Opsware System 4.5 Security Target

Version 1.0

October 28, 2005

Prepared for:
Opsware, Inc.
599 N. Mathilda Avenue
Sunnyvale, CA 94085

Prepared By:
Science Applications International Corporation
Common Criteria Testing Laboratory
7125 Columbia Gateway Drive, Suite 300
Columbia, MD 21046

LIST OF TABLES

1. Security Target Introduction

This section identifies the Security Target and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. Opsware, Inc. provides the TOE, which is Opsware version 4.5. The TOE is a software IT management tool that provides an organization the ability to consistently manage multiple servers through the establishment of best-practices. The TOE is designed to simplify and expedite the administration of servers and the deployment and maintenance of software across a heterogeneous environment. The TOE provides security functionality to ensure administrators are authenticated, control access to the management functionality, and provides the ability to manage the TOE security functionality.

- The Security Target contains the following additional sections:
- TOE Description (Section 2)
- Security Environment (Section 3)
- Security Objectives (Section 4)
- IT Security Requirements (Section 5)
- TOE Summary Specification (Section 6)
- Protection Profile Claims (Section 7)
- Rationale (Section 8)

1.1 Security Target, TOE and CC Identification

ST Title – Opsware System 4.5 Security Target

ST Version – Version 1.0

ST Date – October 28, 2005

TOE Identification – Opsware System 4.5 Patch 1

CC Identification – Common Criteria for Information Technology Security Evaluation, Version 2.1, August 1999, ISO/IEC 15408

1.2 Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security functional requirements, Version 2.1, August 1999, ISO/IEC 15408-2.
  - Part 2 Conformant
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance requirements, Version 2.1, August 1999, ISO/IEC 15408-3.
  - Part 3 Conformant
  - Evaluation Assurance Level 2 (EAL2)

1.3 Conventions, Terminology, Acronyms

This section specifies the formatting information used in the Security Target.

1.3.1 Conventions

The following conventions have been applied in this document:

- All requirements in this ST are reproduced relative to the requirements defined in CC v2.1.
- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: assignment, selection, refinement, and iteration. See section 5 for a description of how these operations are highlighted in this ST. Also see section 5 for a description of how Interpreted requirements are highlighted in this ST.

Other sections of the ST use bolding and italics to highlight text of special interest, such as captions.

1.3.2 Acronyms

The acronyms used within this Security Target:

ACM	Access Control Management
AGD	Administrator Guidance Document
CC	Common Criteria
CD-ROM	Compact Disk Read Only Memory
CM	Control Management
DAC	Discretionary Access Control
DO	Delivery Operation
EAL	Evaluation Assurance Level
FIPS	Federal Information Processing Standards Publication
GB	Gigabyte
I/O	Input/Output
NIST	National Institute of Standards and Technology
OPW	Opsware, Inc. Opsware Version 4.0
PP	Protection Profile
RADIUS	Remote Authentication Dial In User Service
SF	Security Functions
SFR	Security Functional Requirements
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functions
TSP	TOE Security Policy
TSC	TSF Scope of Control

2. TOE Description

The Target of Evaluation (TOE) is the Opsware, Inc. software Opsware 4.5, hereafter referred to as Opsware. The product is designed by Opsware, Inc., located at, 599 N. Mathilda Avenue, Sunnyvale, California 94085. The product is the TOE; there is no distinction between the product and the TOE.

2.1 Product Type

The TOE is a software IT management tool that provides an organization the ability to consistently manage multiple servers through the establishment of best-practices. The TOE is designed to simplify and expedite the administration of servers and the deployment and maintenance of software across a heterogeneous environment.

2.2 Product Description

Opsware provides a foundation for automating tasks associated with the deployment, maintenance, support, and growth of a company's IT infrastructure. Using Opsware enables an organization to decrease time for deployment, add new capacity more quickly, and perform expedited troubleshooting from a single point.

Opsware technology enables an organization to incorporate the established best operation practices of systems administrators in building and maintaining their site infrastructure. Using Opsware technology enables an organization to manage their IT infrastructure uniformly through process automation across all servers.

There are two primary modes of operation for the Opsware System, stand-alone and multimaster. Multimaster mode allows for the organization to be spread over several sites all acting in coordinated fashion. The differences between the installation modes relate solely to the requirements necessary for multiple Model Repository servers. However, in the evaluation configuration only stand-alone mode is allowed.

2.3 Product Features

The TOE implements the following features:

Provision UNIX, Linux and Windows - Build out large heterogeneous environments quickly and build servers in a way that they can be easily updated.

Provision Applications - Rapidly deploy multi-tier applications across multiple servers, simultaneously.

Securely Deploy Patches - Quickly and accurately identify server vulnerabilities and patch large numbers of servers.

Track Application Configurations - Automatically track, store and recover critical software configuration information.

Code and Content Deployment - Consistently install and promote an application release to production, while archiving the previous version.

Track Software Assets - Keep track of the managed servers and managed applications.

Distributed Scripts - Run scripts across multiple servers, simultaneously.

Custom Extensions - Run Custom Extension scripts across multiple servers, simultaneously.

2.4 Security Environment TOE Boundary

The TOE includes both physical and logical boundaries.

The TOE preserves management knowledge for system administrators, network engineers, and database administrators in a centralized knowledgebase (referred to as the Model Repository), which can then be tapped for future actions, allowing this knowledge to be preserved and used by all administrators in the system.

The TOE is comprised of the following components: Opsware Command Center, Model Repository, Software Repository, Data Access Engine, and Command Engine, along with the Opsware Agent installed on managed servers. These components are explained below.

Opsware Command Center
The Opsware Command Center (OCC) is the primary user interface to Opsware. Users accessing the Opsware Command Center are authenticated before gaining access. Through the Opsware Command Center's web-based UI the user can view and update the systems being managed by Opsware. The Opsware Command Center operates primarily via the Data Access Engine, though it talks directly to other backend services to implement some operations. These backend services are considered part of the TOE and are installed during TOE installation. These backend services are an Apache web server, BEA's WebLogic application server, and a Netscape Directory Server LDAP server.

