in conjunction with the underlying Operating System[+] to detect when a change is attempted to a file configured within CimTrak on a watch list. The CimTrak Agent Core Subsystem runs as a service on the Agent server/Network Host server operating system, supporting change detection on a "real time"* basis. The CimTrak Agent component is configured to watch and "lock" files on that particular machine. Filesystem Agents are assigned a discrete Agent per platform as the Agent is installed directly on the machine where monitoring is required.

[+]Application note: The detection of changed files on Agent installed platforms depends, in part, on Operating System components within the Operational Environment.

*"real time" refers to the ability to detect file object changes as they are made vs. the "polling" method which detects file changes at periodic time intervals.

Network Agents are installed on a Host Server machine that provides the interface between the monitored network devices and the Master Repository. Since an Agent cannot be installed directly on a router, for example, a dedicated Host Server is used to provide the platform for Agent operation. Within the Network Agent, the Agent Core subsystem, working with the Network Device Interaction Module, simultaneously polls configuration data from the Network device(s) configured to that Host. This data is hashed by the (Agent) Cryptographic Module and the value is compared against the value obtained during startup from the Master Repository. If these values don't match a change is detected and Corrective Action (remediation) is implemented by the Agent: Security Event Logic subsystem as specified below.

Once a change has been detected, the Agent: Security Event Logic subsystem immediately implements the configured Corrective Action locally for Filesystem Agents and remotely on the applicable network device for Network Agents. This remediation action includes the following options:

- Log Only option - allows the change event but logs the event to the Master Repository for reporting.
- Prompt – a prompt is displayed which allows the authorized user to either allow or disallow the change event.
- Update Baseline – allows the change and forwards the changed file to the Master Repository: CimTrak Database for storage as the new file object baseline.
- File Restore – immediately overwrites the changed file object and forwards the changed file to the Master Repository: CimTrak Database for storage as an unauthorized change.
- Custom – this setting allows the configuration of custom Corrective Action.*

*This custom setting does not represent additional capabilities for remediation but rather allows the selection of specific individual actions contained in other options into unique combinations resulting in a "custom" series of actions.

All detected change attempts, resulting corrective action and actual file/configuration changes are captured by the Agent: Security Event subsystem and forwarded to the Master Repository: PostgreSQL & CimTrak databases for report compilation. This allows CimTrak [Administrators] to monitor change activity and identify when change activity may be indicative of malicious activity on an Agent server/Network Host server or Network device through a series of customizable reports.

1.10.6 Security Management

The Security Management security function provides CimTrak [Administrators] with the functions and features necessary for the configuration, deployment and management of the CimTrak TOE and associated Agent components.

The CimTrak TOE features a separate Management Console application which is installed on the Management Console during initial TOE deployment. [Administrator] sessions using this interface are secured using the Secure Communication methods described in Section 1.10.7 and

require identification and authentication by the TSF prior to allowing access to any TSF/stored user data resources.

Agents of the CimTrak may only access data stored within the Master Repository related to that Agent. There are no interfaces provided by the CimTrak Agent component installed on machines throughout the network and Agent activity on those servers is transparent to Agent server users. Configuration of the Agent and [Administrator] interaction occurs exclusively through the CimTrak Management Console communicating with the Master Repository. Following configuration, changes are pushed to the Agents upon startup. For the Common Criteria Evaluated configuration, the TOE maintains [Administrator] groups which serve as roles for the management of the CimTrak application. The groups/roles supported on the Master Repository machine include: Administrator, Standard User and Auditor. The Administrator role has full read/write access to all aspects of the application and the Standard User only has rights as explicitly configured by the Administrator. The Administrator configures this user's rights based on aspects of TOE management to be delegated to a support [Administrator]. The Auditor role has full access to all security management functions on a read-only basis for purpose of supporting an independent auditor role that may review changes managed by the TOE without having write access.

Network devices are managed using separate object groups for each network device which allows the [Administrator] to specify configuration settings on a device basis. Network devices are differentiated by IP Address during the initial configuration phase. As with Filesystem Agent, Network Agent configuration data is acquired from the Master Repository (Postgre DB subsystem) during Agent startup and initialization.

The application provides a comprehensive Graphic User Interface (GUI) which is used by CimTrak [Administrators] for configuring the application and reviewing reports/audit records produced during operation. The Security Management interface is made up of 4 sections: System Menu, Toolbar, Tree View and an Information Display area. The System Menu provides a series of pull down menus for executing various commands (typical of Windows Operating Systems). A toolbar is provided with commonly used commands/options which allow the [Administrator] to quickly select commonly used items. The Tree View refers to a graphical based view of the CimTrak Repository, CimTrak Agent server/Network devices and Object groups configured within the application. This offers the familiar vertical tree orientation in the left column for selecting particular Agents or Object Groups. The Information Display area lists information in the right column related to the items selected in the tree view with data columns provided based on the items selected. Tabs are available at the top of the frame which group the information display area options based on the applicable category.

1.10.7 Secure Communications

The Secure Communications security function provides the mechanisms within the CimTrak TOE to assure that communications between the 3 major TOE components is protected against eavesdropping, manipulation in transit or transmission errors.

All communications between TOE components is encrypted utilizing symmetric key cryptography (AES/3DES) and is integrity checked using a keyed hash message authentication code (HMAC) using the SHA1 algorithm as described in Section 1.10.3 Cryptographic Operations. These communications use the TLS secure protocol as supported by the FIPS 140-2 Level 2 validated module included within the Cryptographic subsystem.

The secure communication techniques described above are used to secure CimTrak Agent to Master Repository and CimTrak Management Console to Master Repository sessions.

Network Agent sessions with CimTrak managed Network devices are secured using the SSHv2 protocol.

1.10.8 Protection of TOE Functions

TOE security functions cannot be bypassed by design and the CimTrak application provides application domain controls that protect the underlying CimTrak code execution. In addition, a series of access control mechanisms restrict access to the TSF and stored User Data to authorized entities within the protected application domain. Within the Master Repository machine, the CimTrak application assures that only authorized objects within the protected domain are allowed access. Access to the CimTrak Database is coordinated and performed solely by the CimTrak Server. Upon a request for data from the client/agent, the CimTrak server validates the request for security access and that the client/agent is requested data valid for its security level. If the requested request is approved, then the CimTrak server retrieves and decrypts the information from the CimTrak Database and returns it to the requesting TOE component.

The Master Repository monitors overall Hard Disk Drive available space and may be configured to issue an email to the applicable [Administrator] upon reaching an allocated threshold. Hard Drive space is monitored and upon reaching the configured percentage an email may be sent indicating a threshold has been reached. This feature is used for the monitoring of Hard Drive free space as specified in FAU_STG.3 to assure audit records are not lost due to lack of available storage resources.

The Agent component executes an integrity check of the full executable during startup processes to assure that the integrity of the Agent has not been compromised. During the installation of the Agent component, a hash is created of the executable and is stored in a separate error detection code (EDC) file. During startup, the resident cryptographic module hashes the executable and verifies the hashed value against the value stored in the EDC. In the event the comparison process fails or the EDC file is missing, the Agent will not start.

Agents do not listen on any ports, only the Master Repository machine is allowed to listen for Agent communication requests, by design. Agents must initiate communication with the Master Repository and communicate exclusively via TLS secure sessions. Agent passwords are generated during initial configuration using an RNG in a form that is not accessible by human entities (including the Administrator role) and this value is stored as a one way hash.

The TOE stores authentication data used to log on to Network Devices in encrypted form within the CimTrak Master Repository.

The CimTrak application requires that all entities communicating with the TSF are identified and authenticated prior to allowing access. This is enforced as part of the Identification and Authentication security function noted above in Section 1.10.2. Agents may only communicate with the Master Repository (housing the essential application and TSF/User Data databases) after successfully negotiating a Diffie-Hellman or RSA based key exchange, establishing a secure session using TLS and after providing a valid username/password combination. An Agent heartbeat is also monitored by the Heartbeat Detection subsystem within the Master Repository to detect if an Agent presence on the network is unexpectedly interrupted. In the event of a loss of Agent communication, the Agent must re-authenticate to the TSF prior to gaining access.

The Management Console does not maintain TSF data local on disk media storage facilities. The Agent and Management Console components download needed TSF data at the time of use for storage in volatile memory. Following a session, the Management Console only retains registry entries and language preferences within the application. The Agent component also retains a minimal amount of data between sessions. This data is required to initiate and validate the next session during startup and is stored in encrypted form within the Agent component.

1.11 Features Not Evaluated/Excluded from the CC Evaluated configuration

The following features are not evaluated as part of the Common Criteria Evaluation:

- Use of Telnet for Network Agent to Device communication*
- Use of Master Repository FTP Interface
- Compliance Templates (“Compliances”) (PCI, SOX, FISMA)
- CimTrak Industrial Agent (Agent intended for Industrial applications i.e.: PLC vs. IT Resources) – This Agent Component is not included in the Common Criteria release of the product. The Industrial Agent is for Industrial applications (manufacturing machinery) and has no use in an Operational Environment.
- Command Line Interface (CLI) – the Security Management security function is only supported through the Graphical User Interface (GUI) for the Common Criteria Evaluated configuration.
- Configuration Support Interface – executed from the Command Line
- Use of the CimTrak Communications protocol for session encryption
- All Fedora-Linux operating systems are excluded.
- All Apple-Macintosh operating systems are excluded.
- All Sun Solaris operating systems are excluded.
- All Hewlett Packard-HPUX operating systems are excluded.

*Telnet is supported by the product but is excluded from the CC Evaluated Configuration as it does not support encryption.

2 Conformance Claims

The TOE is Common Criteria (CC) Version 3.1, Revision 2 Part 2 Extended

The TOE is Common Criteria (CC) Version 3.1, Revision 2 Part 3 Conformant at EAL 4 + ALC_FLR.2

The TOE is compliant with International Interpretations with effective dates on or before 02/07/08.

No PP compliance claims are made for this Security Target.

3 Security Problem Definition

The TOE is intended to be used either in environments in which, at most, sensitive but unclassified information is processed, or the sensitivity level of information in both the internal and external networks is equivalent.

This section contains assumptions regarding the security environment and the intended usage of the TOE and threats on the TOE and the Operational Environment.

3.1 Assumptions

The following conditions are assumed to exist in the operational environment.

A.ADMIN	The [Administrators] are appropriately trained, not careless, not willfully negligent, non-hostile and follow and abide by the instructions provided in the guidance documentation; however, they are capable of error.
A.PHYSEC	The Master Repository, Network Hosts (Network Agents) and Servers (Filesystem Agents) are housed in a physically secure server room environment.
A.GENPUR	There are no general-purpose computing capabilities (e.g., the ability to execute arbitrary code or applications) and non-TOE related storage repository capabilities on the Network Hosts or the Master Repository machine.
A.PUBLIC	The TOE does not host public data.
A.REMACC	Authorized [Administrators] may access the TOE remotely from the internal and external networks.
A. TIME_STAMP	The Operational Environment shall provide an accurate time source for use in time stamps.

3.2 Threats Addressed by the TOE

The threats discussed below are addressed by the CimTrak TOE. The threat agents are either unauthorized persons or External entities not authorized to use the TOE itself.

T. ADMIN_ERROR	An [Administrator] may incorrectly install or configure the TOE resulting in ineffective security mechanisms.
T.AUDACC	Persons may not be accountable for the actions that they conduct because the audit records are not reviewed, thus allowing an attacker to escape detection.
T.CHANGE	A user or IT entity may make unauthorized changes to files/configuration data on resources within the TSC.

T.MASQUERADE	A user or process may masquerade as another entity in order to gain unauthorized access to data or TOE resources
T.POOR_DESIGN	Unintentional errors in requirements specification or design of the TOE may occur, leading to flaws that may be exploited by a casually mischievous user or program.
T.POOR_IMPLEM	Unintentional errors in implementation of the TOE design may occur, leading to flaws that may be exploited by a casually mischievous user or program.
T.POOR_TEST	Lack of or insufficient tests to demonstrate that all TOE security functions operate correctly (including in a fielded TOE) may result in incorrect TOE behavior being discovered thereby causing potential security vulnerabilities.
T.MEDIATE	A user may gain unauthorized access to user data for which they are not authorized according to the TOE security policy.
T.PROCOM	An unauthorized person or unauthorized external IT entity may be able to view, modify, and/or delete security related information that is sent between a remotely located authorized [Administrator] and the TOE.
T.TSF_COMP	A user or process may cause, through an unsophisticated attack, TSF data, or executable code to be inappropriately accessed (viewed, modified, or deleted).
T.SELPRO	An unauthorized person may read, modify, or destroy security critical TOE configuration data.
T.UNIDENT_ACTION	Failure of the authorized [Administrator] to identify and act upon unauthorized actions may occur.

3.3 Organizational Security Policies

P.ACCOUNT	The authorized users of the TOE shall be held accountable for their actions within the TOE.
P. BANNER	The management console component of the TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the system. Note: This policy is only applicable for the Management Console, and is not applicable to the rest of the TOE due to the absence of a client interface that is capable of displaying an access banner.
P.CRYPTO	Only NIST FIPS 140-2 validated cryptography (methods and implementations) are acceptable for key management.

P.ROLES	The TOE shall provide an authorized Administrator role for secure administration of the TOE. This role shall be separate and distinct from other authorized users.

4 SECURITY OBJECTIVES

This section describes the security objectives for the TOE and the TOE's operating Environment. The security objectives are divided between TOE Security Objectives and Security Objectives for the Operating Environment.

The following are the IT security objectives for the TOE:

O.AUDIT_GEN	The TOE will provide the capability to detect and create records of security relevant events associated with users.
O.ADMIN_GUID	The TOE will provide [Administrators] with the necessary information for secure management.
O.ADMIN_ROLE	The TOE will provide authorized [Administrator] roles to isolate administrative actions.
O.AUDREC	The TOE must provide a means to record a readable audit trail of security-related events, with accurate dates and times, and a means to sort the audit trail based on relevant attributes.
O.AUDIT_PROT	The TOE will provide the capability to protect audit information.
O. BANNER	The Management Console component of the TOE shall display an advisory warning regarding use of the TOE.
O.CHANGE	The TOE shall provide the ability to detect changes to selected file/configuration objects on remote file servers/network devices and perform remediation actions to preserve a secure state.
O.CRYPTO	The TOE shall include one or more cryptographic (modules) that are all validated at FIPS 140-2 series Level 1 or higher. This FIPS 140-2 validated module or modules will perform one or more of the following: key pair generation, digital signature generation and verification, encryption, decryption, secure hash, random number generation and HMAC.
O.CONFIG	The configuration of the TOE is fully identified in a manner that will allow implementation errors to be identified and corrected with the TOE being redistributed promptly.
O.DOC_DESIGN	The design of the TOE is adequately and accurately documented.
O.IDAUTH	The TOE provides mechanisms that control logical user access to the TOE and must uniquely identify and authenticate the claimed identity of all users/External entities, before granting access to TSF or User Data within the TSC.
O. FUNC_TEST	The TOE will undergo security functional testing that demonstrates that the TSF satisfies its security functional requirements.
O.SELF_TEST	The TOE will provide functionality to perform integrity testing of TOE components during startup.

O.MANAGE	TOE will provide all the functions and facilities necessary to support the [Administrators] in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.
O.ENCRYPT	The Master Repository component of the TOE must protect the confidentiality of its dialogue with an authorized [Administrator] (console) or CimTrak Agent components through encryption during sessions from the connected network.
O.MEDIATE	The TOE must protect user data in accordance with its security policy.
O.SELPRO	The Master Repository must protect itself against attempts by unauthorized users to bypass, deactivate, or tamper with TOE security functions.
O.TOE_ACCESS	The TOE will provide mechanisms that control a user's logical access to the TOE.
O.VULN_ANALY	The TOE will undergo some vulnerability analysis to demonstrate that the design and implementation of the TOE does not contain any obvious flaws.

4.1 Security Objectives for the Environment

OE.TIME_STAMPS	The Operational Environment shall provide an accurate time source for use in time stamps for audit records.

