BladeLogic Operations Manager Version 7.4.2

Security Target

Version 2.0

11 November 2009

Prepared for:
BladeLogic, Inc
10 Maguire Road, Building 3
Lexington, MA 02421

Prepared by:
Booz Allen Hamilton
Common Criteria Testing Laboratory
900 Elkridge Landing Road, Suite 100
Linthicum, MD 21090-2950

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 Security Target Introduction

This chapter presents the Security Target (ST) identification information and an overview of the ST. An ST contains the Information Technology (IT) security requirements of an identified Target of Evaluation (TOE) and specifies the functional and assurance security measures offered by that TOE to meet stated requirements.

1.1 Security Target, TOE and CC Identification

This section provides information needed to identify and control this ST and its Target of Evaluation. This ST targets Evaluation Assurance Level (EAL) 3 augmented with ALC_FLR.1.

ST Title:	BladeLogic Operations Manager v7.4.2 Security Target
ST Version:	2.0
ST Publication Date:	11 November 2009
ST Author:	Booz Allen Hamilton
TOE Identification:	BladeLogic Operations Manager Version 7.4.2
CC Identification:	Common Criteria (CC) for Information Technology Security Evaluation, Version 2.3, August 2005 to include all applicable National Information Assurance Partnership (NIAP) and International interpretations through 30 December 2007.
ST Evaluator:	Booz Allen Hamilton Common Criteria Testing Laboratory
Keywords:	Role-based access control, remote server management, real-time audit, compliance policy.

1.2 TOE Overview

The TOE is the BladeLogic Operations Manager Version 7.4.2 that is a data center configuration management solution for remote servers that is comprised of components that reside on various nodes as detailed in Section 2.6. The TOE, when deployed, is a suite of software that provides a comprehensive solution for management of remote servers.

1.3 Security Target Organization

Chapter 1 of this ST provides introductory and identifying information for the BladeLogic Operations Manager v7.4.2 Chapter 2 describes the TOE and provides some guidance on its use. Chapter 3 provides a security environment description in terms of assumptions, threats and organizational security policies. Chapter 4 identifies the security objectives of the TOE and of the Information Technology (IT) environment. Chapter 5 provides the TOE security functional requirements, the assurance requirements, as well as requirements on the IT environment. Chapter 6 is the TOE Summary Specification, a description of the functions provided by the BladeLogic Operations Manager v7.4.2 to satisfy the security functional and assurance requirements. Chapter 7 provides a rationale for claims of conformance to a registered Protection Profile (PP). Chapter 8 provides a rationale, or pointers to rationale, for objectives, requirements, TOE Summary Specification, and PP claims.

1.4 onformance Claims

This ST is CC Part 2 conformant and CC Part 3 augmented with ALC_FLR.1 for EAL3 to include all applicable NIAP and International interpretations through 30 December 2007.
This ST does not claim Protection Profile (PP) conformance.

1.5 Conventions, Terminology and Acronyms

This section identifies the formatting conventions used to convey additional information and terminology. It also defines terminology and the meanings of acronyms used throughout this ST.

1.5.1 Conventions

This section describes the conventions used to denote CC operations on security functional and assurance components and to distinguish text with special meaning. The operations performed on the security functional and assurance components contained in this ST adhere to the following conventions:

Iteration: Allows a component to be used more than once with varying operations. In the ST, a number in parenthesis appended to a component indicates iteration. For example, FMT_MOF.1 Management of security functions behavior (1) and FMT_MOF.1 Management of security functions behavior (2) indicate that the ST includes two iterations of the FMT_MOF.1 component.

Assignment: Allows the specification of an identified parameter. Assignments are indicated using italicized text and are surrounded by brackets (e.g., [*assignment*]).

Selection: Allows the specification of one or more elements from a list. Selections are indicated using bold italicized text and are surrounded by brackets (e.g., [***selection***]).