Opsware uses an Apache web-server to support protection of internal TOE communication by performing SSL encryption through Apache's OpenSSL-based cryptographic module. The Opsware Command Center server is implemented in Java using the BEA WebLogic platform as the application server. Opsware uses a LDAP directory server to store authentication data (e.g. usernames, passwords).

Data Access Engine
The Data Access Engine provides the public API (Application Programming Interface) to the Model Repository. All clients of the Model Repository interact with the Model Repository via APIs defined by the Data Access Engine. Because interactions with the Model Repository go through the Data Access Engine, clients are sheltered from details of and changes to the Model Repository's implementation. The Data Access Engine abstraction also eliminates the need to bind database libraries into every program that implements Opsware.

The Data Access Engine is implemented as a server and hence its API is a network protocol. The protocol for transport of requests to the Data Access Engine is HTTP over the Secure Socket Layer protocol, sometimes referred to as "HTTPS" or "HTTP over SSL." The requests and responses are encoded in an XML dialect known as XML-RPC. This makes the Data Access Engine's network API language-independent.

Model Repository
The Opsware System is model-based. Essentially all Opsware System tools work from or record into a model of the systems that are being managed. The Opsware System component maintaining this model is known as the Model Repository. The Model Repository contains information about servers, network devices, data centers, etc. A data center (usually called Facility in the Opsware documentation) represents a customer site that contains managed servers and or network devices. In short, the Model Repository contains essentially all of the information required to build, operate and maintain all managed sites. Among other things the Model Repository maintains

- a list of all devices under management (servers, network devices and storage devices)
- the configuration of those devices
- the OS, system software and applications installed on servers
- the configuration of each data center
- authentication and security information

The Model Repository is implemented as an Oracle database.

Software Repository
The Software Repository is Opsware's central repository for all software managed by Opsware. It contains software packages for operating systems, application servers (e.g. WebLogic), databases and customer code. Software is stored and pulled from the Software Repository via the Word Gateway. The Word Gateway enforces access control and checks for adherence to policies such as naming and versioning.

Working with the Software Repository, an Opsware Agent can update the software running on the Opsware Agent's server. This process is often called reconciliation.

The Software Repository is also the repository for software configuration information.

Command Engine
The Command Engine is a system for running distributed programs across many servers (usually Opsware Agents). The Command Engine executes scripts written in python, which can make RPC calls on Opsware Agents. These calls are delivered in a secure manner using SSL.

Opsware Agent
Each server managed by the Opsware System has an agent, named the Opsware Agent, which runs on that server. The Opsware Agent is the Opsware System's "agent of change" on the server. Whenever the Opsware System needs to make changes to servers it does so by sending requests to the Opsware Agent. Depending on the request, the Opsware Agent may use global Opsware System services (such as the Data Access Engine and Software Repository) in order to fulfill the request. Some functions that the Opsware Agent supports are software installation and removal, and configuration of software and hardware.

The Opsware Agent is usually idle unless some part of Opsware is trying to effect some change on the server. Periodically the Opsware Agent wakes up and registers itself with the Model Repository. This allows the Model Repository to keep track of machines that have been disconnected from and reconnected to the network.

The Opsware Agent is implemented as an HTTP/HTTPS server. As described earlier in reference to the Data Access Engine, the protocol for communicating with the Opsware Agent is HTTPS and the requests and response are encoded in XML.

Figure 1 shows the Opsware components and their relations within a network.

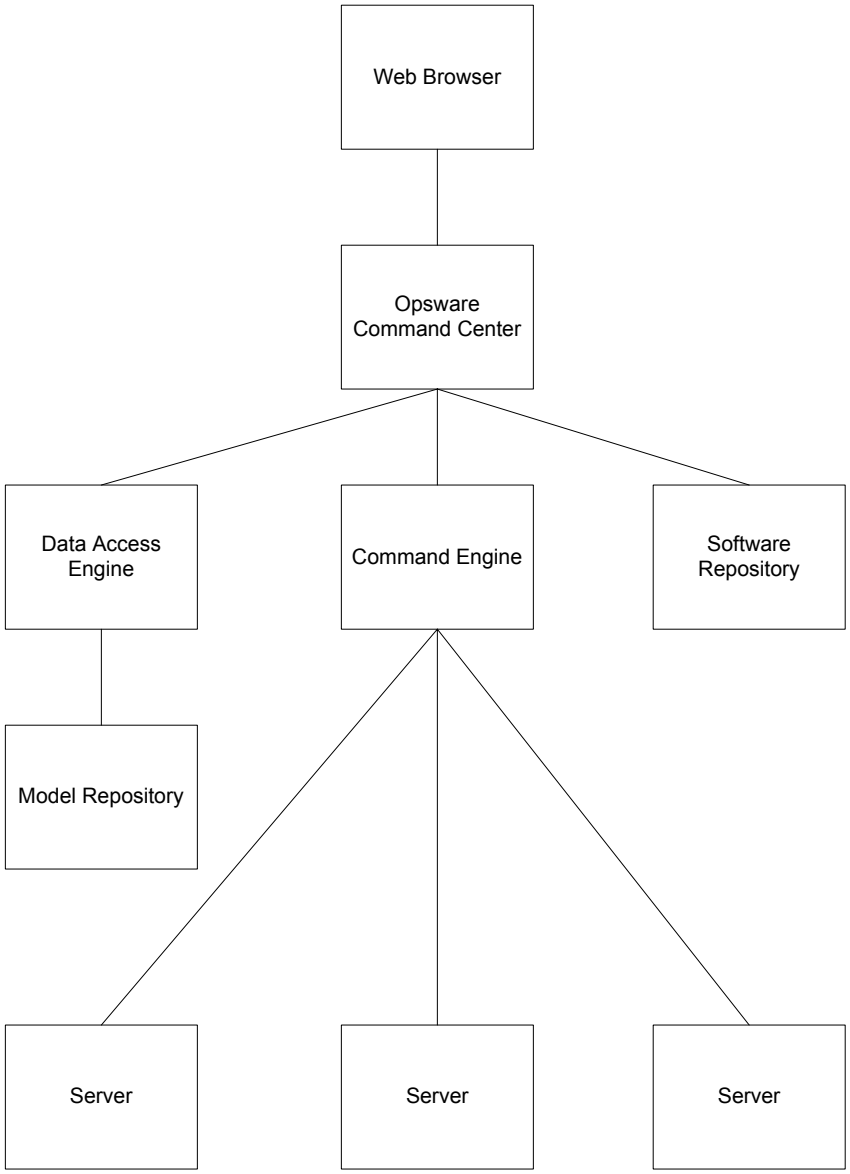

Figure 1: TOE in its IT Environment

The generic servers shown in the illustration (marked "Server") could be any OS that can be managed through the Opsware System, such as Windows, Linux, AIX, HP-UX or Solaris, and they could be running various services, from Oracle databases to web sites. These servers must have the Opsware Agent installed to enable administration through the Opsware System.