The following are the non-IT security objectives, which, in addition to those assumptions, are to be satisfied without imposing technical requirements on the TOE. That is, they will not require the implementation of functions in the TOE hardware and/or software. Thus, they will be satisfied largely through application of procedural or administrative measures.

OE.ADMIN	The [Administrators] are appropriately trained, not careless, not willfully negligent, and follow and abide by the instructions provided in the guidance documentation.
OE.PHYSEC	The Master Repository and Network Host server is in a physically secure server room environment.
OE.GENPUR	There are no general-purpose computing capabilities (e.g., the ability to execute arbitrary code or applications) and non TOE related storage repository capabilities on the Master Repository machine.
OE.PUBLIC	The TOE does not host public data.
OE.NOEVIL	Authorized [Administrators] are non-hostile and follow all [administrator] guidance; however, they are capable of error.
OE.REMACC	Authorized [Administrators] may access the TOE remotely from the internal and external networks.
OE.ADMTRA	Authorized [Administrators] are trained as to establishment and maintenance of security policies and practices.

4.2 Mapping of Security Environment to Security Objectives

The following table represents a mapping of the threats to the security objectives defined in this ST.

	T. ADMIN_ERROR	T.AUDACC	T.CHANGE	T.MASQUERADE	T.POOR_DESIGN	T.POOR_IMPLEM	T.POOR_TEST	T.TSF_COMP	T.SELPRO	T.MEDIAT	T.UNIDENT_ACTION	T.PROCOM	P. BANNER	P.ACCOUNT	P.ROLES	P.CRYPTO
O.AUDIT_GEN		X												X		
OE. TIME_STAMPS		X														
O.ADMIN_GUID	X										X					
O.ADMIN_ROLE															X	
O. BANNER								X					X			
O.TOE_ACCESS				X										X		
O.CHANGE			X							X						
O.CONFIG					X	X										
O.VULN_ANALY					X	X	X									
O.DOC_DESIGN					X		X									
O.FUNC_TEST						X	X									
O.SELPRO								X	X							
O.SELF_TEST									X							
O.MANAGE								X			X	X				
O.MEDIATE										X						
O.AUDREC		X									X			X		
O.AUDIT_PROT		X									X			X		
O.IDAUTH								X			X	X				
O.CRYPTO																X
O.ENCRYPT												X				

Table 3: Summary of Mappings Between Threats and IT Security

4.3 Rationale For IT SECURITY OBJECTIVES

T. ADMIN_ERROR	O.ADMIN_GUID helps to mitigate this threat by ensuring the TOE [Administrators] has guidance that instructs them how to administer the TOE in a secure manner. This guidance helps to mitigate the potential for error in configuration of the TOE in a secure manner.
T.AUDACC	O.AUDIT_GEN addresses this policy by providing the authorized [Administrator] with the capability of configuring the audit mechanism to record the actions of a specific user, or review the audit trail based on the identity of the user. Additionally, the [Administrator's] ID is recorded when any security relevant change is made to the TOE (e.g., access rule modification, start-stop of the audit mechanism, establishment of a trusted channel, etc). OE.TIME_STAMPS further mitigates this threat by providing a time source in the Operational Environment for use in audit records. This threat is also mitigated by O.AUDREC which provides a means to generate & record a readable audit trail of security-related events, with accurate dates and times, and a means to sort the audit trail based on relevant attributes. This threat is partially mitigated by O.AUDIT_PROT which provides the capability to protect audit information.
T.CHANGE	This threat is mitigated by O.CHANGE which specifies that the TOE shall provide the ability to detect changes to selected file/configuration objects on remote file servers/network devices and perform remediation actions to preserve a secure state.
T.MASQUERADE	O.TOE_ACCESS mitigates this threat by controlling the logical access to the TOE and its resources. By constraining how and when authorized users can access the TOE, and by mandating the type and strength of the authentication mechanism this objective helps mitigate the possibility of a user attempting to login and masquerade as an authorized user. In addition, this objective provides the [Administrator] the means to control the number of failed login attempts a user can generate before an account is locked out, further reducing the possibility of a user gaining unauthorized access to the TOE.

T.POOR_DESIGN	O.CONFIG plays a role in countering this threat by requiring the developer to provide control of the changes made to the TOE's design. O.DOC_DESIGN helps to ensure that the TOE's documented design satisfies the security functional requirements. In order to ensure the TOE's design is correctly realized in its implementation, the appropriate level of functional testing of the TOE's security mechanisms must be performed during the evaluation of the TOE. O.VULN_ANALY ensures that the design of the TOE is analyzed for design flaws.
T.POOR_IMPLEMENT	O.CONFIG plays a role in countering this threat by requiring the developer to provide control of the changes made to the TOE's design, although the previous three objectives help minimize the introduction of errors into the implementation. O. FUNC_TEST increases the likelihood that any errors that do exist in the implementation (with respect to the functional specification, high-level, and low-level design) will be discovered through testing. O.VULN_ANALY helps reduce errors in the implementation that may not be discovered during functional testing. Ambiguous design documentation and the fact that exhaustive testing of the external interfaces is not required may leave bugs in the implementation undiscovered in functional testing.
T.POOR_TEST	O.DOC_DESIGN helps to ensure that the TOE's documented design satisfies the security functional requirements. In order to ensure the TOE's design is correctly realized in its implementation, the appropriate level of functional testing of the TOE's security mechanisms must be performed during the evaluation of the TOE. O. FUNC_TEST increases the likelihood that any errors that do exist in the implementation (with respect to the functional specification, high level, and low-level design) will be discovered through testing. O.VULN_ANALY addresses this concern by requiring a vulnerability analysis be performed in conjunction with testing that goes beyond functional testing. This objective provides a measure of confidence that the TOE does not contain security flaws that may not be identified through functional testing.
T.PROCOM	O.ENCRYPT mitigates this threat by protecting the confidentiality of its dialogue with an authorized [Administrator] or agent components through encryption during sessions from the connected network.

	O.MANAGE is necessary because an access control policy is specified to control access to TSF data. This objective is used to dictate who is able to view and modify TSF data, as well as the behavior of TSF functions. O.IDAUTH mitigates this threat by providing mechanisms that control logical user access to the TOE and must uniquely identify and authenticate the claimed identity of all users/External entities, before granting access to TSF or User Data within the TSC.
T.SELPRO	O.SELPRO mitigates this threat by ensuring that the TOE is capable of protecting itself from attack i.e.: against attempts by unauthorized users to bypass, deactivate, or tamper with TOE security functions.
T.TSF_COMP	O.SELPRO mitigates this threat by ensuring that the TOE is capable of protecting itself from attack i.e.: against attempts by unauthorized users to bypass, deactivate, or tamper with TOE security functions. O.MANAGE is necessary because an access control policy is specified to control access to TSF data. This objective is used to dictate who is able to view and modify TSF data, as well as the behavior of TSF functions. O. BANNER helps mitigate this threat by displaying an advisory warning regarding use of the TOE. O.IDAUTH mitigates this threat by providing mechanisms that control logical user access to the TOE and must uniquely identify and authenticate the claimed identity of all users/External entities, before granting access to TSF or User Data within the TSC.
T.MEDIATE	O.MEDIATE ensures that all accesses to user data are subject to mediation. The TOE requires successful authentication to the TOE prior to gaining access to any controlled-access content. By implementing strong authentication to gain access to these services, an attacker's opportunity to conduct a man-in-the-middle and/or password guessing attack successfully is greatly reduced. Lastly, the TSF will ensure that all configured enforcement functions (authentication, access control rules, etc.) must be invoked prior to allowing a user to gain access to TOE or TOE mediated services. The TOE restricts the ability to modify the security attributes associated with access control rules, access to authenticated and unauthenticated services, etc to the [Administrator]. This feature ensures that no other user can modify

	the information flow policy to bypass the intended TOE security policy. O. CHANGE mitigates this threat by providing the ability to detect changes to selected file objects on remote machines and perform remediation actions to preserve a secure state.
T.UNIDENT_ACTION	The threat of an authorized [Administrator] failing to know about malicious audit events produces the objectives of the authorized [Administrator] having the facilities and knowing how to use them (O.ADMIN_GUID). The threat of an authorized [Administrator] failing to know about malicious audit events produces the objectives of the authorized [Administrator] having the capability to use the mechanisms (O.MANAGE) to review audit records. This threat is partially mitigated by O.AUDREC which provides a means to generate & record a readable audit trail of security-related events, with accurate dates and times, and a means to sort the audit trail based on relevant attributes. This threat is partially mitigated by O.AUDIT_PROT which provides the capability to protect audit information.
P.ACCOUNT	O.AUDIT_GEN mitigates this policy by providing the authorized [Administrator] with the capability of configuring the audit mechanism to record the actions of a specific user, or review the audit trail based on the identity of the user. Additionally, the [Administrator]'s ID is recorded when any security relevant change is made to the TOE (e.g., access rule modification, start-stop of the audit mechanism, establishment of a trusted channel, etc.). O.TOE_ACCESS supports this policy by requiring the TOE to identify and authenticate all authorized users prior to allowing any TOE access or any TOE mediated access on behalf of those users. This policy is supported by O.AUDREC which provides a means to record a readable audit trail of security-related events, with accurate dates and times, and a means to sort the audit trail based on relevant attributes. O.AUDIT_PROT mitigates this threat by providing the capability to protect audit information.

P.BANNER	O.BANNER mitigates this threat by providing that the TOE displays an advisory warning regarding use of the TOE.
P.ROLES	The TOE has the objective of providing an authorized Administrator role for secure administration. The TOE may provide other roles as well, but only the role of authorized [Administrator] is required (O.ADMIN_ROLE).
P.CRYPTO	The TOE has the objective of providing Cryptographic functionality (O.CRYPTO) validated to FIPS 140 series Level 1 or higher. This FIPS 140 series validated module or modules will perform one or more of the following: key pair generation, digital signature generation and verification, encryption, decryption, secure hash, random number generation and HMAC.

4.4 Rationale For Assumption Coverage

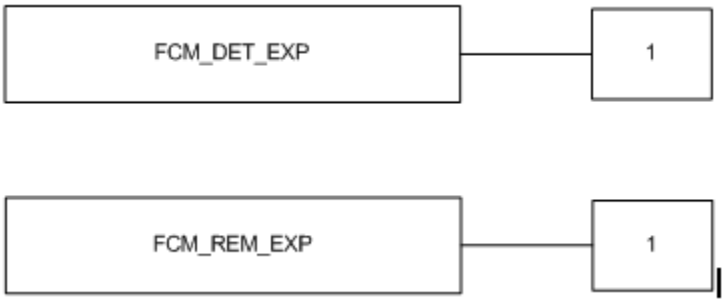
This section provides a justification for each assumption and the security objectives for the environment which cover that assumption.

A.ADMIN	The assumption A.ADMIN is addressed in the objective OE.ADMIN which ensures that [Administrators] of the TOE Environment are competent, trustworthy and conform to guidance supplied in applicable documentation. The assumption is further supported by OE.NO_EVIL which specifies that [Administrators] are non-hostile and by OE.ADMTRA which specifies that in establishment and maintenance of security policies and practices.
A.PHYSEC	The assumption A.PHYSEC is addressed in the objective OE.PHYSEC which specifies that the Master Repository and Network Host server is physically secure and physical access to the Master Repository is controlled to assure only authorized [Administrators] have access.
A.GENPUR	The assumption A.GENPUR is restated directly in the objective OE.GENPUR which specifies that the [Administrator] ensures there are no general-purpose computing capabilities (e.g., compilers, editors, or user applications) available on the Master Repository machine.
A.PUBLIC	The assumption A. PUBLIC is restated directly in the objective OE.PUBLIC which specifies that the TOE does not host public data.
A.REMACC	The assumption A.REMACC is restated directly in the objective OE.REMACC which specifies that Authorized [Administrators] may access the TOE remotely from the internal and external networks.
A.TIME_STAMP	The assumption A.TIME_STAMP is restated directly in the objective OE.TIME_STAMPS which specifies that the Operational Environment shall provide an accurate time source for use in time stamps for audit records.

5 Extended Components Definition

5.1 Class FCM: Change Management (Explicit Class)

This class contains families of functional requirements that relate to the monitoring of specific IT Entities for changes to specified objects and upon detection of those changes, implementation of remediation actions which mitigate potential security violations or other malicious activity associated with unauthorized changes to specified objects.



5.1.1 FCM_DET_EXP (Detection)

Family Behavior

This Family defines the action to be taken in detecting changes to objects that may be indicative of a malicious access attempt or security violation.

Component Leveling

At FCM_DET_EXP.1 Change Detection, the TSF shall detect specified change events from the targeted resource and collected specified data related to change event detected.

Management: FCM_DET_EXP.1

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

a. The management of objects to be monitored and events subject to detection by the TSF.

Audit: FCM_DET_EXP.1

The following actions should be auditable if FAU_GEN.1 Security audit data generation is included in the PP/ST:

a. Minimal: Change events to monitored objects detected by the TSF, applicable file/object name, location, date/time, type of change made.

5.1.1.1 FCM_DET_EXP.1 Change Detection (Explicit)

Hierarchical to: No other components.

Dependencies: None

FCM_DET_EXP.1.1	The [assignment: *TSF Component*] shall be able to detect the following events from the targeted IT System resource(s): [assignment: types of change event detected by the applicable TSF component]
FCM_DET_EXP.1.2	The TSF shall collect the following information relating to the event: [assignment: *information collected by the TSF related to the detection events listed in FCS_DET_EXP.1.1*]
FCM_DET_EXP.1.3	Change detection shall be configurable by the [assignment: *authorized users*] to occur at the following intervals: [assignment: *type of detection/detection interval time value*]

5.1.2 FCM_REM_EXP (Remediation)

Family Behavior

This Family defines the action to be taken in remediating detected changes to objects to mitigate a malicious access attempt or potential security violation.

Component Leveling

At FCM_REM_EXP.1 Change Remediation, the TSF shall perform specified remediation actions based on events detected in FCS_DET_EXP.1.

Management: FCM_REM_EXP.1

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

a. The configuration/management of remediation actions to be taken upon detection of changes to objects monitored by the TSF.

Audit: FCM_REM_EXP.1

The following actions should be auditable if FAU_GEN.1 Security audit data generation is included in the PP/ST:

a. Minimal: Remediation action taken following detection of changes to configured items by the TSF including File/Object identification, (Remediation) Action Taken, Success/Failure, Time/Date.

5.1.2.1 FCM_REM_EXP.1 Change Remediation (Explicit)

Hierarchical to: No other components.

Dependencies: FCS_DET_EXP.1

FCM_REM_EXP.1.1 The TSF shall perform the following remediation function(s) on all changes to [assignment: *change properties/attributes monitored*] on [assignment: *targeted IT Entities*]: [assignment: *remediation actions*]

5.2 Class FPT: Protection of the TSF

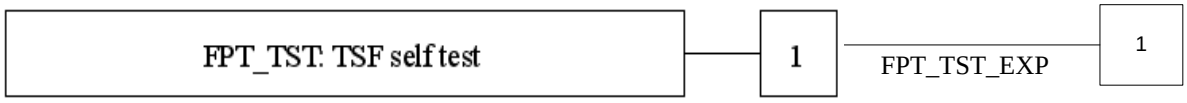
This class contains families of functional requirements that relate to the integrity and management of the mechanisms that constitute the TSF and to the integrity of TSF data (Class FPT description unchanged from CC Part II).

5.2.1 TSF self test (FPT_TST_EXP)

Family Behavior

The family defines the requirements for the self-testing of TSF executable code.

Component leveling



FPT_TST_EXP.1 TSF testing provides the ability to test the TSF's correct operation. These tests may be performed at start-up, periodically, at the request of the authorized user, or when other conditions are met. It also provides the ability to verify the integrity of TSF data and TSF itself.

Management: FPT_TST_EXP.1

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

a) None

Audit: FPT_TST_EXP.1

The following actions should be auditable if FAU_GEN.1 Security audit data generation is included in the PP/ST:

a) Basic: Execution of the TSF self tests and the results of the tests.