Refinement: Allows the addition of details. Refinements are indicated using bold text for additions to the requirements (e.g., **refinement**). In addition, refinements based upon Common Criteria Interpretations Management Board (CCIMB) interpretations are indicated in red italicized text for additions, and strikethrough red italicized text for deletions (e.g., *text added* ~~*text removed*~~).

1.5.2 Terminology

The following additional terms are specific to this ST:

Server: A server is a machine where Remote System Call Daemon (RSCD) Agent software was installed.
Client: Machines that are running Configuration Manager, or Network Shell.
Provisioning: The remote installation of operating systems or applications from the Application server to an RSCD Agent Server.
RBACAdmins: A built-in role with authorizations granting the user in this role permission to read and modify ACL authorizations for all system objects in BladeLogic.
RBACAdmin: The built-in user which is assigned the RBACAdmins role.
BLAdmins: A built-in role with authorizations granting the user in this role permission to change permissions for all system objects.
BLAdmin: The built-in user which is assigned the BLAdmins role.
Roles: A role is an organization entity such as administrators or database managers.

1.5.3 Acronyms

The following acronyms are used in this ST:

Acronym	Description
ACL	Access Control List
AES	Advanced Encryption Standard
CC	Common Criteria
CCIMB	Common Criteria Interpretations Management Board
CLI	Command Line Interface
EAL	Evaluation Assurance Level
GUI	Graphical User Interface
HTTPS	Hyper Text Transfer Protocol Secure Socket
IP	Internet Protocol
IT	Information Technology
NIAP	National Information Assurance Partnership
PP	Protection Profile
PXE	Preboot Execution Environment
RBAC	Role-Based Access Control
RSCD	Remote System Call Daemon
SRP	Secure Remote Password
SSL	Secure Sockets Layer
ST	Security Target
TLS	Transport Layer Security
TOE	Target of Evaluation

2 BladeLogic Operations Manager Version 7.4.2 TOE Description

This section provides a description of the TOE in its evaluated configuration. This includes identification and descriptions for the components that comprise the TOE.

The BladeLogic Operations Manager allows enterprise administrators to view and manage server configurations, deploy software and complex packages of files and other server assets, store server configurations, and compare servers to detect discrepancies in their configurations. The TOE's operations are available for Windows, Solaris, and Linux servers and applications.

The only functionality that is not within the evaluated configuration is provisioning. This function was not included because provisioning does not have a security relevant behavior, nor will it interfere with other security functional claims of the TOE.

The TOE will:

- Record the configuration of a group of server objects at a point in time, a snapshot report.
- Determine whether server configurations match a standard configuration, an audit report.
- Provide a mechanism for automatically correcting any discrepancies, remediation
- Manage the configuration of patches on servers by comparing them to standard configurations, patch analysis.
- Deploy packages and software to install on remote servers without requiring any user interaction.
- Deploy files and directories to remote servers.

2.1 Components of the TOE

The BladeLogic Operations Manager provides a GUI-based system for automating management of remote servers. The TOE gives system administrators the ability to remotely manage applications and patches, perform disaster recovery, and monitor for compliance to standards.

2.2 Jobs

A job is a set of instructions for performing a task on one or more servers. Configuration Manager provides many types of jobs, described below:

- Snapshot Jobs—Records the configuration of a group of server objects at a point in time.
- Audit Reports—Determines whether server configurations comply with a standard configuration.
- Patch Analysis Jobs—manages the configuration of patches on servers by comparing them to standard configurations.

- Deploy Jobs—Deploys BLPackages and software installables to remote servers without user interaction.
- File Deploy Jobs—Copies files and directories from a managed server to multiple locations.
- Network Shell Script Jobs—Deploys and executes Network Shell scripts on the application server and remote servers.
- Batch Jobs—Runs a concatenated series of Deploy Jobs, File Deploy Jobs, and Network Shell Jobs on remote servers.

A Job Run is a job that has been executed at a particular time on one or more servers. There may be many job runs for a single job. A job definition is the set of instructions that are in effect for a particular job run.