2.4.1 Physical Boundaries

The TOE physical boundaries are the external interfaces and the interfaces to the IT environment.

All of the TOE components, except for the Agent, can be installed on a single physical system, or spread across multiple physical systems. However, in the evaluated configuration all of the TOE components, except for the Agent, are installed on the same platform. The Agent must be installed on all managed servers.

The external interfaces are used to interact with users and the managed servers. These interfaces are enumerated through the Opsware Command Center and the Opsware Agent. The Opsware Command Center provides an administrative interface for all TOE management functions. From this one point, all TOE components are managed.

Administration of the Opsware System and the managed servers is performed via a web browser connecting to a web service on the Opsware Command Center server. From this interface, all aspects of the software can be managed. The Opsware Agent acts as the agent of management on the IT environment being managed through the TOE. The Agent implements the commands from the Command Center on the local servers on which the Agent is installed.

The interfaces to the IT environment refer to interfaces that provide any necessary services to the TOE that are necessary for the TOE to function properly. These interfaces are to the operating system on the platform upon which the TOE component is installed.

The following table specifies the server operating system requirements for the TOE.

Opsware Component	Server OS Requirements
Opsware Command Center	Sun Solaris 8, Red Hat Linux Advanced Server (AS) 2.1
Data Access Engine	
Model Repository	
Software Repository	
Command Engine	
Opsware Agent	Red Hat Linux 6.2, 7.1, 7.2, 7.3, 8.0, AS 2.1, 3.0, Enterprise Server (ES) 2.1, ES 3.0, Workstation 3.0; Sun Solaris SunOS 5.6, 5.7, 5.8, 5.9; HP-UX 10.20, 11.00, 11.11/11i ; Windows NT 4.0, Windows Server 2000, Windows Server 2003; AIX 4.3, 5.1, 5.2

In addition to these operating system requirements, the Model Repository requires an Oracle database for the central storage of all configuration information.

2.4.2 Logical Boundaries

The logical boundaries of the TOE include the functions of the TOE interfaces.

2.4.2.1 User Data Protection

The TOE enforces a Management Access Control policy, which restricts access to the management functions of the TOE. This protection requires that users of the TOE be authenticated before any access to the management functions is granted. Once access is granted, user access to management functions is controlled by the assigned user privileges.

2.4.2.2 Identification and Authentication

The TOE requires users to provide unique identification and authentication data before any administrative access to the system is granted. The TOE provides the ability to define levels of authority for users, providing administrative flexibility. Full administrators have the ability to define groups and their authority and they have complete control over the TOE. In Opsware security privileges are associated with Groups, and it is by assigning users to one or more groups that users get access to features within Opsware. Groups are managed by authorized administrators, and all

discussion of "security privileges" in this document should be understood to mean Groups and their associated privileges to access various parts of the TOE.

2.4.2.3 Security Management

The TOE is managed through the Opsware Command Center, a web-based interface. Through this interface TOE management can be performed by providing the administrators the ability to manage user attributes and privileges, as well as assign roles for different levels of administrative access.

2.4.2.4 Protection of Security Functions

The TOE provides protection of all data that is transferred internally between disparate TOE components. The TOE protection ensures that modifications to the transferred data between disparate TOE components can be detected, preventing unauthorized data from being transferred into the TOE. All components, except for the Agent, are installed on the same platform. The Agent must be installed on each server that is managed by the TOE. Therefore, the only disparate communication is between the Agent and the core components (e.g. Command Engine, Software Repository). Note that Figure 1 is not meant to depict "all" internal TOE communication but only to depict the TOE components and the major security relevant communication paths amongst the TOE components.

3. Security Environment

The TOE security environment describes the security aspects of the intended environment in which the TOE is to be used and the manner in which it is expected to be employed.

The statement of the TOE security environment defines the following:

- Threats that the product is designed to counter,
- Assumptions made on the operational environment and the method of use intended for the product,
- Organizational security policies with which the product is designed to comply.

3.1 Threats to Security

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

3.1.1 TOE Threats

T.PRIVIL	An unauthorized user may gain access to the TOE and exploit system privileges to gain access to TOE security functions and data.
T.TRANSIT	An unauthorized user may eavesdrop on communications between separate parts of the TOE allowing them to intercept and modify transmitted information.

3.2 Organization Security Policies

The following policies apply to the TOE and the intended environment of the TOE.

P.MANAGE	The TOE shall only be managed by authorized users.
P.PROTECT	The TOE shall be protected from unauthorized accesses and modification of TOE data and functions.

3.3 Secure Usage Assumptions

The following usage assumptions are made about the intended environment of the TOE.

3.3.1 Physical Assumptions

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

3.3.2 Personnel Assumptions

A.MANAGE	There will be one or more competent individuals assigned to manage the TOE and the security of the information it contains.
A.NOEVIL	The authorized administrators are not careless, willfully negligent, or hostile, and will follow and abide by the instructions provided by the TOE documentation.

4. Security Objectives

This section defines the security objectives of the TOE and its supporting environment. Security objectives, categorized as either IT security objectives or non-IT security objectives, reflect the stated intent to counter identified threats and/or comply with any organizational security policies identified. All of the identified threats and organizational policies are addressed under one of the categories below.

4.1 IT Security Objectives for the TOE

The following security objectives are intended to be satisfied by the TOE.