5.2.1.1 FPT_TST_EXP.1 TSF testing

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_TST_EXP.1.1 The TSF shall run a suite of self tests [selection: during initial start-up, periodically during normal operation, at the request of the authorized user, at the conditions [assignment: conditions under which self test should occur]] to demonstrate the correct operation of [selection: [assignment: parts of TSF], the TSF].

6 Security Requirements

The security requirements that are levied on the TOE and the Operational Environment are specified in this section of the ST. These security requirements are defined in Sections 0

TOE Security Functional Requirements	
FAU_ARP.1	Security Alarms
FAU_GEN.1	Audit Generation
FAU_GEN.2	User Identity Association
FAU_SAA.1	Potential Violation Analysis
FAU_SAR.1	Audit Review
FAU_SAR.3	Selectable Audit Review
FAU_STG.1	Protected audit trail storage
FAU_STG.3	Action in case of possible audit data loss
FCS_CKM.1a	Symmetric Key Generation – ***software RNG***
FCS_CKM.1b	Asymmetric Key Generation – ***software RNG***
FCS_CKM.1c	Symmetric Key Generation – ***private key generation***
FCS_COP.1a	Cryptographic operation: ***Repository Encryption***
FCS_COP.1b	Cryptographic operation: ***File Comparison (Hashing) – SHA1***
FCS_COP.1c	Cryptographic operation: ***Agent Session Encryption***
FCS_COP.1d	Cryptographic operation: ***Console Session Encryption***
FCS_COP.1e	Cryptographic operation: ***Network Agent device sessions***
FCS_CKM.4	Cryptographic key destruction
FDP_ACC.1	Subset Access Control
FDP_ACF.1	Security Attribute access control
FDP_ITT.1	Basic internal transfer protection
FDP_ROL.1	Basic Rollback

TOE Security Functional Requirements	
FIA_AFL.1	Authentication Failure Handling
FIA_ATD.1	User Attribute Definition
FIA_UID.1	Timing of Identification– ***CimTrak Management Console to Master Repository***
FIA_UID.2	Identification before any action – ***Agent to Master Repository***
FIA_UAU.1	Timing of authentication– ***CimTrak Management Console to Master Repository***
FIA_UAU.2	Authentication before any action **–** ***Agent to Master Repository***
FIA_SOS.1	TSF Verification of Secrets
FIA_SOS.2	TSF Generation of Secrets
FMT_MOF.1a	Management of Security Function behavior-***Administrator***
FMT_MOF.1b	Management of Security Function behavior-***Administrator, Standard User***
FMT_MSA.1a	Management of Security Attributes-***Administrator***
FMT_MSA.1b	Management of Security Attributes- ***Administrator, Standard User***
FMT_MSA.3	Static Attribute Initialization
FMT_MTD.1a	Management of TSF Data **–** ***Create, Modify, Delete***
FMT_MTD.1b	Management of TSF Data **–** ***Create, Modify, Delete***
FMT_MTD.1c	Management of TSF Data **–** ***Query***
FMT_MTD.1d	Management of TSF Data **–** ***Query***
FMT_MTD.1e	Management of TSF Data **–** ***Query***
FMT_SMF.1	Specification of Management Functions
FMT_SMR.1	Security Roles
FPT_ITT.1	Internal TOE TSF data transfer
FTA_SSL.3	TSF Initiated Termination
FTA_TAB.1	Default TOE access banners
FCM_DET_EXP.1	Change Detection (Explicit)
FCM_REM_EXP.1	Change Remediation (Explicit)
FPT_TST_EXP.1	TSF Testing (Explicit)

Table 4: Functional Requirements

6.1 TOE Security Functional Requirements

The SFRs defined in this section are taken from Part 2 of the CC.

6.1.1 Class FAU: Security Audit

6.1.1.1 FAU_ARP.1 – Security Alarms

FAU_ARP.1.1 The TSF shall take **the configured remediation action (if applicable), log the event and email the configured email address** upon detection of a potential security violation.

6.1.1.1 FAU_GEN.1 – Audit Generation

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

Start-up ~~***and shutdown****~~ of the audit functions; All auditable events for the not specified level of audit; and **events listed in Table 5**.

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

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

For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, **no other audit relevant information**.

*Audit records cannot be shutdown.

Component	Event	Details
FAU_ARP.1	Security Alarms	Alerts/Action taken due to security violation (see Section 7.1.1)
FAU_SAA.1	Potential Violation Analysis	Enable/Disable of analysis mechanism, automated responses by mechanism
FCS_COP.1c,d,e	Agent/Console to Repository session encryption/decryption	Error Creating new SSL object Error accepting SSL connection
FCS_COP.1a	Repository data encryption/decryption	Invalid encryption cipher; encrypt Invalid encryption cipher; decrypt Key Mismatch: encrypt, decrypt HMAC Mismatch
FDP_ACF.1	Security Attribute access control	Successful requests to perform an operation on a subject covered by the CimTrak Access Control SFP
FDP_ROL.1	Rollback of User Data	All Successful Rollback operations
FIA_UID.1	All use of the user identification mechanism	User identity
FIA_UAU.2	All use of the user authentication mechanism	Success/Failure authentication
FMT_MOF.1	All modifications in the behavior of the functions of the TSF	Security function modification affecting behavior
FMT_SMR.1	Modifications to the group of users that are part of a role	User identity
FTA_SSL.3	TSF initiated termination	Termination of session by locking mechanism, time/date/username
FCM_DET_EXP.1	Change Detection Events	Event Detected, File ID, Description of change made, Agent server/Network Host server User who initiated the change, Time/Date
FCM_REM_EXP.1	Remediation Actions	File ID, Action Taken, Success/Failure, Time/Date

Table 5: Audited Events

6.1.1.2 FAU_GEN.2 –User Identity Association

FAU_GEN.2.1 The TSF shall be able to associate each auditable event with the identity of the user* that caused the event.

*user here refers to an Administrative-user for Master Repository audit events, and the user identified by the Agent server/Network Host server Operating System for Agent audit events: i.e.: file change attempts.

6.1.1.3 FAU_SAA.1 – Potential Violation Analysis

FAU_SAA.1.1 The TSF shall be able to apply a set of rules in monitoring the audited events and based upon these rules indicate a potential violation of the enforcement of the SFRs.

FAU_SAA.1.2 The TSF shall enforce the following rules for monitoring audited events:

- Accumulation or combination of **unauthorized or unexpected changes to locked files on Agent servers/Network devices** known to indicate a potential security violation;
- **Exceeding the Administrator configured threshold for Master Repository disk space**
- **Agent does not check-in with the Master Repository within a configured period of time (heartbeat monitor).**
- **Failure of Master Repository integrity checking during startup**
- **Exceeding the Administrator, Standard User configured threshold for "Monitor Parameters" (CPU, Memory, Disk Space)**
- **No other rules.**

6.1.1.4 FAU_SAR.1 – Audit Review

FAU_SAR.1.1 The TSF shall provide **Administrator, Auditor** with the capability to read **audit data listed in Table 5** from the audit records.

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

6.1.1.5 FAU_SAR.3 – Selectable Audit Review

FAU_SAR.3.1 The TSF shall provide the ability to perform **sorting** of audit data based on **Event Date/Time.**

6.1.1.6 FAU_STG.1 Protected audit trail storage

FAU_STG.1.1 The ~~TSF~~ ***Master Repository component of the TSF*** shall protect the stored audit records from unauthorized deletion.

FAU_STG.1.2 The ~~TSF~~ ***Master Repository component of the TSF*** shall be able to prevent unauthorized modifications to the stored audit records in the audit trail.

6.1.1.7 FAU_STG.3 Action in case of possible audit data loss

FAU_STG.3.1 The ~~TSF~~ ***Master Repository component of the TSF*** shall **send an email to the configured email address** if the ***Master Repository storage resource exceeds*** **a configured percentage of available storage**.

Note: The entire drive is monitored for available storage as there is not a separate audit record allocation.

6.1.2 Class FCS: Cryptographic Support

6.1.2.1 FCS_CKM.1a – Symmetric Key Generation – *software RNG*

FCS_CKM.1.1a The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm Software RNG using **AES/3DES** and specified cryptographic key sizes **128, 192, 256 bit (192 bit 3DES)** that meet the following: **ANSI X9.31, FIPS 186-2, or FIPS SP 800-90**

6.1.2.2 FCS_CKM.1b – Asymmetric Key Generation – *software RNG*

FCS_CKM.1.1b The TSF shall generate asymmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm **Diffie-Hellman, RSA** and specified cryptographic key sizes **512, 1024, 2048** that meet the following: **RFC 2459 (x.509), RFC 2631 (DH), ANSI X9.31 (RSA) FIPS PUB 140-2**.

6.1.2.3 FCS_CKM.1c – Symmetric Key Generation – *private key generation*

FCS_CKM.1.1c The TSF shall generate cryptographic keys ***from a provided passphrase string*** in accordance with a specified cryptographic key generation algorithm ***Pseudorandom String to Key Derivation Operation*** using **HMAC-SHA-1** and specified cryptographic key sizes **160 bits with an 8 bit salt** that meet the following: **RFC 2898, PBKDF-2, PKCS#5v2**

6.1.2.4 FCS_CKM.4 Cryptographic key destruction

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method **zeroization** that meets the following: **FIPS 140-2**

6.1.2.5 FCS_COP.1a – Cryptographic operation: *Repository Encryption*

FCS_COP.1.1a The TSF shall perform **R*epository Encryption*/Decryption** in accordance with a specified cryptographic algorithm **AES, 3DES** and cryptographic

key sizes **128,192,256 (3DES 192 bit)** that meet the following: **RFC 3565 (AES), RFC 2451 (3DES)**, **FIPS PUB 140-2, FIPS PUB 46-3 (3DES)**.

6.1.2.6 FCS_COP.1b – Cryptographic operation: *File Comparison (Hashing) – SHA1*

FCS_COP.1.1b The TSF shall perform **Hashing** in accordance with a specified cryptographic algorithm **SHA1** and cryptographic **512 bit block size** that meet the following: **RFC 3174, FIPS PUB 140-2**

6.1.2.7 FCS_COP.1c – Cryptographic operation: *Agent Session Encryption*

FCS_COP.1.1c The TSF shall perform **Agent Session Encryption/Decryption/Integrity Checking** in accordance with a specified cryptographic algorithm **TLS using AES, 3DES with SHA1 HMAC** and cryptographic key sizes **128,192,256 (3DES 192 bit) with 1024, 2048 bit (RSA)** that meet the following: **FIPS PUB 140-2, FIPS PUB 46-3 (3DES) RFC 2246 (TLS)**.

6.1.2.8 FCS_COP.1d – Cryptographic operation: Console Session Encryption

FCS_COP.1.1d The TSF shall perform **Console session encryption/decryption** in accordance with a specified cryptographic algorithm **TLS using AES, 3DES with SHA1 HMAC** and cryptographic key sizes **128,192,256 (3DES 192 bit) with - 1024, 2048 bit (RSA)** that meet the following: **FIPS PUB 140-2, FIPS PUB 46-3 (3DES), RFC 2246 (TLS)**.

6.1.2.9 FCS_COP.1e – Cryptographic operation: *Network Agent device sessions*

FCS_COP.1.1e The TSF shall perform **Network Agent session encryption/decryption** in accordance with a specified cryptographic algorithm **SSHv2 using AES, 3DES** and cryptographic key sizes **128,192,256 (3DES 192 bit)** that meet the following: **RFC 4254**.

6.1.3 Class FDP: User Data Protection

6.1.3.1 FDP_ACC.1 – Subset Access Control

FDP_ACC.1.1 The TSF shall enforce the **CimTrak access control SFP** on:

Subjects: **External entities** accessing a given Agent server file system/Network device, [Administrators] or External entities accessing the authoritative copy* on the master repository**

Objects: **Files or configuration data configured as "locked", Files/configuration data contained in locked object groups, Authoritative copy of user data stored on the master repository, Agent Files or individual File Objects selectively encrypted with an Administrator assigned passphrase ("Private Key" feature)**

Operations: **Read, Write**

*Authoritative Copy (as defined in Section 1.5.1) Refers to a saved copy of "locked" User data stored in the Master Repository for the purpose of restoring files to the last known approved state.

**External entities – refers to a human or non human user within either the TOE or Operational Environment

6.1.3.2 FDP_ACF.1 – Security Attribute access control

FDP_ACF.1.1 The TSF shall enforce the **CimTrak access control SFP** to objects based on the following: **Files designated as locked on Agent server/Network devices, User data stored within CimTrak Repository (authoritative copy), Agent or File Objects selectively encrypted through the "Private Key" feature**.

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 locked object on an Agent server/Network device may be added or modified if the associated watch properties for the file or object group is set to Log Only, Update Baseline or Prompt (with administrator action).

a user data file/configuration stored within the master repository may be accessed by a CimTrak Agent when the Agent has been configured by the authorized Administrator to "restore from repository" a locked file that has been changed user data may only be accessed by the original Agent

User data stored within the Master Repository may be accessed by a CimTrak [Administrator] upon successful authentication by the TSF as a user holding the role: Administrator, Standard user, Auditor.

User data stored within the Master Repository and encrypted with an Administrator assigned passphrase ("Private Key" feature) may be accessed by a CimTrak [Administrator] upon successful authentication by the TSF as a user holding the role: Administrator and upon entering the applicable passphrase used to secure the applicable Agent data or user Data Object.

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **No additional rules**.

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the **no additional rules**.

6.1.3.3 FDP_ITT.1 – Basic internal transfer protection

FDP_ITT.1.1 The TSF shall enforce the **CimTrak access control SFP** to prevent the disclosure, modification of user data when it is transmitted between physically-separated parts of the TOE.

6.1.3.4 FDP_ROL.1 – Basic Rollback

FDP_ROL.1.1 The TSF shall enforce **the CimTrak access control SFP** to permit the rollback of the **read/write/delete** on the **"locked" data stored on the Agent server/Network device**.

FDP_ROL.1.2 The TSF shall permit operations to be rolled back within the **Administrator configured value of data generations (default 10 generations)**.

6.1.4 Class FIA: Identification and Authentication

6.1.4.1 FIA_AFL.1 Authentication Failure Handling

FIA_AFL.1.1 The TSF shall detect when an administrator configurable positive integer within **1 – 99 of** unsuccessful authentication attempts occur related to CimTrak Management **Console login to the CimTrak Master Repository.**

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall **disable the User Account**.*

*note – this lockout feature does not apply to the Administrator role so the root Administrator is not locked out of the application.

6.1.4.2 FIA_ATD.1 User Attribute Definition

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users: **Username, Password, User Group/Role assignment***

*note: user group and role are synonymous since CimTrak uses User Groups to create/enforce roles and manage privileges within the application.

6.1.4.3 FIA_UID.1 Timing of Identification– CimTrak Management Console to Master Repository

FIA_UID.1.1 The TSF shall allow **Selection of Applicable Language, Access the "About" screen, access to the Display Banner** on behalf of the user to be performed before the user is identified.

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

6.1.4.4 FIA_UID.2 User identification before any action – Agent to Master Repository

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

6.1.4.5 FIA_UAU.1 Timing of authentication – CimTrak Management Console to Master Repository

FIA_UAU.1.1 The TSF shall allow **Selection of Applicable Language, Access the "About" screen, access to the Display Banner** of the user to be performed before the user is authenticated.

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

6.1.4.6 FIA_UAU.2 User authentication before any action – Agent to Master Repository

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

6.1.4.7 FIA_SOS.1 Verification of secrets

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet **minimum number of characters (10), 2 lowercase, 2 uppercase, 2 numbers, and 2 special characters for [Administrators] passwords, verified not to include dictionary terms, verified not to match previous 4 passwords.**

6.1.4.8 FIA_SOS.2 TSF Generation of Secrets

FIA_SOS.2.1 The TSF shall provide a mechanism to generate secrets that meet **a random number generated alphanumeric password which meets minimum number of characters (10), 2 lowercase, 2 uppercase, 2 numbers, and 2 special characters for [Administrators] passwords, verified not to include dictionary terms, verified not to match previous 4 passwords.**

FIA_SOS.2.2 The TSF shall be able to enforce the use of TSF generated secrets for **password generation.**

6.1.5 Class FMT: Security Management

6.1.5.1 FMT_MOF.1a – Management of Security Function behavior - *Administrator*

FMT_MOF.1.1a The TSF shall restrict the ability to modify the behavior of the functions:

- **ID & Authentication (Create Users, Modify User attributes – privileges, role assignment)**
- **Security Audit (email notification settings, applicable email address for notification)**

to **the Administrator**.