2.2.1 Snapshot Jobs

Snapshots record the configuration of a group of server objects at a point in time. Snapshots are particularly useful for capturing standard configurations that can be used when performing audits. To take a snapshot, the user must run a Snapshot Job. When the user defines a Snapshot Job, a snapshot of components or live server objects is taken.
Using snapshot results, users can:

- Base an audit on a snapshot.
- Bundle snapshot results into a BLPackage or software package.
- Export the snapshot results into a configuration file.

2.2.2 Packages

The TOE allows users to create packages and other types of objects users can store in a staging area (i.e., the Depot). Once these objects have been created, jobs can be used to deploy the objects on multiple servers. The following types of objects can be created and stored in the Depot:

- Software—Windows or UNIX-style executables or patches.
- BLPackages—Aggregations of many types of server objects. BLPackages can be used to deploy complex server configurations.
- Network Shell Scripts—Scripts that can be stored as depot objects and then deployed using a Network Shell Script Job.
- Files—Files that can be stored as depot objects and then added to BLPackages.

The Depot workspace is where users perform a number of tasks to organize depot content (e.g., files, directories, etc.).

2.2.3 Depot Workspace

The Depot workspace holds objects that a role has created or shared with another role. Generally these objects are needed to execute Configuration Manager jobs. The Depot is used to hold files, BLPackages, Software Packages and Network Shell Scripts.

2.2.4 Audit Jobs

Audit Jobs can be used to determine whether server configurations match a standard configuration. Audit results can be used to automatically correct any deviations from that standard. Audit jobs can be defined by selecting a component, snapshot, or live server objects. Using audit reports, users can:

- Identify servers that have deviated from a standard configuration
- Synchronize one or more server configurations
- Export the audit reports

2.2.5 Patch Analysis Jobs

A Patch Analysis Job can be used to manage the configuration of patches on servers by comparing them to industry-standard configurations. Using patch analysis, users can:

- Identify servers with patch configurations that deviate from the standard.
- Deploy any patches necessary to synchronize a server's configuration with the recommended configuration (done via a deploy job).

The TOE provides patch management capabilities for Microsoft Windows 2003 servers. A Patch Analysis Job examines all types of Windows patches and service packs that should be installed on a server, including patches and service packs for both Microsoft operating systems and other Microsoft products such as SQL Server and Office. When a Patch Analysis Job is run, the Application Server checks to ensure remotely managed servers have the latest version of these files.

2.2.6 File Deploy Jobs

File Deploy Jobs allow the user to deploy (or push) multiple files and directories to one or more managed servers. When a directory is deployed, the contents of the directory are copied recursively, meaning that all sub-directories and their contents are also deployed. File Deploy Jobs also allow the users to choose the files and directories wanted for deployment from managed servers. Users may select files and directories that are available under the Live node for a server in the Servers workspace. Deploy files stored in the Depot must be bundled as a BLPackage and then deployed by a Deploy Job. Deploying files as BLPackages provides far more control over a job, including the ability to simulate its deployment, automatically roll the job back when a failure occurs, and manually undo the job.

2.2.7 Deploy Jobs

Deploy Jobs allow users to deploy software packages or a BLPackage to one or more remote servers. Both software packages and BLPackages are executable packages that can be deployed unattended.

If a user wants to uninstall software packages, an uninstall job must be created. An uninstall job is nothing more than a Deploy Job that pushes a software package to servers where the uninstall should occur and then runs an uninstall command. Users can control the flow of an uninstall job just as a Deploy job, and can retry and undo an uninstall job. If the user only wants to deploy files or directories, a File Deploy Job can be used rather than a Deploy job. However, bundling files and directories as BLPackages and using a Deploy Job to deploy them gives the user considerably more control over a job, including the ability to simulate its deployment, automatically roll the job back when a failure occurs, and manually undo the job.