O.PROTECT	The TOE must protect itself from unauthorized modifications and access to its functions and data.
O.EADMIN	The TOE must include a set of functions that allow effective management of its functions and data.
O.ACCESS	The TOE must allow only authorized users to access the TOE and only appropriate functions and data applicable for their role.

4.2 IT Security Objectives for the Non-IT Environment

The following security objectives are intended to be satisfied by non-IT aspects of the TOE environment.

O.INSTAL	Those responsible for the TOE must ensure that the TOE is delivered, installed, managed, and operated in a manner which is consistent with IT security.
O. PHYCAL	Those responsible for the TOE must ensure that those parts of the TOE critical to security policy are protected from any physical attack.
O.PERSON	Personnel working as authorized administrators shall be carefully selected to include only those that are not careless, not willfully negligent, not hostile, will adhere to the guidance and instructions provided in the TOE documentation, and are trained for proper TOE operation.
O.CREDEN	Those responsible for the TOE must ensure that all access credentials are protected by the users in a manner which is consistent with IT security.

5. IT Security Requirements

This section of the ST details the security functional requirements (SFR) for the TOE and the IT Environment that will support the TOE. The SFR were drawn from the CC Part 2.

CC defined operations for assignment; assignment, selection, refinement, and iteration, selection, and refinement were used to tailor the requirements to the level of detail necessary to meet the stated security objectives. The convention used in this section to highlight these operations is described below:

- Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]). Note that in cases where a selection operation is combined with an assignment operation and the assignment is null, the assignment operation is simply deleted leaving on the completed selection to identify the combination of operations.
- Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., [***selection***]).
- Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …").

Iteration: allows a component to be used more than once with varying operations. In the ST, iteration is indicated by a letter in parenthesis placed at the end of the component. For example FDP_ACC.1(a) and FDP_ACC.1(b) indicate that the ST includes two iterations of the FDP_ACC.1 requirement, a and b.

The appropriate International Interpretations have been applied to the requirements included in this ST. Note that no National Interpretations have been applied in this ST. The convention used in this section to identify those requirements to which International Interpretations have been applied is described below:

- Interpreted Requirements: Requirements that have been modified based upon an International Interpretation are identified by an italicized parenthetic comment following the requirement element that has been modified (e.g. *(per International Interpretation #51)*).

5.1 TOE Security Functional Requirements

The following table lists the SFRs to be satisfied by the TOE.

Security Functional Class	Security Functional Components
User Data Protection (FDP)	FDP_ACC.1 Subset Access Control
	FDP_ACF.1 Security attribute based access control
Identification and authentication (FIA)	FIA_ATD.1 User attribute definition
	FIA_UAU.2 User authentication before any action
	FIA_UAU.7 Protected authentication feedback
	FIA_UID.2 User identification before any action
Security management (FMT)	FMT_MOF.1 Management of security functions behavior
	FMT_MSA.1 Management of Security Attributes
	FMT_MSA.3 Static attribute initialization
	FMT_SMF.1 Specification of management functions

Security Functional Class	Security Functional Components
	FMT_SMR.1 Security roles
	FMT_MTD.1 Management of TSF data (authentication data)
Protection of the TSF (FPT)	FPT_ITT.1 Basic internal TSF data transfer protection

Table 1 Security Functional Components

5.1.1 User Data Protection (FDP)

5.1.1.1 Subset access control (FDP_ACC.1)

5.1.1.1.1 FDP_ACC.1.1

The TSF shall enforce the **[Management Access Control SFP]** on **[subjects: authorized administrators, objects: Management Interfaces, operations: access to Management Interfaces]**.

5.1.1.2 Security attribute based access control (FDP_ACF.1)

5.1.1.2.1 FDP_ACF.1.1

The TSF shall enforce the **[Management Access Control SFP]** to objects based on the following:

[subjects (authorized administrators) -

user identity

security privileges

objects (management interfaces) -

security privileges].

(per International Interpretation #103)

5.1.1.2.2 FDP_ACF.1.2

The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **[An authorized user will be granted access to the management interface if their user identity is assigned the appropriate privilege(s)]**.

5.1.1.2.3 FDP_ACF.1.3

The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **[none]**.

5.1.1.2.4 FDP_ACF.1.4

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

5.1.2 Identification and authentication (FIA)

5.1.2.1 User attribute definition (FIA_ATD.1)

5.1.2.1.1 FIA_ATD.1.1

The TSF shall maintain the following list of security attributes belonging to individual users: **[user identity, authentication data, and security privileges]**.

5.1.2.2 User authentication before any action (FIA_UAU.2)

5.1.2.2.1 FAU_UAU.2.1

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

5.1.2.3 Protected authentication feedback (FIA_UAU.7)

5.1.2.3.1 FAU_UAU.7.1

The TSF shall provide only **[obscured feedback]** to the user while the authentication is in progress.

5.1.2.4 User identification before any action (FIA_UID.2)

5.1.2.4.1 FIA_UID.2.1

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

5.1.3 Security management (FMT)

5.1.3.1 Management of security functions behavior (FMT_MOF.1)

5.1.3.1.1 FMT_MOF.1.1

The TSF shall restrict the ability to **[*modify the behavior of*]** the functions **[identification and authentication]** to **[authorized administrators]**.

5.1.3.2 Management of security attributes (FMT_MSA.1)

5.1.3.2.1 FMT_MSA.1.1

The TSF shall enforce the **[Management Access Control SFP]** to restrict the ability to **[*modify, delete, and create*]** the security attributes **[user identity and security privileges]** to **[authorized administrators]**.

5.1.3.3 Static attribute initialization (FMT_MSA.3)

5.1.3.3.1 FMT_MSA.3.1

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

5.1.3.3.2 FMT_MSA.3.2

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

5.1.3.4 Specification of Management Functions (FMT_SMF.1)

5.1.3.4.1 FMT_SMF.1.1

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

a) **management of user accounts (create, delete, modify)]**

(per International Interpretation #65)

5.1.3.5 Security roles (FMT_SMR.1)

5.1.3.5.1 FMT_SMR.1.1

The TSF shall maintain the roles **[authorized administrators and users]**.