6.1.5.2 FMT_MOF.1b – Management of Security Function behavior – *Administrator, Standard User*

FMT_MOF.1.1b The TSF shall restrict the ability to modify the behavior of the functions:

- **Security Management (application configuration)**
- **Cryptographic settings Agent/Console (algorithm, key size)**
- **Access Control (File/Object locked status)**
- **Change Management (File/Object monitor settings, remediation settings)**
- **Repository Properties window (change Console-Repository Encryption, Global Agent-Repository Encryption, SMTP, Syslog, SNMP, Logon Banner, Block/Grant Access to IP addresses)**

to **the Administrator, Standard User**.

6.1.5.3 FMT_MSA.1a – Management of Security Attributes-*Administrator, Standard User*

FMT_MSA.1.1a The TSF shall enforce the **CimTrak Access Control SFP** to restrict the ability to create, modify the security attributes

- **File Object Group Properties (group name, location, description, date/time, contact, URL, number of revisions to keep, number**

of events to keep, number of intrusions to keep, keep intrusion size)

- **File Watch Group Properties (corrective action setting, Authoritative Copy enable/disable, File Comparison method (hash), Event Detection method, Sync enable)**
- **Lock, Unlock Object Groups**
- **Repository Storage threshold value (% full)**
- **Cryptographic function settings – Agent Data Encryption (Encryption type (algorithm), key length, HMAC Type (algorithm)**
- **Cryptographic function settings – Management Console (Encryption type (algorithm), key length, HMAC Type (algorithm)**

to **the Administrator, Standard User**.

6.1.5.4 FMT_MSA.1b – Management of Security Attributes-*Administrator*

FMT_MSA.1.1b The TSF shall enforce the **CimTrak Access Control SFP** to restrict the ability to create, modify the security attributes

- **Permissions Window (grants permissions and emails to specified Users or User Groups)**
- **CimTrak User Group/Role assignments**
- **Agent log upload interval**

to **the Administrator**

6.1.5.5 FMT_MSA.3 – Static Attribute Initialization

FMT_MSA.3.1 The TSF shall enforce the **CimTrak access control SFP** to provide restrictive default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the **None** to specify alternative initial values to override the default values when an object or information is created.

6.1.5.6 FMT_MTD.1a – Management of TSF Data-*Create, Modify, Delete*

FMT_MTD.1.1a The TSF shall restrict the ability to Create, Modify, Delete the

- **Master Repository disk storage threshold % (value)**
- **Master Repository Properties/Configuration parameters**

to **the Administrator, Standard User.**

6.1.5.7 FMT_MTD.1b – Management of TSF Data – *Create, Modify, Delete*

FMT_MTD.1.1b The TSF shall restrict the ability to Create, Modify, Delete the

- **Users window (Add/Delete/Edit Users, as well as assign email addresses to Users)**
- **Users Window: Edit User Accounts**
- **Email security event & notification settings,**

to **the Administrator**

6.1.5.8 FMT_MTD.1c – Management of TSF Data – *Query*

FMT_MTD.1.1c The TSF shall restrict the ability to query the

- **Logged on Users Tab**

to **the Administrator**

6.1.5.9 FMT_MTD.1d – Management of TSF Data – *Query*

FMT_MTD.1.1d The TSF shall restrict the ability to query the

- **Changes Pending Approval Window**

to **the Administrator, Standard User**

6.1.5.10 FMT_MTD.1e – Management of TSF Data – *Query*

FMT_MTD.1.1e The TSF shall restrict the ability to query the

- **View Areas, Document Controls, Object Groups**
- **View Audit Logs**

to **the Administrator, Standard User, Auditor**

6.1.5.11 FMT_SMF.1 – Specification of Management Functions

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

- **File/Object Group management**
- **File Watch Group management**
- **File Monitoring properties management**

- **Event detection method management**
- **Agent management including upload interval**
- **CimTrak User/Group management**
- **Agent Viewing Groups (Area) management**
- **CimTrak Reports Management**
- **File Comparison functions**
- **Email Notification function**
- **Cryptographic key management**
- **"Private Key" feature activation during install (Agent based), via Management Console (object based)**

6.1.5.12 FMT_SMR.1 – Security Roles

FMT_SMR.1.1 The TSF shall maintain the roles **Administrator, Standard User, Auditor**.

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

6.1.6 Class FPT: Protection of the TSF

6.1.6.1 FPT_ITT.1 – Internal TOE TSF data transfer

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

6.1.7 Class FTA: TOE Access

6.1.7.1 FTA_SSL.3 – TSF Initiated Termination

FTA_SSL.3.1 The TSF shall terminate an interactive session after an **Administrator Configured Value (default 10 minutes).**

6.1.7.2 FTA_TAB.1 Default TOE access banners

FTA_TAB.1.1 Before establishing a user session, the ***management console component of the*** TSF shall display an advisory warning message regarding unauthorized use of the TOE.

6.1.8 Class FCM: Change Management (Explicit Class)

6.1.8.1 FCM_DET_EXP.1 Change Detection (Explicit) +

FCM_DET_EXP.1.1 The **CimTrak Agent** shall be able to detect the following events from the targeted IT System resource(s):

Agent Machine Monitoring Parameters: Network Utilization, CPU usage, Fixed Memory Usage (RAM), Available Disk Space

- **Detection of Logged On entities' IP Address/PC Identifier where the locked file modification occurred within a Windows shared directory ("Forensic Data" – Windows Agents only).**

Any modifications to Agent server/Network device files/configuration configured as "locked" based on the following properties:

- **Filesystem Agents: Windows: File/Data Properties as listed in Table 6: File Properties – Windows**
- **Filesystem Agents: Linux: File/Data Properties as listed in Table 7: File Properties – Linux**
- **Network Agents: Network Device Configuration properties (varies based on Network device type)**

FCM_DET_EXP.1.2 The TSF shall collect the following information relating to the event:

a) Date and time of the event, type of event, subject identity, and the outcome of the event; and

b) The actual file properties changed within the file/configuration

c) Connection type which resulted in the file change.

d) Action taken by the TSF upon detection of the event.

e) Agent Machine Monitoring Parameters: Network Utilization (%) CPU usage (%), Fixed Memory Usage (RAM) (%), Disk Space available (%)

f) IP addresses/PC Identifier of entities logged on where a locked file modification occurred within a Windows shared directory ("Forensic Data" – Windows Agents only).

File Properties Locked (Windows)
Hash of File Contents
File Size
Group Security Information
DACL Security Information
Create Time (If not a directory)
Modify Time (If not a directory)
Read Only Configurable Checking
Archive Configurable Checking
Compressed
Hidden
Offline
Reparse Point
Sparse Point
System
Temporary

Table 6: File Properties – Windows

File Properties Locked (Linux)
Hash of File Contents
File Size
Creation Time (If not a directory)
Modify Time (If not a directory)
Read Only Configurable Checking
Archive Configurable Checking
Owner Write, Read, Execute
Group Write, Read, Execute
Root Write, Read, Execute
GID
UID
Target Link

Table 7: File Properties – Linux

FCM_DET_EXP.1.3 Change detection shall be configurable by the **authorized Administrative-user** to occur at the following intervals:

- Real-Time Detection – File changes are immediately identified
- Poll base Detection – File changes evaluated are an Administrative-user configured value in minutes.

[+]Application note: The detection of changed files on Agent installed platforms depends, in part, on Operating System components within the Operational Environment.

6.1.8.2 FCM_REM_EXP.1 Change Remediation (Explicit)

FCM_REM_EXP.1.1 The TSF shall perform the following remediation function(s) on all changes to **locked properties detected** on **Agent servers/Network Devices**:

One of the following, as configured:

a) Restore Mode - Restore changed file to previous version from the CimTrak repository

b) Log Only Mode – no changes made event logged only

c) Update baseline Mode – forward changed file copy to the repository and increment file revision

d) Prompt for Administrator Action Mode – allows an authorized Administrator to accept or reject changes

e) Custom Mode – Administrator configured corrective action.

6.1.9 Class FPT: Protection of the TOE

6.1.9.1 FPT_TST_EXP.1 – TSF Testing (Explicit)

FPT_TST_EXP.1.1 The TSF shall run a suite of self tests during initial start-up to demonstrate the correct operation of **executable code of** the TSF.

6.2 Rationale For TOE Security Requirements

	O. BANNER	O.AUDIT_GEN	O.CHANGE	O.TOE_ACCESS	O.ADMIN_ROLE	O.SELPRO	O.SELF_TEST	O.MANAGE	O.MEDIATE	O.AUDREC	O.AUDIT_PROT	O.IDAUTH	O.CRYPTO	O.ENCRYPT
FAU_ARP.1			X											
FAU_GEN.1		X												
FAU_GEN.2		X												
FAU_SAA.1			X											
FAU_SAR.1										X				
FAU_SAR.3										X				
FAU_STG.1											X			
FAU_STG.3											X			
FCS_CKM.1a,b,c													X	X
FCS_CKM.4													X	
FCS_COP.1a,b,c,d,e													X	X
FDP_ACC.1									X					
FDP_ACF.1									X					
FDP_ITT.1									X					
FDP_ROL.1									X					
FIA_AFL.1				X		X						X		
FIA_ATD.1				X										
FIA_UID.1												X		
FIA_UID.2												X		
FIA_UAU.1												X		
FIA_UAU.2												X		
FIA_SOS.1												X		
FIA_SOS.2												X		
FMT_MOF.1a,b								X						
FMT_MSA.1a,b								X						
FMT_MSA.3								X						
FMT_MTD.1a-e								X						

FMT_SMF.1								X						
FMT_SMR.1					X			X						
FPT_ITT.1														X
FPT_TST_EXP.1						X	X							
FTA_SSL.3						X								
FTA_TAB.1	X					X								
FCM_DET_EXP.1			X											
FCM_REM_EXP.1			X											

Table 8: Mapping Between Security Functions and IT Security Objectives

6.2.1 TOE Security Functional Requirements Rationale

The security requirements are derived according to the general model presented in Part 1 of the Common Criteria. Specifically, Table 8: Mapping Between Security Functions and IT Security Objectives illustrates the mapping between the security requirements and the security objectives and Table 3: Summary of Mappings Between Threats and IT Security demonstrates the relationship between the threats, policies and IT security objectives. The functional and assurance requirements presented in this Security Target are mutually supportive and their combination meets the stated security objectives.

Security Objective	Mapping Rationale
O.BANNER	FTA_TAB.1 specifies that the TOE displays a TOE Access Banner during startup advising user of implications of misuse of TSF resources.
O.AUDIT_GEN	FAU_GEN.1 specifies that the TOE generates audit records for security related events FAU_GEN.2 specifies that audit records generated by the TOE associate each auditable event with the identity of the user that caused the event.
O.CHANGE	FCM_DET_EXP.1 specifies that the TOE detect changes to files configured as "locked" on Agent servers/Network devices FCM_REM_EXP.1 specifies that the TOE executes remediation measures upon detection of a file changes as specified in FCM_DET_EXP.1. FAU_ARP.1 specifies that the TOE takes specified action upon detection of a potential security violation. FAU_SAA.1 specifies that the TOE applies a set of rules in monitoring events and based upon these rules indicate a potential violation of the enforcement of the SFRs
O.TOE_ACCESS	FIA_AFL.1 specifies that the TOE disables a CimTrak account upon an Administrator configurable number of failed authentication attempts.

Security Objective	Mapping Rationale
	FIA_ATD.1 defines the attributes of users, including a user ID that is used by the TOE to determine a user's identity and/or group memberships and enforce what type of access the user has to the TOE.
O.ADMIN_ROLE	The TOE establishes [Administrator] roles. The authorized Administrator (role: Administrator) is given privileges to perform certain tasks that other users will not be able to perform. These privileges include, but are not limited to, access to audit information and security functions. (FMT_SMR.1)
O.SELPRO	FIA_AFL.1 specifies that the TOE disables a CimTrak account upon an Administrator configurable number of failed authentication attempts. FPT_TST_EXP.1 specifies that the TOE runs a self test during startup to verify the integrity of the TSF executable code. FTA_SSL.3 specifies that [Administrator] sessions between the CimTrak Management Console and the Master Repository are terminated by the TSF after 10 minutes of inactivity. FTA_TAB.1 specifies that the TOE displays a TOE Access Banner during startup advising user of implications of misuse of TSF resources.
O.SELF_TEST	FPT_TST_EXP.1 specifies that the TOE runs a self test during startup to verify the integrity of the TSF executable code.
O.MANAGE	FMT_MOF.1a provides that the TOE's management function can only be accessed and utilized (modified) by [Administrators]. FMT_MOF.1b provides that the TOE's management function can only be accessed and utilized (modified) by Administrators, Standard User roles. FMT_MTD.1a specifies the TSF data that can be created, modified or deleted by use of the TOE's management functions by Administrators, Standard User roles. FMT_MTD.1b specifies the TSF data that can be created, modified or delete by use of the TOE's management functions by Administrators. FMT_MTD.1c specifies the TSF data that can be queried by use of the TOE's management functions by Administrators. FMT_MTD.1d specifies the TSF data that can be queried by use of the TOE's management functions by Administrators, Standard User roles. FMT_MTD.1e specifies the TSF data that can be queried by use of the TOE's management functions by Administrators, Standard User, Auditor roles. FMT_SMR.1 defines the roles provided by the TOE. FMT_SMF.1 specifies the management functions supported by the TOE.

Security Objective	Mapping Rationale
	FMT_MSA.1a specifies that the TOE enforces the CimTrak Access Control SFP to restrict the ability to modify the applicable security attributes to the Administrator. FMT_MSA.1b specifies that the TOE enforces the CimTrak Access Control SFP to restrict the ability to modify the applicable security attributes to the Administrator, Standard User roles. FMT_MSA.3 specifies that the TOE enforces the CimTrak Access Control SFP to provide restrictive default values for security attributes that are used to enforce the SFP and that no user can specify alternative default values.
O.MEDIATE	FDP_ACC.1 specifies that TOE enforces the CimTrak Access Control SFP on specified subjects, objects and operations. FDP_ACF.1 specifies the rules enforced by the CimTrak Access Control SFP on subject, object and operations listed in FDP_ACC.1. FDP_ITT.1 specifies that enforces the CimTrak access control SFP to prevent the disclosure, modification of user data when it is transmitted between the Master Repository and Agent TOE components. FDP_ROL.2 specifies that the TOE enforces the CimTrak access control SFP to permit the rollback of changes on the "locked" data stored on applicable Agent servers/Network Host servers based on a configurable number of saved generations of saved data (default 10)
O.AUDREC	FAU_SAR.1 specifies that [Administrators] (Administrator, Standard user, Auditor) are explicitly given the capability to read all audit information from the audit records, and the records are provided in a suitable manner for interpretation. FAU_SAR.3 specifies that audit records may be sorted by Audit Event Time/Date.
O.AUDIT_PROT	FAU_STG.1 ensures that the TOE provides for the storage of audit data in a manner that protects the data from unauthorized deletion and prevents any modification to TOE audit records. FAU_STG.3 specifies that the Master Repository component of the TOE sends an email to the configured email address in the event the audit trail storage exceeds the administrator configured threshold.
O.IDAUTH	FIA_AFL.1 specifies that the TOE disables a user account upon an administrator configurable number of failed authentication attempts. FIA_UID.1 specifies that the TOE for CimTrak Management Console to Master repository sessions allows access to a language selection option, banner and online help prior to identification but requires identification prior to accessing all other aspects of the TSF. FIA_UID.2 specifies that the Master Repository requires identification prior to allowing any access to the Master Repository by the CimTrak Agents.

Security Objective	Mapping Rationale
	FIA_UAU.1 specifies that the TOE for CimTrak Management Console to Master repository sessions allows access to a language selection option, banner and online help prior to authentication but requires authentication prior to accessing all other aspects of the TSF. FIA_UID.2 specifies that the Master Repository requires authentication prior to allowing any access to the Master Repository by the CimTrak Agents. FIA_SOS.1 specifies that the TOE verifies that passwords are verified by the TOE to meet specified requirements for minimum length and complexity. FIA_SOS.2 specifies that the TOE generates secrets using a random number generator for creation of passwords that meet specified criteria.
O.CRYPTO	FCS_CKM.1a specifies that the TOE generates Symmetric cryptographic keys using either 3DES or AES with specified key lengths which adhere to specified standards AES/3DES meets ANSI X9.31, FIPS 186-2, or FIPS SP 800-90 . FCS_CKM.1b specifies that the TOE generates Asymmetric cryptographic keys using Diffie-Hellman, RSA with specified key lengths which adhere to a specified standards and FIPS PUB 140-2. FCS_CKM.1c specifies that the TOE generates cryptographic keys used for the Private Key feature using a passphrase (string) to key function in accordance with the referenced standards. FCS_CKM.4 specifies that the TOE provides Cryptographic key destruction in accordance with FIPS PUB 140-2. FCS_COP.1a specifies that the TOE performs encryption/decryption of user data stored in the Master Repository using AES, 3DES with specified key sizes which conform to the specified standard and FIPS PUB 140-2. FCS_COP.1b specifies that the TOE performs File Comparison via Hashing of user data stored in the Master Repository using SHA1 with specified key sizes which conform to the specified standard and FIPS PUB 140-2. FCS_COP.1c specifies that the TOE performs Agent Session Encryption of data in transit to the Master Repository using AES or 3DES with SHA1 HMAC, RSA with specified key sizes which conform to the specified standards and FIPS PUB 140-2. FCS_COP.1d specifies that the TOE performs Console Session Encryption/Decryption of data in transit to the Master Repository using AES or 3DES (SHA1 HMAC), RSA with specified key sizes which conform to the specified standard and FIPS PUB 140-2. FCS_COP.1e specifies that the TOE performs Network Agent device session Encryption/Decryption of data in transit over SSHv2 using AES or 3DES with specified key sizes which conform to the specified standard.
O.ENCRYPT	FCS_COP.1c specifies that the TOE performs Agent Session Encryption of data in transit to the Master Repository using TLS with AES or 3DES with SHA1 HMAC, RSA with specified key sizes which conform to the specified

Security Objective	Mapping Rationale
	standards and FIPS PUB 140-2. FCS_COP.1d specifies that the TOE performs Console Session Encryption/Decryption of data in transit to the Master Repository using TLS with AES or 3DES (SHA1 HMAC), RSA with specified key sizes which conform to the specified standard and FIPS PUB 140-2. FCS_COP.1e specifies that the TOE performs Network Agent device session Encryption/Decryption of data in transit over SSHv2 using AES or 3DES with specified key sizes which conform to the specified standard. FPT_ITT.1 specifies that the TOE protects TSF data from disclosure or modification when it is transmitted between separate parts of the TOE (Master Repository and management console/Server agents).

6.3 Rationale for Explicitly Stated Security Requirements

The Table below presents the rationale for the inclusion of the explicit requirements found in this Security Target.

Explicit Requirement	Identifier	Rationale
FCM_DET_EXP.1	Change Detection (Explicit)	This requirement is explicitly stated as the SFRs in Part II do not adequately address the ability of the TOE to detect file changes based on configured file attributes & object groups. An SFR does not exist within Part II that describes the ability for a TOE to detect when a change has been made to a specified object that the TOE has explicitly been configured to monitor for change.
FCM_REM_EXP.1	Change Remediation (Explicit)	This requirement is explicitly stated as the SFRs in Part II do not adequately address the ability of the TOE to remediate file changes based on configured rules for specified Agent/File Object group combinations. An SFR does not exist within Part II that describes the ability for a TOE to execute configured remediation actions upon a specified object once a change has been detected.
FPT_TST_EXP.1	TSF Testing (Explicit)	This requirement is explicitly stated as the TOE does not contain the mechanisms for users to verify the integrity of TSF code; only integrity checking during startup operations is performed. FPT_TST.2 and FPT_TST.3 are not satisfied.

Table 9: Explicitly Stated SFR Rationale

6.4 Rationale For IT Security Objectives

	O.ADMIN_GUID	O.CONFIG	O.DOC_DESIGN	O.FUNC_TEST	O.VULN_ANALY
ALC_DEL.1	X				
AGD_OPE.1	X				
ALC_CMC.4		X			
ALC_CMS.4		X			
ALC_FLR.2		X			
ADV_TDS.3			X		
ADV_FSP.4			X		
ATE_COV.2				X	
ATE_FUN.1				X	
ATE_IND.2				X	
AVA_VAN.3					X

Table 10: Security Objectives to Assurance Requirements

6.5 Rationale For IT Security Requirement Dependencies

This section includes a table of all the security functional requirements and their dependencies and a rationale for any dependencies that are not satisfied.

Functional Component	Dependency	Included/Rationale
FAU_ARP.1	FAU_SAA.1	Yes
FAU_GEN.1	FPT_STM.1	No
FAU_GEN.2	FAU_GEN.1, FIA_UID.1	Yes
FAU_SAA.1	FAU_GEN.1	Yes
FAU_SAR.1	FAU_GEN.1	Yes
FAU_SAR.3	FAU_SAR.1	Yes
FAU_STG.1	FAU_GEN.1	Yes
FAU_STG.3	FAU_STG.1	Yes
FCS_CKM.1a,b,c	FCS_COP.1, FCS_CKM.4	Yes

Functional Component	Dependency	Included/Rationale
FCS_CKM.4	FCS_CKM.1	Yes
FCS_COP.1a, b, c, d, e	FCS_CKM.1, FCS_CKM.4	Yes
FDP_ACC.1	FDP_ACF.1	Yes
FDP_ACF.1	FDP_ACC.1, FMT_MSA.3	Yes
FDP_ITT.1	FDP_ACC.1	Yes
FDP_ROL.1	FDP_ACC.1	Yes
FIA_AFL.1	FIA_UAU.1	Yes
FIA_ATD.1	None	None
FIA_UAU.1	FIA_UID.1	Yes, through FIA_UID.1
FIA_UAU.2	FIA_UID.1	Yes, through FIA_UID.2
FIA_UID.1	None	None
FIA_UID.2	None	None
FIA_SOS.1	None	None
FIA_SOS.2	None	Yes
FMT_MOF.1	FMT_SMR.1, FMT_SMF.1	Yes
FMT_MSA.1	FMT_SMR.1, FMT_SMF.1, FDP_ACC.1	Yes
FMT_MSA.3	FMT_MSA.1, FMT_SMR.1	Yes
FMT_MTD.1a,b	FMT_SMR.1, FMT_SMF.1	Yes
FMT_SMF.1	None	None
FMT_SMR.1	FIA_UID.1	Yes (via FIA_UID.2)
FPT_ITT.1	None	None
FPT_TST_EXP.1	None	None
FTA_SSL.3	None	None
FTA_TAB.1	None	None
FCM_REM_EXP.1	FCS_DET_EXP.1	Yes
FCM_DET_EXP.1	None	None

6.6 Rationale For IT Security Requirement Dependencies not satisfied

FPT_STM.1 Time Stamps dependency to FAU_GEN.1

The requirement for FPT_STM.1 Time Stamps is not satisfied by the TOE as the TOE does not have mechanisms for maintaining or managing a time source within the application. The TSF does support the use of Time Stamp in audit records by leveraging the time source mechanism provided through the Agent server/Network Host server Operating System and the Master Repository Operating System. Therefore, this dependency is satisfied through an OE.TIME_STAMPS security objective for the Operational Environment.

6.7 Security Assurance Measures

The assurance measures provided for this Security Target are described in detail in evidence documentation to be provided to the evaluation team during the course of the evaluation of this TOE.

Assurance Class	Assurance components
ADV: Development	ADV_ARC.1 Security architecture description
	ADV_FSP.4 Complete functional specification
	ADV_IMP.1 Implementation representation of the TSF
	ADV_TDS.3 Basic modular design
AGD: Guidance documents	AGD_OPE.1 Operational user guidance
	AGD_PRE.1 Preparative procedures
ALC: Life-cycle support	ALC_CMC.4 Production support, acceptance procedures and automation
	ALC_CMS.4 Problem tracking CM coverage
	ALC_DEL.1 Delivery procedures
	ALC_DVS.1 Identification of security measures
	ALC_FLR.2 Flaw Reporting Procedures
	ALC_LCD.1 Developer defined life-cycle model
	ALC_TAT.1 Well-defined development tools
ASE: Security Target evaluation	ASE_CCL.1 Conformance claims
	ASE_ECD.1 Extended components definition
	ASE_INT.1 ST introduction
	ASE_OBJ.2 Security objectives
	ASE_REQ.2 Derived security requirements
	ASE_SPD.1 Security problem definition

Assurance Class	Assurance components
	ASE_TSS.1 TOE summary specification
ATE: Tests	ATE_COV.2 Analysis of coverage
	ATE_DPT.2 Testing: security enforcing modules
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent testing - sample
AVA: Vulnerability assessment	AVA_VAN.3 Focused vulnerability analysis

Table 11: Security Assurance Measures

6.8 Rationale for Security Assurance

EAL 4 + ALC_FLR.2 was chosen to provide a "enhanced basic" level of independently assured security. The chosen assurance level is consistent with the threat environment. Specifically, that the threat of malicious attacks is not greater than "enhanced basic" and the product will have undergone a search for obvious flaws and a focused vulnerability analysis.

6.9 Rationale for TOE Security Functions

This section provides a table demonstrating the tracing of TOE security functions back to aspects of the security functional requirements (SFRs).

A justification that the security functions are suitable to cover the SFRs can be found in Section 7.1.

	Security Audit	Identification and Authentication	Cryptographic Operations	Access Control	Change Management	Security Management	Secure Communications	Protection of the TOE
FAU_ARP.1	X							X
FAU_GEN.1	X							
FAU_GEN.2	X							
FAU_SAA.1	X							
FAU_SAR.1	X							
FAU_SAR.3	X							
FAU_STG.1	X							

	Security Audit	Identification and Authentication	Cryptographic Operations	Access Control	Change Management	Security Management	Secure Communications	Protection of the TOE
FAU_STG.3	X							
FCS_CKM.1a,b,c			X				X	
FCS_COP.a,b,c,d,e			X				X	
FDP_ACC.1				X				
FDP_ACF.1				X				
FDP_ITT.1				X				
FDP_ROL.1				X				
FIA_AFL.1		X						
FIA_ATD.1		X						
FIA_UID.1		X						
FIA_UID.2		X						
FIA_UAU.1		X						
FIA_UAU.2		X						
FIA_SOS.1		X						
FIA_SOS.2		X						
FMT_MOF.1a,b						X		
FMT_MSA.1a,b						X		
FMT_MSA.3						X		
FMT_MTD.1a-e						X		
FMT_SMF.1						X		
FMT_SMR.1						X		
FPT_ITT.1							X	X
FPT_TST_EXP.1								X
FTA_SSL.3		X						X
FTA_TAB.1								X
FCM_DET_EXP.1					X			
FCM_REM_EXP.1					X			

Table 12: TOE Security Function to SFR Mapping

7 TOE Summary Specification

7.1 TOE Security Functions

The TOE's security functionality is organized by the following 8 Security Functions:

- Security Audit
- Identification and Authentication
- Cryptographic Operations
- Access Control
- Change Management
- Security Management
- Secure Communications
- Protection of the TOE

7.1.1 Security Audit

7.1.1.1 CimTrak Audit Generation – FAU_GEN.1, FAU_GEN.2

The CimTrak TOE generates audit records for security events exceeding the "minimal" level of auditing as defined within FAU_GEN.1 Common Criteria Security Functional Requirements (SFR). Audit events are logged for Agent related events such as detection of a change to a locked file, associated remediation actions and application configuration events executed through the Management Console such as changes to security attributes.

Agent audit events are generated by the Agent Core Subsystem and Security Event Logic as part of the CimTrak Agent software component residing on each Agent server/Network Host server. The CimTrak Agent runs as a service on the Agent server(s)/Network Host server(s) and upon detection of a change to any of the locked file attributes (listed in Table 6: File Properties – Windows or Table 7: File Properties – Linux) or locked Network device configuration attributes, the Agent Core Subsystem calls the Security Event Subsystem which records the change event, remediation action taken and outcome. The Agent Core Subsystem then passes this information over secure channels to the Master Repository machine using the CimTrak communication protocol. The Event Notification subsystem leverages the Cryptographic Subsystem to create a SHA1 hash of the record and stores the event (associated with the time/date stamp and Agent ID) into the PostgreSQL database. The SHA1 hash value is verified during startup to confirm the integrity of audit records.

The CimTrak Master Repository may also be configured to route audit records to external resources via SNMP (for administrative messaging), SMTP (email notification), the WebTrends

Extended Logging Format (for use with the WebTrends firewall reporting tool), and Syslog for external storage of audit log records.

Administrative audit events are generated by the Event Notification Subsystem within the CimTrak application software loaded on the Master Repository machine. For logging of authentication related log events such as failed login attempts, the authentication subsystem calls the Event Notification subsystem which generates the event log, generates a SHA1 hash and stores it in the PostgreSQL database. [Administrator] application configuration and management activities are captured by the Client Message Processing subsystem (which passes objects to/from the management console) and the Authentication subsystem which detects authentication related events from the management console such as failed logins and access attempts. The Event Notification subsystem transfers the log data, including the User ID of the user causing the event, to the PostgreSQL database subsystem for storage.

Agent Logs (stored on Master Repository)	Master Repository (Server) Logs
Event Date/Time	Event Date/Time
Event (description)	Event
Absolute Path	Correction
Completion Date/Time	Performed by
Event Code	Modified by
Path	Absolute Path
	Completion Date/Time
	Event Code
	Path

Table 13: Audit Log detail by type

7.1.1.2 Review of Audit Logs – FAU_SAR.1, FAU_SAR.3

The CimTrak TOE provides a resource for the viewing of audit records through the CimTrak Management Console interface to the Master Repository machine. The CimTrak Management Console renders a full function GUI interface which provides the [Administrator] with the resources to configure and manage the application. In addition, the GUI provides viewing and report generation tools which allow for the viewing of CimTrak audit records and event reports.

The Client Core Subsystem, within the CimTrak management console component, provides the GUI interface utilizing a Visual Basic based (VB.net) object-oriented security management application. CimTrak uses an Active X component, written in C^{++}, to provide the data object transfers between the security management application and the PostgreSQL and CimTrak databases within the Master Repository. The CimTrak Management console enforces access controls for [Administrator] sessions as described in Section 7.1.2, Identification and Authentication.

Audit logs are provided for [Administrators] review in a columnar table which allows the selection of the CimTrak Server, CimTrak Agent or Object Group as highlighted in the tree view as part of the left side navigation pane. Audit log records are presented by "Event Date/Time", "Event" (type), "Correction" (remediation action taken), and "Performed by" (Agent/User ID). Upon clicking on the Event Date/Time column header, the contents in the table are sorted by the date and time of the listed events. By design, no user may modify audit logs in any manner and audit logs are integrity checked during startup to assure audit log data is intact.