5.1.3.5.2 FMT_SMR.1.2

The TSF shall be able to associate users with roles.

5.1.3.6 Management of TSF data (authentication data) (FMT_MTD.1)

5.1.3.6.1 FMT_MTD.1.1

The TSF shall restrict the ability to **[*modify*]** the **[authentication data for users]** to **[authorized administrators and users]**.

5.1.4 Protection of the TOE security functions (FPT)

5.1.4.1 Basic internal TSF data transfer protection (FPT_ITT.1)

5.1.4.1.1 FPT_ITT.1.1

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

5.2 TOE Security Assurance Requirements

The security assurance requirements for the TOE are the Evaluation Assurance Level (EAL) 2 components as specified in Part 3 of the Common Criteria. The minimum strength of function for mechanisms used within the TOE is SOF-basic. No operations are applied to the assurance components.

Assurance Class	Assurance Components
Configuration Management (ACM)	ACM_CAP.2 Configuration items
Delivery and Operation (ADO)	ADO_DEL.1 Delivery procedures

Assurance Class	Assurance Components
	ADO_IGS.1 Installation, generation, and start-up procedures
Development (ADV)	ADV_FSP.1 Informal Function Specification
	ADV_HLD.1 Descriptive high-level design
	ADV_RCR.1 Informal correspondence demonstration
Guidance Documents (AGD)	AGD_ADM.1 Administrator guidance
	AGD_USR.1 User guidance
Tests (ATE)	ATE_COV.1 Evidence of coverage
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent testing - sample
Vulnerability assessment (AVA)	AVA_SOF.1 Strength of TOE security function evaluation
	AVA_VLA.1 Developer vulnerability analysis

Table 2 EAL2 Assurance Components

5.2.1 Configuration Management (ACM)

5.2.1.1 Configuration Items (ACM_CAP.2)

5.2.1.1.1 ACM_CAP.2.1D

The developer shall provide a reference for the TOE.

5.2.1.1.2 ACM_CAP.2.2D

The developer shall use a CM system.

5.2.1.1.3 ACM_CAP.2.3D

The developer shall provide CM documentation.

5.2.1.1.4 ACM_CAP.2.1C

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

5.2.1.1.5 ACM_CAP.2.2C

The TOE shall be labeled with its reference.

5.2.1.1.6 ACM_CAP.2.3C

The CM documentation shall include a configuration list.

5.2.1.1.7 International Interpretation RI #3

The configuration list shall uniquely identify all configuration items that comprise the TOE. *(per International interpretation #3)*

5.2.1.1.8 ACM_CAP.2.4C

The configuration list shall describe the configuration items that comprise the TOE.

5.2.1.1.9 ACM_CAP.2.5C

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

5.2.1.1.10 ACM_CAP.2.6C

The CM system list shall uniquely identify all configuration items.

5.2.1.1.11 ACM_CAP.2.1E

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

5.2.2 Delivery and Operation (ADO)

5.2.2.1 Delivery Procedures (ADO_DEL.1)

5.2.2.1.1 ADO_DEL.1.1D

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

5.2.2.1.2 ADO_DEL.1.2D

The developer shall use the delivery procedures.

5.2.2.1.3 ADO_DEL.1.1C

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

5.2.2.1.4 ADO_DEL.1.1E

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

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

5.2.2.2.1 ADO_IGS.1.1D

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

5.2.2.2.2 ADO_IGS.1.1C

The installation, generation and start-up documentation shall describe all the steps necessary for secure installation, generation, and start-up of the TOE. *(per International Interpretation #51)*

5.2.2.2.3 ADO_IGS.1.1E

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

5.2.2.2.4 ADO_IGS.1.2E

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

5.2.3 Development (ADV)

5.2.3.1 Informal Function Specification (ADV_FSP.1)

5.2.3.1.1 ADV_FSP.1.1D

The developer shall provide a functional specification.

5.2.3.1.2 ADV_FSP.1.1C

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

5.2.3.1.3 ADV_FSP.1.2C

The functional specification shall be internally consistent.

5.2.3.1.4 ADV_FSP.1.3C

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

5.2.3.1.5 ADV_FSP.1.4C

The functional specification shall completely represent the TSF.

5.2.3.1.6 ADV_FSP.1.1E

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

5.2.3.1.7 ADV_FSP.1.2E

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

5.2.3.2 Descriptive high-level design (ADV_HLD.1)

5.2.3.2.1 ADV_HLD.1.1D

The developer shall provide the high level design of the TSF.

5.2.3.2.2 ADV_HLD.1.1C

The presentation of the high level design shall be informal.

5.2.3.2.3 ADV_HLD.1.2C

The high level design shall be internally consistent.

5.2.3.2.4 ADV_HLD.1.3C

The high level design shall describe the structure of the TSF in terms of subsystems.

5.2.3.2.5 ADV_HLD.1.4C

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

5.2.3.2.6 ADV_HLD.1.5C

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

5.2.3.2.7 ADV_HLD.1.6C

The high level design shall identify all interfaces to the subsystems of the TSF.

5.2.3.2.8 ADV_HLD.1.7C

The high level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.

5.2.3.2.9 ADV_HLD.1.1E

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

5.2.3.2.10 ADV_HLD.1.2E

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

5.2.3.3 Informal correspondence demonstration (ADV_RCR.1)

5.2.3.3.1 ADV_RCR.1.1D

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

5.2.3.3.2 ADV_RCR.1.1C

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

5.2.3.3.3 ADV_RCR.1.1E

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

5.2.4 Guidance Documents (AGD)

5.2.4.1 Administrator Guidance (AGD_ADM.1)

5.2.4.1.1 AGD_ADM.1.1D

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

5.2.4.1.2 AGD_ADM.1.1C

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

5.2.4.1.3 AGD_ADM.1.2C

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

5.2.4.1.4 AGD_ADM.1.3C

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

5.2.4.1.5 AGD_ADM.1.4C

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

5.2.4.1.6 AGD_ADM.1.5C

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

5.2.4.1.7 AGD_ADM.1.6C

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

5.2.4.1.8 AGD_ADM.1.7C

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

5.2.4.1.9 AGD_ADM.1.8C

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

5.2.4.1.10 AGD_ADM.1.1E

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

5.2.4.2 User Guidance (AGD_USR.1)

5.2.4.2.1 AGD_USR.1.1D

The developer shall provide user guidance.