7.1.1.3 Monitoring of Audited Events – FAU_ARP.1, FAU_SAA.1

The Master Repository monitors specified security events based on configured rules and, upon matching of the applicable rule, triggers a series of actions to mitigate the effect of the event on the TSF, capture the changes for later review and/or notify the configured Administrative User. Included in this rule set is:

a. Accumulation or combination of unauthorized or unexpected changes to locked files on Agent servers/Network devices known to indicate a potential security violation

b. The Agent Core Subsystem fails to contact the Repository Machine (heartbeat subsystem) at configured intervals to verify its online status

c. Failure of Master Repository integrity checking of audit records

d. Exceeding the Administrator, Standard User configured threshold for Master Repository disk space

e. Exceeding the Administrator, Standard User configured threshold for "Monitor Parameters" (CPU, Memory, Disk Space)

Upon detection of an unauthorized change on a given Agent server/Network device (a), an audit event is generated by the Agent server/Network Host server Security Event Processing Subsystem and is passed to the Master Repository: Event Notification subsystem which stores the audit record in the PostgreSQL database. In addition, the Agent Security Event Logic subsystem will immediately implement the configured action including one or more items from the list detailed in SFR FCM_REM_EXP.1 Change Remediation and generate an email message to the configured email address. For violation of the audit storage parameter or failure of an Agent to contact the Repository machine (item b, c, d above), an email is generated by the Master Repository: Event Notification subsystem. Agent related audit event actions are implemented by the Security Event Logic and Agent Core subsystems within the applicable CimTrak Agent. The Agent Security Event Logic subsystem communicates with the Security Event Processing within the Master Repository machine for transfer of User Data to the Master Repository (where applicable). The Agent: Security Event Processing subsystem also coordinates notification of the event via email, acquisition of TSF data related to the Event and, via the Event Notification subsystem, transfer of audit data to the PostgreSQL database.

7.1.1.4 Protection of Audit Logs – FAU_STG.1, FAU_STG.3

The CimTrak TOE stores audit logs on the CimTrak Master Repository machine within the PostgreSQL database subsystem. Upon exhaustion of a configured threshold of the available Master Repository storage resources, an email notification is sent to the configured email address so the recipient may take steps to prevent audit log data loss due to exhaustion of allocated storage space. The threshold for this function can be set by Administrator or Standard User roles. In addition, a pop-up window appears on the CimTrak Management console indicating the event. The Administrator role may configure the email notification trigger based on percent of Master Repository disk storage space used as well as various other status parameters monitored by the CimTrak application. Only Master Repository disk space is included as a configured monitored value for the CC evaluated configuration. The Event Notification subsystem, in conjunction with the underlying OS, monitors audit space usage and triggers the event notification when applicable.

Audit logs may not be modified based on the implementation approach of the PostgreSQL database within the Master Repository CimTrak application. Integrity checks are performed on audit logs during the Master Repository machine startup process, utilizing the Cryptographic Subsystem, to verify that audit log data remains intact and uncorrupted. If integrity errors are encountered during Master Repository startup, an audit event is generated and an email notification is sent by the Master Repository as noted above. Any log items that fail the SHA1 hash verification step during startup are highlighted in red for easy identification by the CimTrak [Administrator].

The CimTrak Master Repository may also be configured optionally to offload audit records to external servers in the Operational Environment.

7.1.2 Identification and Authentication

7.1.2.1 Identification & Authentication – [Administrator]/Agent access – FIA_UID.1, FIA_UID.2, FIA_UAU.1, FIA_UAU.2, FIA_ATD.1

The CimTrak application requires that [Administrators] and Agents communicating with the Master Repository are identified and authenticated prior to accessing the TSF. The TOE maintains Username, Password, User Group/Role* assignment as User based security attributes. Following a successful Diffie-Hellman or RSA based key exchange and TLS session establishment, the Agent must present a valid ID/Password combination in order to access the Master Repository. The CimTrak Management Console login screen allows [Administrators] to access a “Language preference” pull down menu and an “About” screen (which provides version details of the application) prior to Identification or Authentication of the user. All other functions require Identification and Authentication prior to accessing Master Repository (TSF) resources. For the purposes of identification and authentication to the CimTrak application, the security attributes: Username, Password, and (assigned) User Group/Roles are managed for each [Administrator] by CimTrak. These security attributes are stored in the PostgreSQL database in

the form of a one-way SHA1 hash for use in validating login credentials for management console sessions.

*note: user group and role are synonymous since CimTrak uses User Groups to create/enforce roles and manage privileges within the application.

7.1.2.2 Network Host Authentication (Network Agent) to Network Devices - FIA_UID.2, FIA_UAU.2

Network devices monitored by CimTrak Agents running on Network Host servers are accessed using the applicable username/password credentials required to access configuration files for those network devices. This authentication information is entered during initial installation and is saved in the Master Repository in encrypted form. Each time the Network Host (Agent) needs to poll the Network device to verify locked objects, the authentication data is sent from the Master Repository to the Network Host (Agent) and is used by the Network Agent to access the device. Authentication data is neither stored nor cache on the Network Host machine. Secure communication techniques between the Network Host (Agent) and the Master Repository are identical to that of Filesystem Agents.

7.1.2.3 Agent authentication to Master Repository – FIA_UID.2, FIA_UAU.2

The CimTrak Agent communicates to the Master Repository using an encrypted channel over the CimTrak secure communication protocol. Agents must contact the Master Repository when requesting a session and the Master Repository presents an X.509 certificate to the requesting Agent for validation prior to initiating the key negotiation to establish the TLS based session.

Following validation of the Master Repository (server) certificate, the Agent executes a Diffie-Hellman or RSA based key exchange and establishes the secure channel with the Master Repository via a TLS encrypted session. (See Section 7.1.7 Secure Communications.) Once the secure session is established, the CimTrak Agent: Authentication Subsystem forwards the Agent Identifier and password to the Master Repository: Authentication Subsystem for validation. The Agent password is hashed by the Master Repository: Cryptographic subsystem and is then compared to stored values in the PostgreSQL database for validation. The password used to authenticate the Agent server/Network Host server to the Master Repository is created during initial installation. A random password is generated using the Cryptographic Subsystem RNG within the CimTrak Agent component and is saved to the PostgreSQL database in the Master Repository without [Administrator] intervention. By design, the password generated during Agent installation is not visible or known to any human user and is only maintained within the applicable Agent software, as a one-way hash, and in an encrypted form within the PostgreSQL component of the Master Repository. All communications between the CimTrak Agent and the Master Repository requires the establishment of the TLS based secure session, identification to a valid CimTrak Agent ID and validation of password credentials.

<u>[Administrator] authentication to the Master Repository</u>

The [Administrator] initiates a login with the Master Repository through the CimTrak Management Console. Upon launching the CimTrak management console application, the

Management Console initiates the TLS based session, executes a Diffie-Hellman or RSA based key exchange and establishes the symmetric key encrypted session using TLS, as in the Agent description above. Following establishment of the TLS session, the TOE Banner is displayed and the CimTrak management console login screen is presented. The [Administrator] may access on this screen the language preference pull down menu and the "About" screen (which displays CimTrak application version information) prior to entering login credentials. All other TSF access requires Identification and Authentication of the [Administrator]. The [Administrator] enters the applicable username and password and the Authentication subsystem, within the CimTrak Management Console component, passes these credentials to the Master Repository: Authentication Subsystem. The credentials are validated and a validated role is returned, enabling an administrative session and controlling access to data objects based on role.

7.1.2.4 TSF Initiated [Administrator] session termination – FTA_SSL.3

The CimTrak management console [Administrator] session will terminate after an Administrator configured time of inactivity to limit the possibility that an open session is available to an unauthorized user. The Common Criteria evaluated configuration specifies that this value should be set to 10 minutes by default. This requires that a [Administrator] re-authenticate to the Master Repository after a session has been terminated by the TSF. This function is managed by the Master Repository, which detects when a session becomes inactive and closes the connection with the CimTrak Management Console upon reaching the configured session timeout value. The Master Repository sends a 1 minute timeout warning to the Management Console to allow the continuation of the current session upon user intervention (accepting the dialog query); otherwise the session is timed out.

7.1.2.5 Authentication Failure Handling – FIA_AFL.1

The CimTrak Management Console supports disabling a user account upon reaching a configured threshold of successive failed logins. If the [Administrator] attempts to initiate a management session with the Master Repository and provides an invalid username/password combination, an audit log entry is generated; a failed login response is presented to the user and counter increments the failed login attempt. Upon reaching a configured number of failed logins, the applicable account is deactivated prohibiting access for that account through the CimTrak management console. Only the Administrator (role) may re-enable this account once de-activated. For the Common Criteria Evaluated configuration this value is set to disable the account after 10 consecutive login failures. This functionality is useful to thwart brute force attacks through the management console and applies only to Standard Users, Auditor roles. The omission of the Administrator role from this lockout policy prevents the "root" Administrator role from being locked out of the application due to failed logins.

The counter is reset upon a successful login for that particular account. The username entered during the login attempt determines which account the failed logins apply to.

7.1.2.6 TSF enforcement of password policies– FIA_SOS.1

The CimTrak TOE enforces password policies for use in authenticating [Administrators] and Agent servers/Network Host servers. This allows the Administrator to configure one of 4 options for password enforcement:

- No requirements enforced
- CimTrak Password Policy enforcement based with a user-entered password
- CimTrak Password Policy enforcement with a randomly-generated password
- Custom Password settings

The Common Criteria evaluated configuration requires that the above properties enforce the following password policy as a minimum: CimTrak [Administrator] and Agent passwords must include 10 characters minimum, 2 lowercase, 2 uppercase, 2 numbers, 2 special characters, verified not to include dictionary terms, verified not to match previous 4 passwords. (CimTrak Password Policy.)

7.1.2.7 TSF generation of password credentials– FIA_SOS.2

During initial Master Repository configuration, the Administrator selects the minimum password strength requirements which include an option to require that all passwords used by the TOE are be generated using a RNG within the CimTrak Cryptographic subsystem. When this option is selected, the TSF will generate a password that meets the specified password policy requirements (noted above in FIA_SOS.1) and display the resultant password to the Administrator through a pop up window.

Based on this option, the CimTrak Agent component also generates an RNG derived alphanumeric password during initial installation for the purpose of authenticating that Agent to the Master Repository machine. The password strength requirements are acquired by the Agent from the Master Repository during the final step of Agent installation. The Agent authentication subsystem calls the cryptographic subsystem, which uses a software random number generator, to create a pseudo-random alphanumeric password. This password is only known to the Agent component and is passed to the PostgreSQL database within the Master Repository where it is stored as a one-way hash for use in authenticating Agent sessions.

7.1.3 Cryptographic Operations

The CimTrak application utilizes cryptographic subsystems in the CimTrak Agent, CimTrak Master Repository and CimTrak Management Console components to provide cryptographic key generation and encryption/decryption operations on user data and communication sessions. The FIPS 140-2 Level 2 validated, Cimcor Cryptographic Module is used within all cryptographic subsystems employed by CimTrak to provide cryptographic functionality.

7.1.3.1 Key Generation – FCS_CKM.1a, b, c

The CimTrak application utilizes Symmetric Keys in encrypting/decrypting communication sessions over the TLS secure communications protocol. This includes sessions between deployed CimTrak Agent servers/Network Host servers and/or CimTrak Management Console sessions and the Master Repository. Symmetric cryptographic keys are also used for the purpose of securing user data at rest within the Master Repository: CimTrak database through a default CimTrak symmetric key and optionally encrypted with an additional Administrator selected passphrase used to create a "Private Key". This "Private Key" is applied on either an Agent basis where all Agent data is encrypted with the assigned "Private Key" or on an Object basis where a "Private Key" can be assigned to a specific database object within the Master Repository. Symmetric keys are generated through a random number generator using either the AES algorithm with configurable key sizes: 128, 192, 256 bit or the 3DES with a key size of 192 bits. All symmetric key generation conforms to FIPS PUB 140-2 through the FIPS 140-2 Level 2 validated Cimcor Cryptographic Module (FIPS 140-2 Certificate #1315).

Asymmetric keys are generated for Console session encryption; TLS based Agent to Master Repository sessions and X.509 Master Repository certificate creation using Diffie-Hellman and RSA algorithms.

Master keys used for data at rest encrypted are generated through the use of this subsystem on the Master Repository during initial installation and configuration processes.

Symmetric cryptographic keys are used to encrypt/decrypt data during secure TLS sessions Asymmetric keys are generated through a software RNG within the respective cryptographic subsystems using Diffie-Hellman or RSA algorithms and configurable key sizes of 512, 1024 or 2048.

The "Private" key feature creates keys when the applicable Administrator selects the "Private Key" option either during Agent installation or when viewing Master Repository database objects using the Management Console. Once selected, this option requires the entry of a passphrase which is then forwarded from the Agent (Agent based) or the Management Console (Object based) and is presented to the Master Repository. The passphrase is passed from the Agent to the Security Event Processing Subsystem which then calls the Master Repository Core subsystem to initiate the Key Generation process. The Master Repository Core subsystem sends the passphrase to the Cryptographic Module which uses it to generate a symmetric key using the Password-Based Key Derivation Function (PBKDF2). This creates a derived symmetric key from a passphrase using HMAC-SHA-1, which is used to Encrypt/Decrypt the applicable data. Once this key is created, it is encrypted with the default (CimTrak) Master Repository key and is stored in the PostgreSQL database in the same manner as the default CimTrak key.

The Algorithm and Key size configured by the Administrator during Master Repository installation is used by default for this encryption/decryption operation (i.e., either AES or 3DES) in the same manner as where the default Master Repository symmetric key is used.

Configuration of algorithms and keys sizes for key generation is managed through the CimTrak management console GUI through pull down menus which only allow permitted FIPS 140-2 validation algorithm/key combinations.

7.1.3.2 Cryptographic Key Destruction – FCS_CKM.4

Cryptographic keys are destroyed by the Cryptographic subsystem within the Master Repository, CimTrak Agent or CimTrak Management Console through zeroization (overwrite with random data) of the memory locations containing key material and reallocation of the memory following zeroization processes. A single pass overwrite with random data is executed by the Cimcor Cryptographic Module.

Session keys are zeroized when sessions are closed between the Master Repository and CimTrak Agent or CimTrak management console.

7.1.3.3 Cryptographic Operations – FCS_COP.1a, b, c, d, e

Repository Encryption – FCS_COP.1a

The CimTrak application stores an authoritative copy of User Data within the Master Repository for the purpose of preserving a master copy of protected (locked) user data and to allow the immediate overwriting of unauthorized changes to designated locked files.

The CimTrak Master Repository: Cryptographic Subsystem utilizes symmetric key encryption to secure User Data stored within the CimTrak database using either the 3DES or AES based on Administrator configuration as noted above. As the applicable Agent Core subsystem passes user data to the Client Messaging Processing subsystem for storage in the CimTrak database, calls are made to the Cryptographic Subsystem within the Master Repository to symmetrically encrypt the user data. Following encryption, the data is stored within the CimTrak database with the associated Agent server/Network device (identifying) attributes. All user data stored within the CimTrak database is encrypted with the default symmetric key and may also be encrypted with an Administrator assigned "Private Key" which adds a second layer of encryption. This "Private Key" encryption step precedes encryption using the default key but follows the same approach as described above.

File Comparison (Hashing) – FCS_COP.1b

The CimTrak Master Repository component creates a message digest of user data stored in the CimTrak database for the purpose of integrity checking. As an item is stored in the CimTrak database, a hashing operation is performed by the cryptographic subsystem using the SHA1 algorithm. This hashing operation results in a stored value that is used by the TSF to verify that user data stored within the CimTrak database (as the authoritative copy) is intact and unchanged. This value is checked during a sync operation after an Agent has connected to the Master Repository. Hash verification is also performed on CimTrak audit logs stored within the PostgreSQL database for integrity checking of logs when a log read request is made. If changes have been made to a log record, those entries are displayed in red when displayed to the requested user.

Agent Session Encryption – FCS_COP.1c

Sessions between CimTrak Agents and the CimTrak Master Repository machine are secured through a symmetrically encrypted session using TLS. These sessions must be initiated by the CimTrak Agent which initiates a Diffie-Hellman or RSA key exchange and then

encryption/decryption of session traffic. Encryption processes use either the 3DES or AES algorithm along with key sizes as noted above under "Key Generation – FCS_CKM.1a, b, c". Secure TLS sessions established also include HMAC integrity checking to validate transmissions during the session utilizing SHA1. The resident Agent Cryptographic Subsystem and CimTrak Communication Protocol Subsystem are invoked by the Agent Core subsystem to initiate the session, conduct the key exchange/session encryption and communicate using the CimTrak communication protocol.