5.2.4.2.2 AGD_USR.1.1C

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

5.2.4.2.3 AGD_USR.1.2C

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

5.2.4.2.4 AGD_USR.1.3C

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

5.2.4.2.5 AGD_USR.1.4C

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

5.2.4.2.6 AGD_USR.1.5C

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

5.2.4.2.7 AGD_USR.1.6C

The user guidance shall describe all security requirements for the IT environment that are relevant to the user.

5.2.4.2.8 AGD_USR.1.1E

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

5.2.5 Security Testing (ATE)

5.2.5.1 Evidence of coverage (ATE_COV.1)

5.2.5.1.1 ATE_COV.1.1D

The developer shall provide evidence of the test coverage.

5.2.5.1.2 ATE_COV.1.1C

The evidence of the test coverage shall show the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.

5.2.5.1.3 ATE_COV.1.1E

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

5.2.5.2 Functional testing (ATE_FUN.1)

5.2.5.2.1 ATE_FUN.1.1D

The developer shall test the TSF and document the results.

5.2.5.2.2 ATE_FUN.1.2D

The developer shall provide test documentation.

5.2.5.2.3 ATE_FUN.1.1C

The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.

5.2.5.2.4 ATE_FUN.1.2C

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

5.2.5.2.5 ATE_FUN.1.3C

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

5.2.5.2.6 ATE_FUN.1.4C

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

5.2.5.2.7 ATE_FUN.1.5C

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

5.2.5.2.8 ATE_FUN.1.1E

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

5.2.5.3 Independent testing – sample (ATE_IND.2)

5.2.5.3.1 ATE_IND.2.1D

The developer shall provide the TOE for testing.

5.2.5.3.2 ATE_IND.2.1C

The TOE shall be suitable for testing.

5.2.5.3.3 ATE_IND.2.2C

The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

5.2.5.3.4 ATE_IND.2.1E

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

5.2.5.3.5 ATE_IND.2.2E

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

5.2.5.3.6 ATE_IND.2.3E

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

5.2.5.4 Strength of TOE security function evaluation (AVA_SOF.1)

5.2.5.4.1 AVA_SOF.1.1D

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

5.2.5.4.2 AVA_SOF.1.1C

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

5.2.5.4.3 AVA_SOF.1.2C

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

5.2.5.4.4 AVA_SOF.1.1E

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

5.2.5.4.5 AVA_SOF.1.2E

The evaluator shall confirm that the strength claims are correct.

5.2.5.5 Developer analysis (AVA_VLA.1)

5.2.5.5.1 AVA_VLA.1.1D

The developer shall perform a vulnerability analysis. *(per International Interpretation #51)*

5.2.5.5.2 AVA_VLA.1.2D

The developer shall provide vulnerability analysis documentation *(per International Interpretation #51)*

5.2.5.5.3 AVA_VLA.1.1C

The vulnerability analysis documentation shall describe the analysis of the TOE deliverables performed to search for obvious ways in which a user can violate the TSP. *(per International Interpretation #51)*

5.2.5.5.4 AVA_VLA.1.2C

The vulnerability analysis documentation shall describe the disposition of obvious vulnerabilities.[1]

5.2.5.5.5 AVA_VLA.1.3C

The vulnerability analysis documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE. *(per International Interpretation #51)*

5.2.5.5.6 AVA_VLA.1.1E

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

5.2.5.5.7 AVA_VLA.1.2E

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

6. TOE Summary Specification

This chapter describes the security functions and associated assurance measures.

6.1 TOE Security Functions

Each of the security function descriptions is organized by the security requirements corresponding to the security function. Hence, each function is described by describing how it specifically satisfies each of its related requirements. This serves to both describe the security functions and rationalize that the security functions are suitable to satisfy the necessary requirements.

6.1.1 User Data Protection

FDP_ACC.1 Subset Access Control

The TOE does not allow access to the management interface prior to successful user authentication.

FDP_ACF.1 Security Attribute based access control

The TOE maintains a list of user accounts and each user account includes authentication data and associated security privileges. A user must successfully authenticate with a valid username and authentication data before any access will be granted to the management interface. Based upon the user identity, the user is assigned security privileges by the authorized administrator. Management interfaces are also assigned security privileges that must be possessed by a user to successfully invoke that management interface. The TSF ensures that a user has been assigned the appropriate privilege before allowing that user to access a management interface by checking the privileges assigned to the user against those required by the interface.

6.1.2 Identification and Authentication

FIA_ATD.1 User Attribute Definition

The TOE maintains the following attributes for each user account in the TOE: user identity (referred to as the user name), authentication data (password), and their assigned role (privileges). The authentication data is a case-sensitive, alpha-numeric value that can be from 6 to 31 characters in length. These user attributes are defined by the authorized administrator.

FIA_UAU.2 User Authentication before Any Action and **FIA_UID.2 User identification before any action**

The Opsware System provides a single user interface for all user actions taken within the software: the Opsware Command Center, which is served through the Opsware Command Center web server. The TOE requires users (authorized administrators) to provide unique identification (user name) and authentication data (passwords) before access to the system is granted. The TOE compares the entered password to the password assigned to account associated with the user name entered and only allows access to the system if the passwords match. Therefore, no administrative actions are allowed until the TOE successfully authenticates the user.

FIA_UAU.7 Protected authentication feedback

When authentication data is entered into the TOE, the input is obscured to prevent unauthorized viewing of the entered data.

6.1.3 Security Management

FMT_MOF.1 Management of security functions behavior

The TOE restricts the management of identification and authentication to authorized administrators. Only authorized administrators can change the behavior of the Identification and Authentication security function by changing the definition of a user account or by adding or removing user accounts.

After the TOE has authenticated a user the Opsware System displays the Opsware Command Center home screen. The TOE only displays only servers and management tasks the user is authorized to perform.