Where supported by Network devices, the above algorithm and key size combinations are applicable for use in establishing secure sessions between the Network Agent and Network devices using the SSHv2 protocol.

The session described above between the Network Agent and the CimTrak Master Repository also applies to SSHv2 sessions between the Network device and the Network Agent running on the Network Host server. The session between the Network Device and the Network Agent uses OpenSSH. The OpenSSH implementation uses the Cimcor Cryptographic Module within the Network Agent Cryptographic Module subsystem.

The following cipher suites are supported for TLS sessions by CimTrak implementing the FIPS 140-2 validated Cimcor Cryptographic Module:

DHE-RSA-AES256-SHA

AES256-SHA

DHE-RSA-AES128-SHA

AES128-SHA

EDH-RSA-DES-CBC3-SHA

DES-CBC3-SHA

CimTrak Management Console Session Encryption – FCS_COP.1d

Sessions between the CimTrak Management Console machine and the Master Repository are conducted over TLS sessions as describe above for Agent to Master Repository sessions. These sessions leverage the cryptographic subsystem and CimTrak Communication protocol subsystem resident on the Management Console machine. AES or 3DES algorithms may be specified for console sessions along with key sizes are described in FCS_CKM.1a. Traffic sent between the Management Console and the Master Repository is validated using an SHA1 HMAC as enforced by the CimTrak implementation of TLS. Cipher suites available for use are the same as those listed above for Agent – Master Repository sessions.

Network Agent device sessions - FCS_COP.1e

Sessions between the Network Agent (Host) machine and applicable network devices managed by CimTrak may be encrypted using SSHv2. The LibSSH "C" libraries are called by the Network Device Interaction Module running on the Network Host computer to establish a secure tunnel between the Network Agent and the targeted network device. As with the TLS based sessions described above, these SSHv2 sessions utilize either the AES or 3DES encryption algorithm and keys sizes as defined in FCS_COP.1e.

7.1.4 Access Control

The Access Control security function for CimTrak provides the technical means for the protection of User Data and enforcement of access controls for that data, within either the Agent server/Network device or as stored in the CimTrak Master Repository. The CimTrak Access Control SFP defines the rules for accessing User Data in support of the essential change management functionality provided by the software. The context of the application of access controls on Agent server/Network device user data is limited to those objects which have been "locked" or selected for protection by the application. CimTrak monitors these items and therefore can institute access and change controls on them; however, items not identified by the [Administrator] as being locked or protected are not impacted by the CimTrak access control SFP. All access through the CimTrak management console is subject to the CimTrak Access Control SFP.

7.1.4.1 CimTrak Access Control – FDP_ACC.1, FDP_ACF.1, FDP_ROL.1

Access to User Data designed as locked on CimTrak Agent servers/Network devices is managed by the particular settings established by the [Administrator] for that object or by object groups. A locked object on any CimTrak Agent server/Network device may only be changed (write/delete operations) if that object is configured for one of the following remediation options: "Log Only" (which simply logs the change and allows it to proceed), "Update Baseline" (which allows the change and also exports the changed object to the Master Repository to be saved), or "Prompt" (which allows the [Administrators] to determine whether to allow the change in "real time"*).

*"real time" refers to the ability to detect file object changes as they are made vs. the "polling" method which detects file changes at periodic time intervals.

The "Restore from Repository" setting explicitly disallows the change by immediately overwriting it with the authoritative copy of the object from the CimTrak database in the Master Repository. This has a similar effect as restricting write access to an object since as soon as the change is detected it is immediately overwritten with the baseline object as specified in the Basic Rollback FDP_ROL.1. This provides the core protection measures of CimTrak for critical files or objects designated as "locked" on Agent server/Network devices.

These access controls are implemented on the Agent servers/Network devices by the CimTrak Agent service running on the Agent server or Network Host operating system. For Filesystem Agents, the underlying OS detects when a file change is attempted and that flag is detected by the Security Event logic subsystem. The Security Event logic subsystem then implements the applicable remediation action based on the security attributes loaded into the CimTrak Agent Security Event Logic during startup. Network Agents operate in a similar fashion; however, the Network Device Interaction module employs a poll monitor that polls the configuration status of the applicable Network devices and performs a hash value comparison between the current configuration and the stored hash value within the Master Repository. When the values don't match, a change is detected and the Security Event logic subsystem on board the Network Host implements the configured remediation action as noted above.

User Data stored within the Master Repository can be accessed by CimTrak Agents (not directly but through requests issued to the Master Repository machine) when the Restore from Repository setting has been configured for that file or object group. This allows the immediate reversal of unauthorized changes. The CimTrak Agent Security Event Logic working with the Agent Core subsystem contacts the Master Repository machine and requests the applicable object be forwarded for restoration on the Agent server/Network device. These requests are received by the Master Repository Security Event Processing subsystem which accesses the CimTrak database to immediately dispatch the needed file to the requesting Agent server/Network Host server.

[Administrators] may also access User Data stored on the Master Repository once properly authenticated as described in Section 7.1.2, Identification and Authentication. This data is available to [Administrators] to allow inspection of changes made to "locked" files through the CimTrak Management Console GUI. Multiple generations of objects are maintained to allow for later inspection by [Administrators] within the CimTrak database housed in the Master Repository. [Administrators] include the roles: Administrator, Standard User and Auditor.

User data protected using the "Private Key" feature requires the Administrator attempting to access the data to enter the passphrase associated with the particular Agent or Object "Private Key" passphrase used. CimTrak prompts the user to enter the passphrase via the Management Console and the passphrase is passed to the Master Repository where it is used by the Cryptographic Module to generate a symmetric key. The generated key is then compared with the key stored within the PostgreSQL database within the Master Repository. If the value matches, the data is decrypted and sent to the Management Console where the user can view the contents. If a "Private Key" is assigned on an Agent basis (during Agent installation) all Object Groups within that Agent inherit the assigned Agent "Private Key"; however, this key assignment can be overridden by assigning a different "Private Key" on an Object basis. Any Master Repository database Object can be protected by a "Private Key" assigned by the [Administrator] through the Management Console. There is no backup storage or recovery available for "Private Keys" assigned through this feature. For a "Private Key" protected item, the applicable passphrase must be entered in order to create, view, modify or compare Objects within the CimTrak Master Repository. All CimTrak [Administrators] can still lock, unlock, add and delete folders in terms of their "watched" status, even if "Private Key" is encrypted but they cannot access the User Data within those folders.

7.1.4.2 User Data Protection within the Master Repository – FAU_STG.3

User Data is stored in the CimTrak database within the Master Repository machine. This database is a proprietary database, written as an extension of PostgreSQL, which is designed to allow a broad range of data types to allow User Data to be stored in its original format.

CimTrak Master Repository storage resources are monitored to assure that allocated space is not exhausted leading to potential loss of User Data. Upon reaching a configured value of space usage, an email notification is sent to the CimTrak configured email address for [Administrator] notification. This is the same Master Repository storage resource monitoring referenced above for the protection of security audit records.

7.1.4.3 User Data protection during transfer – FDP_ITT.1

User data transfers which occur between CimTrak Agent servers/Network Host servers and the Master Repository Machine or the CimTrak Management Console and the Master Repository Machine are secured via the symmetrically encrypted TLS based sessions. The CimTrak implementation of TLS assures that transmissions are not disclosed or modified in transit by requiring that sessions are conducted using symmetric key cryptography implemented in accordance with FIPS 140-2 Level 2 requirements. Additionally, transmission integrity checks are performed as part of the TLS protocol using a SHA1 HMAC. Additional information regarding user data protection during transmission is contained in Section 7.1.3, Cryptographic Operations and Section 7.1.7, Secure Communications. This functionality is provided through the Cryptographic Subsystem contained within Agent servers/Network Host servers, the CimTrak Management Console, and the Master Repository.

7.1.5 Change Management

The CimTrak TOE's core functionally relates to the protection and change management of files or objects on Agent servers/Network devices deployed within the internal network environment. Through "real time"* or poll based monitoring of selected file objects, the CimTrak Agent running as a service on the underlying or host Server OS, detects file/object change attempts and immediately applies remediation action based on [Administrator] configured rules. As noted above in Section 7.1.4 Access Control, remediation measures may include overwriting of unauthorized file/object changes with an authoritative copy stored within the Master Repository. In addition, since previous versions of locked files may be compiled within the Master Repository, the [Administrator] may revert to previous file/object versions by pushing file changes from the Master Repository machine to Agent servers/Network devices in the deployed environment.

*"real time" refers to the ability to detect file object changes as they are made vs. the "polling" method which detects file changes at periodic time intervals.

7.1.5.1 Change Detection – FCM_DET_EXP.1

Change detection is established through security attribute settings made by the [Administrator] for each targeted Agent server/Network device. The applicable Agent server/Network device is selected from the CimTrak management console and the [Administrator] can directly select objects to protect (lock) or may lock multiple objects by using pre-established object groups. These security attributes are stored within the PostgreSQL database component within the Master Repository and are passed over secure channels to the applicable Server Agent/Network host agent during startup initialization. Once received by the Agent component, the attributes are passed to the Security Event logic subsystem.

During the Agent startup, the CimTrak service starts and the Agent server or Network host Operating System+ monitors files designated through CimTrak as locked for file changes. Filesystem Agents rely on the underlying Operating System mechanisms to identify potential changes whereas Network Agents utilize a polling monitor within the NDIM subsystem which

performs a scanning read function on network device configuration data. A single Network host/Network Agent combination can support multiple network devices, such as routers, as each device is managed in an exclusive object group by the Master Repository. The dedicated object group allows specific configuration settings and connection instructions to be applied for each network device. These multiple network devices are simultaneously monitored by the polling monitor through the use of multiple threads running concurrently for each network device with a CimTrak locked configuration.

The Event Detection Method used for monitoring of file/object changes may be based on either real-time or poll based monitoring. Real-time monitoring constantly scans for file attribute changes and immediately takes action whereas poll based monitoring provides monitoring scans at minute intervals as configured within the CimTrak application. The Event Detection Method is configured as a Watch Property for each Object Group defined.

In addition to the locked file object monitoring, Agents also monitor parameters on the installed Agent platform that allow [Administrators] to view Network utilization, CPU usage, Fixed Memory usage (RAM) and available Disk space. These "Monitoring Parameters" report these values in percent (%) as part of the Heartbeat monitoring of installed Agents within the network. The Agent Monitor Parameters window also allows thresholds to be set that trigger an alarm when a specified condition exceeds the configured threshold for a specified period of time. Once the alarm has been triggered, the event is placed in the CimTrak log.

The configurable parameters that create the thresholds for creating an alarm event include the following categories and value ranges:

- Devices (Network Connections (utilization), CPUs, Disk Space, Memory Use (RAM))

  For each device, the following conditions, percentages, and time interval in seconds are evaluated:

- Conditions (Equal, Greater than, Greater than or equal, Less than, Less than or equal)
- Percentage (0% through 100%)
- After (sec) (1 through 172800)

  Examples: Settings = CPU, Greater than, 80%, After 300 seconds, therefore, an alarm is generated if the CPU usage for the Agent machine exceeds 80% for 5 minutes. Settings=Network Connections, Greater than, 40%, after 300 seconds, therefore, an alarm is generated if the Network utilization exceeds 40% (of max) for 5 minutes.

For Filesystem Agents, once a change to an object is identified by the Operating System, the Security Event logic component evaluates the change attempt and determines if a locked file/object was affected. In the event a locked file has changed, the Security Event logic immediately implements the remediation action indicated based on the security attributes loaded in memory during startup. The criteria used to determined if a change has occurred are the file properties contained in Table 6: File Properties – Windows and Table 7: File Properties – Linux. If any of the listed properties is detected to have been changed, then the remediation action is triggered.

Windows Filesystem Agents also provide "Forensic Data" for locked objects inside a shared Windows directory. In the event of a possible intrusion, a list of possible IP Addresses/PC

Identifiers is displayed within the Forensics Data window in the CimTrak Management Console GUI. This allows [Administrators] to investigate the source of the intrusion attempt using the provided IP Addresses.

Network Agents execute a polling sequence from the Network host machine that reads configuration data from applicable network devices. The NDIM subsystem working with the Agent Core module then calls the cryptographic module which hashes the data set and the NDIM compares it with stored values obtained from the Master Repository. In the event the file hashes do not match, a change is presumed and remediation action is taken by the Agent Security Event logic subsystem.

Following a detected change event, the Agent Security Event logic component also collects details about the event as well as the actual User Data changes made for storage in the Master Repository. This information is used to produce detailed reports which allow [Administrators] to identify when the changes were made, who initiated the changes and the exact content of the change. The following information is collected by the Security Event Logic subsystem component of the Agent for each change event and is passed to the Master Repository for storage:

- Date and time of the event, type of event, subject identity, and the outcome of the event.
- The actual file/configuration properties changed within the file.
- Connection type which resulted in the file change.
- Action taken by the TSF upon detection of the event.

[+]Application note: The detection of changed files on Agent installed platforms depends, in part, on Operating System components within the Operational Environment.

7.1.5.2 Change Remediation – FCM_REM_EXP.1

Upon detection of a change to a locked file/object the Security Event Logic subsystem within the Agent server/Network Host server initiates remediation action based on security attributes configured to that particular object or object group. Filesystem Agents may apply remediation directly to the file on the installed platform's hard disk whereas the Network Agent remotely connects to the applicable network device and implements a remote configuration change as applicable.

Log Only mode:

If the file/object has been configured in Log Only mode, the Security Event logic passes information regarding the change including the date/time of the event, type of event, subject identity and the outcome to the Security Event Processing subsystem within Master Repository. The audit data is then passed to the Security Event processing subsystem within the Master Repository and in turn is processed to the PostgreSQL database for storage and retrieval. Noteworthy about this remediation mode is that it only logs the event and may also update the saved baseline stored within the CimTrak database if so configured. To save the file change into the Master Repository, the Security Event logic component of the Agent uploads a copy of the changed file/object into memory and passes this information to the Security Event Processing

subsystem of the Master Repository which stores it in the applicable database structure. The Log Only Mode option is typically used initially during the deployed to evaluate typical file object change patterns during normal use.

Restore mode:

In Restore mode, the Security Event Logic subsystem within the Agent component immediately contacts the Master Repository Security Event Processing subsystem, using the CimTrak secure communication protocol, with a request for the baseline file saved within the Master Repository for that particular Agent server/Network device. The Security Event Processing subsystem passes this request to the CimTrak database, acquires the baseline file and forwards the file to the Agent server/Network device to restore the changed file with the baseline. In parallel, a copy of the file object change is uploaded into memory and this information, along with audit details, are passed to the Master Repository. The copy of the unauthorized file change is stored in the CimTrak database (intrusion file) and the audit details are passed to the PostgreSQL database.

Update Baseline mode:

The Update Baseline mode performs all the operations listed in the log only mode, and allows the change to be sustained but forwards a new baseline copy of the changed file/object to the Master Repository for storage within the CimTrak database and increments the managed revision of the file. This manages the file change as an authorized change and updates the revision history. This allows for multiple changes to be logged and maintained through CimTrak with a thorough revision history and actual copies of the files stored in the CimTrak proprietary (user data) database. The [Administrator] can review changes at a later date and revert to a previous version of the file/object, from the CimTrak management console, if an undesired change is detected at a later date. To facilitate the "Update Baseline" option, the Security Event logic component of the Agent uploads a copy of the changed file/object into memory along with audit log data and passes this information, using the CimTrak secure communication protocol, to the Security Event Processing subsystem of the Master Repository. Audit data is then stored within the PostgreSQL database and the new baseline is stored in the CimTrak database.

Prompt for Administrator Action mode:

The Prompt for Administrator Action mode is selected to allow the Agent Security Event Logic subsystem to store the event for later review and upon a request made by the CimTrak Management Console Client; the Master Repository accesses the stored information and notifies the Management Console application. When this occurs, the object "change status" and associated information is displayed on the Management Console along with options so the [Administrator] may select the appropriate action for immediate implementation. The action selection is passed to the Master Repository which routes the action decision to the Agent Security Event Logic for local implementation. Regardless of the selection made, the event is logged within an audit record. If the decision is to allow a new baseline, the steps listed above under the Update Baseline mode apply.