FMT_MSA.1 Management of security attributes

Only users with administrative privilege to access the User Administration sub-interface can set up authorizations for other Users. Therefore, the TOE only allows authorized administrators to be able to create, modify, and delete the following attributes which are associated with user accounts: security privileges and user identity.

FMT_MSA.3 Static attribute initialization

When the authorized administrator creates an account, there are no security privileges assigned to the account by default and no user can change this default behavior. However, after creation of the account the authorized administrator can assign security privileges to the account.

FMT_SMF.1 Specification of management functions

The TOE provides interfaces to the authorized administrator that allow for manage of user accounts, including the ability to create, delete and modify existing accounts.

FMT_MTD.1 Management of TSF Data (Authentication Data)

The TOE only allows the authorized administrators the ability to change the authentication data (passwords) associated with all accounts. Additionally, the TOE allows users to change only their own password.

FMT_SMR.1 Security Roles

Opsware provides centralized management through the Opsware Command Center. Through this interface an authorized administrator can assign the various privileges available within the software to users. This is done through the creation of roles, which consist of the identification of a security privilege or a number of security privileges. These roles are then assigned to individual user accounts.

The TOE has one pre-defined role, Authorized Administrator, as well as the capability to define multiple customized roles, each with its own set of privileges.

- Authorized Administrator: this role can perform all management functions on the TOE. A user with this role can manage user accounts (create, delete, modify), along with other security-relevant information.
- Users: all other roles as defined by the authorized administrator.

6.1.4 Protection of Security Functions

FPT_ITT.1 Basic internal TSF data transfer protection

Only the Agent component of the TOE is separate from the other TOE components. The TOE protects internal communications between the Agent component and the other TOE components from by using digital signatures on all transmitted information. The TOE uses SSL to enforce this protection of communication as well as public/private key pairs for signatures. These key pairs are created during the initial installation and deployment of the software. By using SSL, the data transmitted between the Agent and other parts of the TOE are encrypted and digitally signed. Therefore, the data is protected from disclosure and modification during transmission between the Agent and other TOE components.

6.2 TOE Security Assurance Measures

The following assurance measures are applied to satisfy the Common Criteria EAL2 assurance requirements:

- Process Assurance;
- Delivery and Guidance;
- Design Documentation;
- Tests; and
- Vulnerability Assessment.

6.2.1 Process Assurance

6.2.1.1 Configuration Management

The configuration management measures applied by Opsware ensure that configuration items are uniquely identified. The configuration management measures and Configuration List are documented in:

- Opsware Configuration Item List

The Configuration Management assurance measure satisfies the ACM_CAP.2 assurance requirements

6.2.2 Delivery and Guidance

Opsware provides delivery documentation that explains how the TOE is delivered and procedures to identify the TOE, allow detection of unauthorized modifications of the TOE and installation and generation instructions at start-up. Opsware's delivery procedures describe the steps to be used for the secure installation, generation, and start-up of the TOE. These procedures are documented in:

- Opsware Delivery and Operation Procedures

Opsware provides guidance that addresses the installation and initialization procedures. The installation and generation procedures describe the steps necessary to install Opsware products in accordance with the evaluated configuration. Opsware also provides guidance that describes how to manage and use the TOE security functionality.

The administrator guidance is documented in:

- Opsware Configuration and Management Guide

The Delivery and Guidance assurance measure satisfies the following Assurance requirements:

- ADO_DEL.1;
- ADO_IGS.1;
- AGD_ADM.1; and,
- AGD_USR.1.

6.2.3 Development

The Design Documentation provided for OPW is provided in two documents:

- Opsware Functional Specification
- Opsware High-level Design

These documents serve to describe the security functions of the TOE, its interfaces both external and between subsystems, the architecture of the TOE (in terms of subsystems), and correspondence between the available design abstractions (including the ST). The Design Documentation security assurance measure satisfies the following security assurance requirement:

- ADV_FSP.1;

- ADV_HLD.1; and,
- ADV_RCR.1.

6.2.4 Tests

The Test Documentation is found in the following documents:

- Opsware Test Coverage
- Opsware Test Plan
- Opsware Test Procedures

These documents describe the overall test plan, testing procedures, the tests themselves, including expected and actual results. In addition, these documents describe how the functional specification has been appropriately tested.

The Tests assurance measure satisfies the following assurance requirements:

- ATE_COV.1;
- ATE_FUN.1; and,
- ATE_IND.2.

6.2.5 Vulnerability Assessment

Each probabilistic or permutational mechanism used by the TOE must satisfy the SOF-Basic requirements. The only probabilistic or permutational mechanism used in the TOE is the authentication mechanism (FIA_UAU). Opsware has performed a strength of function analysis that indicates that the authentication mechanism fulfills at least SOF-basic. Similarly, Opsware performed a vulnerability analysis of the TOE to identify weaknesses that can be exploited in the TOE. Both the strength of function analysis and the vulnerability analysis are documented in:

- Opsware Vulnerability Assessment

The Vulnerability Assessment assurance measure satisfies the following assurance requirements:

- AVA_SOF.1; and,
- AVA_VLA.1.

7. Protection Profile Claims

There are no PP claims for this evaluation.

8. Rationale

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

- Security Objectives;
- Security Functional Requirements;
- Security Assurance Requirements;
- Security Functional Requirement Dependencies;
- Explicitly Stated Requirements
- TOE Summary Specification;

8.1 Security Objectives Rationale

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

8.1.1 Security Objectives Rationale for the TOE and Environment

This section provides evidence demonstrating the coverage of organizational policies and usage assumptions by the security objectives.

	O.PROTECT	O.EADMIN	O.ACCESS	O.TIME	O.PHYCAL	O.INSTAL	O.PERSON	O.CREDEN
A.LOCATE					X			
A.MANAGE							X	
A.NOEVIL					X	X	X	X
A.TIME				X				
T.PRIVIL	X		X					
T.TRANSIT	X							
P.MANAGE	X	X	X					
P.PROTECT	X		X					

Table 3 Environment to Objective Correspondence

8.1.1.1 A.LOCATE

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

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

8.1.1.2 A.MANAGE

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

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

8.1.1.3 A.NOEVIL

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

The O.INSTAL objective ensures that the TOE is properly installed and operated and the O.PHYCAL objective provides for physical protection of the TOE by authorized administrators. The O.CREDEN objective supports this assumption by requiring protection of all authentication data. O.PERSON ensures that the authorized administrators are not careless, negligent, or hostile, and will adhere to the TOE documentation.