Custom mode:

The Custom mode remediation setting allows [Administrators] to configure remediation to take specific corrective actions based on different types of intrusion events. This mode allows action

options to be separately configured (using the standard remediation actions above) based on whether the locked file was changed, deleted or a new file was added to a locked directory.

7.1.6 Security Management

The CimTrak TOE provides security management functions through a full featured GUI, hosted from the CimTrak management console, and running on a dedicated workstation in the Operational Environment. Administrator and Standard User roles may configure and manage the CimTrak application exclusively through this interface following initial installation, including [Administrator] account management, CimTrak Agent configuration/management and review of audit logs. Baseline security attributes such as Cryptographic security attributes and Password Policy related settings are established using an installation wizard running directly on the Master Repository machine. These installation settings allow the Administrator to establish critical security related policies during installation that are enforced during all future use of the application. The Common Criteria Evaluated configuration includes full privileges enabled through configuration options to the Standard User role.

Security management sessions are conducted over TLS to assure confidentiality with a SHA1 HMAC to assure integrity of messages passed during Administrative sessions.

The management console application consists of a Visual Basic (VB.net) application and GUI forms which execute within the .NET framework installed on the underlying Microsoft Windows Operating System. Data objects are passed between the GUI application and the Master Repository machine utilizing a C^{++} based Active X component integrated within the CimTrak Management Console.

7.1.6.1 CimTrak Security Management Functions – FMT_SMF.1, FMT_MSA.1a, b, FMT_MSA.3

The CimTrak management console GUI provides essential application management resources through a combination of a left frame that displays in tree view format information such as Agent servers/Network devices, Object Groups and Objects and a right frame which provides a series of tabs for the configuration of settings related to selected objects. This interface also allows the viewing of current status, logs and statistics by selecting the appropriate source and tab combination.

Upon the creation of a new user or agent account, the TSF applies restrictive default values for security attributes used to enforce the CimTrak Access Control SFP. New CimTrak [Administrator]/Agent privileges must be assigned during account creation and by default have no default access rights or privileges to objects under the TOE Scope of Control (TSC) or CimTrak Master Repository resources.

The mechanism for passing configuration settings and security attributes to and from the GUI is the Active X C^{++} code that works with the Visual Basic GUI application running on the management console. Active X is used to transfer entered values from the GUI on the management console workstation, through the secure session provided by the cryptographic

subsystem and CimTrak communication protocol to successful pass the values to the Client Message Processing subsystem within the Master Repository. The Client Message Processing subsystem orchestrates the transfer of data to and from the TSF data repository maintained in the PostgreSQL database.

7.1.6.2 Managing Agents, Object Groups and Watch Groups – FMT_SMF.1, FMT_MSA.1b

Object groups allow the grouping of particular data object types managed by CimTrak on Agent servers/Network devices to allow the entire set of object types to be identified under a single group identifier. Object groups are created and configured by selecting the applicable Agent server/Network device and using the right click menu to access the create Object Group option. This renders a configuration screen that allows for the selection of files/folder to include in the object group, the number of revisions, events and copies of unauthorized changes to maintain.

The Watch group setting allows a single group to be created that contains a set of preconfigured Watch Properties, which determine the action to take if an event is detected. These Watch properties determine how the CimTrak application Agent component will react once a given event is detected. These are configured through a second page and allows the monitoring and watch properties to be configured including the remediation action to take, file comparison (hash) algorithm to use, and the event detection method to employ ("real time"*/poll based). Upon completing this configuration, the values are passed to the PostgreSQL database and then may be immediately activated to the applicable Agent through a synchronization button.

*"real time" refers to the ability to detect file object changes as they are made vs. the "polling" method which detects file changes at periodic time intervals.

Security attributes managed through this interface include:

- File Object Group Properties (group name, location, description, date/time, contact, URL, number of revisions to keep, number of events to keep, number of intrusions to keep, keep intrusion size)
- File Watch Group Properties (corrective action setting, Authoritative Copy enable/disable, File Comparison method (hash), Event Detection method, Sync enable)

7.1.6.3 User Groups/Roles, Agent Groups – FMT_SMR.1, FMT_SMF.1, FMT_MSA.1a,b

User Groups allow the associated of particular [Administrators] to access levels represented by configured group memberships. For the Common Criteria Evaluated configuration, the CimTrak application utilizes an Administrator, Standard User and Auditor User Groups which serve as roles for managing access. User Group/Role management within CimTrak is provided from a system pull down menu within the GUI that allows the Administrator to create or modify CimTrak users. Once the user is created, they can be assigned to a User Group within the GUI that establishes the level of access afforded that user based on the privileges assigned to that

group. CimTrak supports 3 levels of [Administrators]: Administrators (full access), Standard Users (configured access) and Auditors (read-only).

CimTrak also allows Agents to be logically grouped into Areas or Agent Groups. This allows a group of Agents to be delegated to a particular [Administrators] for control and management. This is configured by selecting the Master Repository IP address in the Left Frame, and accessing the pull down menu where Group identification parameters are entered. Following this configuration, the new Area appears in the left frame tree view and Agents may be dragged and dropped into the Area icon to create these Agent grouping associations.

Only the Administrator role can create or modify Permissions and email settings to roles or other users/user groups via the Permission Window within the GUI.

7.1.6.4 Audit log and Reports management - FMT_SMF.1, FMT_MTD.1e

Audit logs are accessed from the CimTrak GUI by selecting the applicable Server, Agent, Object in the Left Tree View and clicking on the Event log tab. This provides Event logs entries displayed in a tabular format. Additional tabs are included for other logs categories including Intrusion Log, Pending Repair (listing of pending actions), Monitor Info, and Generation information. When an item is selected in the Tree view, the management console application passes a query to the Client Message Processing subsystem on the Master Repository machine, which acquires the request data objects from the PostgreSQL database and passes it to the Client Core subsystem for rendering on the displayed GUI page. All [Administrator] roles (Administrator, Standard User and Auditor) may view audit logs.

7.1.6.5 Comparison Tools, Deployment and Email Notifications- FMT_SMF.1, FMT_MSA.1b

The CimTrak TOE provides various tools for Administrators and Standard Users to investigate file changes, malicious file additions or other activities that may demonstrate a threat to internal network servers.

The File Comparison tool is accessed through the CimTrak GUI by selecting an Event Log containing a file and accessing the right click menu. This yields five options: “View the File”, Download the File (to the management console), Compare with the authoritative copy (at time of intrusion), “Compare with the authoritative copy (current)” or “Add to excludes”. When one of the compare options is selected, the GUI launches a viewing screen that shows the authoritative copy and the modified copy in a side to side (comparison) format. Both file content and file attributes are shown in the display. Differences between the two files are highlighted based on a color coded legend (changed, added, deleted) for easy identification of the changes made. The management console renders this content by creating a query to the PostgreSQL database that is routed through the Client Message Processing subsystem to the PostgreSQL database where the data is acquired.

The Deployment feature allows the Administrator role to revert to a previous version of a locked file and push the authoritative copy of this version to a specific CimTrak Agent server/Network

device. The generation tab as described above under "Audit log and Reports management" is selected and a list of the saved generations of the file objects selected in the tree view is displayed by date. Upon selecting a generation, the right click menu may be accessed and the "deploy" option selected resulting in the deployment of the previous versions from the Master Repository: CimTrak database to the applicable Agent server(s)/Network device(s). The TSF mechanism that provides this functionality is largely the same as described for the "Restore" function, except that the request for the restoration originates from the CimTrak management console vs. the Agent server or Network Host server.

The Email Notification function is accessed by selecting the Master Repository from the Tree View and selecting the Mail Icon on the GUI. This allows for [Administrators] (based on the configured email address) to be notified by email for specified events. The applicable [Administrator] or user group is selected and then the appropriate categories of events are selected for email notification.

7.1.6.6 Cryptographic functions configuration & management –FMT_SMF.1, FMT_MSA.1b

Cryptographic function management screens are provided during installation processes through an Agent installation setup wizard. These installation screens, run from the Master Repository directly, allows the Administrator role to configure the Encryption algorithm to be used for securing Agent to Master Repository sessions and sessions between the CimTrak Management Console and Master Repository. Pull down menus allow the establishment of which encryption algorithm to use, key length and HMAC (algorithm). Different Agents may be configured to use different encryption types or they can all be configured to a single algorithm/key combination for convenience. A separate configuration screen is provided for User Data encryption in the CimTrak database, Agent communication encryption parameters and Management Console encryption. Pull down menu options only include secure values for use in implementing Cryptographic algorithms, and key sizes. The Common Criteria evaluated configuration requires that only FIPS 140-2 validated algorithms/key combinations are configured for use in the CimTrak TOE.

Following the configuration of this series of wizard settings, these values are saved to the PostgreSQL database for implementation by the application.

7.1.6.7 Password Policy Management- FMT_SMF.1, FMT_MSA.1a

Password policy management settings are presented in the same series of installation wizard screens as the cryptographic functions noted above. The password validation settings screen allows the configuration of password strength settings (with or without random password generation, Custom settings which allow for tailoring of specific rules for password enforcement or the no password policy option. These values are saved to the PostgreSQL database along with the cryptographic settings above upon completion of the installation wizard.

7.1.6.8 Management of Security Function Behavior – FMT_MOF.1a,b

The aforementioned security management functions are used by the Administrator role and Standard User role to modify the behavior the CimTrak application. Access controls provided by the TSF mechanisms as described in Section 7.1.2, Identification and Authentication, restrict the ability to modify the behavior of security functions to the role of Administrator as specified in security function requirement FMT_MOF.1. Read-only access to these settings is allowed for members of the Auditor role and may be optionally configured for members of the Standard User role. By default, the Standard User cannot access security attribute configuration screens, the Administrator has full read/write access to these settings and the auditor has read-only access.

This is enforced within the CimTrak management console by the Client Core subsystem which controls access to GUI screen objects and write privileges. Configuration settings for read-only users are grayed out within the GUI indicating that the authenticated role does not allow modification of selected attributes for the logged in [Administrator].

7.1.6.9 TSF Data Management – FMT_MTD.1a-e

The TOE uses the aforementioned infrastructure to control access to security attributes as well as TSF data. Access restrictions are enforced based on the authenticated CimTrak role and is enforced by the Client Core subsystem within the CimTrak Console Management machine. The TOE restricts the ability to modify TSF data within the application to the applicable [Administrator] roles. Standard User's privileges are given full privileges for the Common Criteria evaluation configuration. This allows Standard Users the same access to TSF data as Administrators with the following exceptions: Access to the Users Windows used to add/delete/edit Users and assigned email address to those accounts in restricted to users holding the Administrator role only.

In addition, only Administrators may access (query) the Logged on Users tab within the Administrator secure management interfaces.

All Administrative Users can view configured Areas, Document Controls and Object Groups.

Full descriptions of TSF data access are characterized in FMT_MTD.1a – e.

7.1.7 Secure Communications

The Secure Communication Security function assures that all communication between distributed parts of the TOE is secured using TLS. As described in Section 7.1.3, Cryptographic Operations, the TLS protocol secures these sessions utilizing symmetric key cryptography with HMAC integrity checking. Communications are conducted over port 3749 which is registered with ICANN*, however, the port number may be changed based on [Administrator] preference. The CimTrak Master Repository may also restrict connections to IP Addresses which have been allowed by prior Administrator configuration. Only the Master Repository may accept connections, the CimTrak Management Console and CimTrak Agents do not accept connections from any source.

* Internet Corporation for Assigned Names and Numbers

7.1.7.1 Secure sessions between the Agent/Client and Master Repository – FCS_CKM.1a, b; FCS_COP.1c, d; FPT_ITT.1

Sessions must begin from either the Agent or Management Console component as these components are not open (listening) for unsolicited connections from the internal network. The respective Cryptographic Subsystems provide the cryptographic support to the CimTrak Communication protocol subsystems in establishing the secure session. The Agent/Console cryptographic subsystem and Master Repository cryptographic subsystem negotiate a TLS secure session to secure data passed between the respective CimTrak components over the internal network.

CimTrak utilizes X.509 (version 3) certificates to validate the Master Repository server to the Agent machine when establishing communications. Certificates are generated automatically when initially ran or in the event the existing certificate has been deleted. These certificates are generated by the CimTrak application itself and do not involve Administrative user actions. The Master Repository Cryptographic subsystem supports certificate generation through its Cimcor Cryptographic Module.

Network devices are supported by the CimTrak Network Agent which implements SSHv2 through the Cimcor Cryptographic Module contained within the Network Agent: Cryptographic Module. The Network Device Interaction Module orchestrates this communication session with scripts provided during initial setup. Algorithm and key sizes supported are identical to those described above for Agent to Master Repository sessions (FCS_COP.1c).

7.1.8 Protection of the TOE

The CimTrak application maintains a series of TOE protection mechanisms intended to prevent intentional or accidental modifications to the executable code, monitor CimTrak repository hard drive resources to assure availability and to preserve communications security when information is passed between separate parts of the TOE.

7.1.8.1 CimTrak self test – FPT_TST_EXP.1

The CimTrak application executes a self test upon startup of each CimTrak component to verify the integrity of each respective code set. This includes the CimTrak application installed on the Master Repository machine, CimTrak Agent server/Network Host server machine(s), and CimTrak Management Console machine. Integrity checking of the executable is initiated by the respective Core subsystems during startup through SHA1 hashing techniques. Upon installation of the application on the 4 hardware platforms (master repository, Agent server, Network Host server, management console), a SHA1 hash is created by the resident Cryptographic subsystem using the SHA1 Hash function (see Section 7.1.3). This SHA1 hash value is stored for verification purposes on the respective machines. During startup process, the Cryptographic subsystem executes a Hash function against the executable code and the value derived is compared against the stored hash value to assure that no changes have been made to the executable program. Audit logs reflect a successful integrity check through a startup “success”

audit event and if integrity check fails during startup, the executable is stopped from running and an error message is rendered to the Management Console.

7.1.8.2 CimTrak data transfer protection – FPT_ITT.1

The CimTrak TOE provides data transfer protection by securing all communications between TSF components through encrypted sessions using TLS as specified in Section 7.1.7. This assures that untrusted entities cannot access TSF components through a malicious data transfer attempt. The Master Repository does not initiate communication with either Agent servers/Network Host servers or the CimTrak Management Console. This feature is enforced by the CimTrak Communications protocol subsystems on all three TSF components in association with resident FIPS 140-2 validated cryptographic modules.

7.1.8.3 CimTrak Resource utilization monitoring - FAU_ARP.1, FAU_SAA.1

The CimTrak application monitors hard disk free space on the Master Repository to assure it does not deplete available space leading to a Denial of Service (DoS) or loss of the ability to generate and store audit records. Once reaching a configured percent full, an email is sent to the configured email address as described in Section 7.1.1.

The CimTrak Master Repository also maintains a heartbeat monitor with deployed CimTrak Agents to assure that Agents are available on the network and operational. This feature is combined with the Agent Statistics reporting functions which passes Agent usage statistics to the Master Repository on a regular basis. The Administrator role assigns a value in minutes for how often the Agent must contact the Master Repository with an Agent Statistics update. If the configured time elapses without contact from the Agent, the Master Repository Security Event processing subsystem issues a notification email to the configured email address within the CimTrak application and an audit log is generated.

The Master Repository employs mechanisms to optimize availability of the Server to Agents by delaying responses to Agent requests during cycles of high CPU usage or I/O activity. By managing these responses, the Master Repository assures that adequate resources are available to process Agent requests in an efficient manner.

7.1.8.4 CimTrak Access Banner - FTA_TAB.1

The CimTrak management console application presents a banner upon launch that alerts the user, through an advisory warning message, that unauthorized use of the application or access to CimTrak resources may result in applicable sanctions. This provides a level of security in advising a potential attacker of consequences to unauthorized use of the TOE and could potentially thwart an undetermined attacker. The CimTrak banner is presented by the Client Core subsystem as part of the CimTrak management console application startup process.