8.1.1.4 T.PRIVIL

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

The O.ACCESS objective provides for ensuring that only authorized users can access the TOE and gain access to TOE functions and data. The O.PROTECT objective addresses this threat by providing TOE self-protection.

8.1.1.5 T.TRANSIT

An unauthorized user may eavesdrop on communications between separate parts of the TOE allowing them to intercept and modify transmitted information.

The O.PROTECT objective addresses this threat by ensuring that the TOE is protected from unauthorized modification to its functions and data.

8.1.1.6 P.MANAGE

The TOE shall only be managed by authorized users.

The O.ACCESS objective supports this policy by preventing unauthorized access to any data, only allowing authorized users access to the TOE. O.EADMIN ensures there is a set of management functions for administrators to use.

8.1.1.7 P. PROTECT

The TOE shall be protected from unauthorized accesses and modification of TOE data and functions.

The O.ACCESS objective supports this policy by preventing unauthorized access to any data, only allowing authorized users access to the TOE. The O.PROTECT objective addresses this policy by providing TOE self-protection.

8.2 Security Requirements Rationale

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

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

8.2.1 Security Functional Requirements Rationale

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

	O.PROTECT	O.EADMIN	O.ACCESS
FDP_ACC.1			X
FDP_ACF.1			X
FIA_ATD.1			X
FIA_UAU.2	X		
FIA_UAU.7			X
FIA_UID.2	X		
FMT_MOF.1		X	
FMT_MSA.1		X	
FMT_MSA.3		X	
FMT_SMF.1		X	
FMT_SMR.1	X		
FMT_MTD.1		X	
FPT_ITT.1	X		

Table 4 Objective to Requirement Correspondence

8.2.1.1 O.PROTECT

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

The TOE is required to authenticate all users prior to any administrative access. [FIA_UAU.2, FIA_UID.2] The TOE requires that users be assigned to roles to determine the level of access granted to the TOE. [FMT_SMR.1] The TOE is required to detect modifications to internal data transferred between disparate parts of the TOE. [FPT_ITT.1]

8.2.1.2 O.EADMIN

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

The TOE must provide the authorized administrators the ability to manage the user accounts of the TOE as well as authentication data. [FMT_MTD.1] The TOE places restrictions on access to the management interface. These restrictions include creating new accounts, modifying security roles and authentication data. [FMT_MOF.1, FMT_MSA.1, FMT_SMF.1] The TOE ensures restrictive default settings for new user accounts. [FMT_MSA.3]

8.2.1.3 O.ACCESS

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

The TOE must prevent unauthorized users from accessing the administrative interface. [FDP_ACC.1, FDP_ACF.1]. The TOE requires that all users of the TOE be unique and have unique data [FIA_ATD.1]. The TOE is required to prevent authentication data feedback from being used by unauthorized users [FIA_UAU.7].

8.3 Security Assurance Requirements Rationale

This ST contains the assurance requirements from the CC EAL2 assurance package and is based on good commercial development practices. This ST has been developed for a generalized environment with a medium level of risk to the assets. The security environment assumes physical protection and the TOE itself offers only a very limited interface and can only be configured during initialization, offering essentially no opportunity for an attacker to subvert the security policies without physical access. As such, it is believed that EAL 2 provides an appropriate level of assurance in the security functions offered by the TOE.

8.4 Requirement Dependency Rationale

The ST satisfies all the requirement dependencies of the Common Criteria, except as noted below. Table 5 Requirement Dependency Rationale lists each requirement from Section 5.1 with a dependency and indicates which requirement was included to satisfy the dependency, if any. For each dependency not included, a justification is proved.

Functional Component	Dependency	Included
FDP_ACC.1	FDP_ACF.1	YES
FDP_ACF.1	FDP_ACC.1	YES
FIA_UAU.2	FIA_UID.2	YES
FIA_UAU.7	FIA_UAU.2	YES
FMT_MOF.1	FMT_SMF.1 FMT_SMR.1	YES
FMT_MSA.1	FDP_ACC.2 FMT_SMF.1 FMT_SMR.1	YES
FMT_MSA.3	FMT_MSA.1 FMT_SMR.1	YES
FMT_SMR.1	FIA_UID.2	YES
FMT_MTD.1	FMT_SMF.1 FMT_SMR.1	YES

Table 5 Requirement Dependency Rationales

8.5 Explicitly Stated Requirements Rationale

There are no explicitly stated requirements.

8.6 Strength of Function Claims Justification

The TOE is targeted at a generalized IT environment with good physical access security and competent administrators. Within such environments it is assumed that attackers will have a low attack potential. As such, minimum and explicit strength of function claims is ‘SOF-basic’ which is appropriate for the intended environment. Note that the only applicable mechanisms (i.e., those that are probabilistic or permutational yet not cryptographic) are related to the identification and authentication security function and specifically to the following security functional requirements: FIA_UAU.2, FIA_UID.2.

8.7 TOE Summary Specification Rationale

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

requirements. Furthermore, all of the security functions are necessary in order for the TSF to provide the required security functionality.

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

	ACCESS CONTROL	IDENTIFICATION AND AUTHENTICATION	SECURITY MANAGEMENT	SELF PROTECTION
FDP_ACC.1	X			
FDP_ACF.1	X			
FIA_ATD.1		X		
FIA_UAU.2		X		
FIA_UAU.7		X		
FIA_UID.2		X		
FMT_MOF.1			X	
FMT_MSA.1			X	
FMT_MSA.3			X	
FMT_SMF.1			X	
FMT_SMR.1			X	
FMT_MTD.1			X	
FPT_ITT.1				X

Table 6 Security Functions vs. Requirements Mapping