2.2.8 Network Shell Script Jobs

Network Shell Script Jobs allow the user to deploy and execute a Network Shell script, outside the evaluation configuration, previously saved in the Depot. In addition, Network Shell Scripts may be run on the Application server or on the targeted managed servers.

2.2.9 Batch Jobs

Batch Jobs are useful when users must perform multiple discrete jobs. For Example, a Batch Job can deploy a series of BLPackages to update a distributed application that consists of components running on database, application, and web servers.

2.3 Role Based Access Control

In BladeLogic, access control is managed through role-based and object-based authorizations. Role-based access control (RBAC) limits the actions users can perform on a system-wide basis. RBAC Manager is a graphical user interface for managing access permissions. RBAC administrators grant access permissions to users by defining a set of authorizations bundled in a role, and assigning users to those roles. A user can have many roles, but a user can only assume one role at a time. A role and a corresponding Access Control List (ACL) can grant access to specified servers, authorize users to perform actions and manage all permissions granted through the TOE.

The definition of a system object includes a set of authorizations specifying roles that can access the object and the actions those roles can perform. Authorizations can be set for all system objects in BladeLogic, including objects that function as resources, such as servers and components.

After using RBAC to define roles and users, the content of the Agent ACLs is pushed down to Agents on remote servers via ACL push jobs, thereby restricting access to those servers. On remote servers, RBAC converts authorization (i.e., permission) information into entries in the TOE's configuration files. The configuration files are access control lists (ACLs) that define user access to an agent..

2.4 TOE Overview

The TOE is the BladeLogic Operations Manager version 7.4.2 When deployed, the TOE is a suite of software that provides a comprehensive solution for the management of remote servers. The TOE provides two authentication mechanisms (i.e., Secure Remote Password and Active Directory) however, only the Secure Remote Password implementation is utilized as part of the TOE's evaluated configuration. The TOE includes the following components:

- **Configuration Manager–** a Graphical User Interface (GUI)-based system for automating management of remote servers.
- **BladeLogic Command Line Interface (BLCLI) –** A command line interface that allows access to the Configuration Manager to perform most procedures.
- **Network Shell –** A cross-platform shell with full scripting capability that gives seamless access to remote servers from central management workstations.
- **BladeLogic Application Server –** A server that manages communication between management consoles and remote servers. It also controls Configuration Manager's interaction with the database, file, and mail servers.
- **BladeLogic Reports Server –** A Web Browser on the Administration console is used to connect to the BladeLogic Reports Server. This communication serves as a web-

based report viewer that provides reports about the servers managed by the TOE. Reports are generated from data obtained from the Reporting Data Warehouse interface.

- **RSCD Agent –** Remote servers that are installed and running with specific software to allow access from the BladeLogic Application Server or Network Shell clients operating on the Administration Console.

- **BladeLogic Core Database –** An interface to the SQL Server from the BladeLogic Application Server. The Core Database stores configuration and event data used by other BladeLogic Operations Manager components in the SQL Server Database.

- **BladeLogic Reporting Data Warehouse –** An interface to the SQL Server from the BladeLogic Reports Server. The Reporting Data Warehouse is populated using information from the core BladeLogic database and agent logs.

2.4.1 Excluded From TOE

- Provisioning - Does not have a security relevant behavior, nor will it interfere with other security functional claims of the TOE.
- Blcred utility – BLCLI users can use this utility to obtain an SSO credential. However, in the evaluated configuration, BLCLI users will obtain the SSO credential through Configuration Manager prior to logging on to the TOE.
- Network Shell (NSH) Proxy Service – In the evaluated configuration, Network Shell users will communicate directly with the RSCD Agents.
- Active Directory (AD)/Kerberos authentication mechanism – SRP authentication is used in the evaluated configuration.
- Blasadmin utility – used for setting up the initial configuration of the Application Server.

2.5 TOE Architecture

The TOE maintains a three-tier architecture that consists of client, server, and middle tiers. Figure 1 illustrates the relationship between the major components of the three-tiered BladeLogic system used for the evaluated configuration of the TOE.

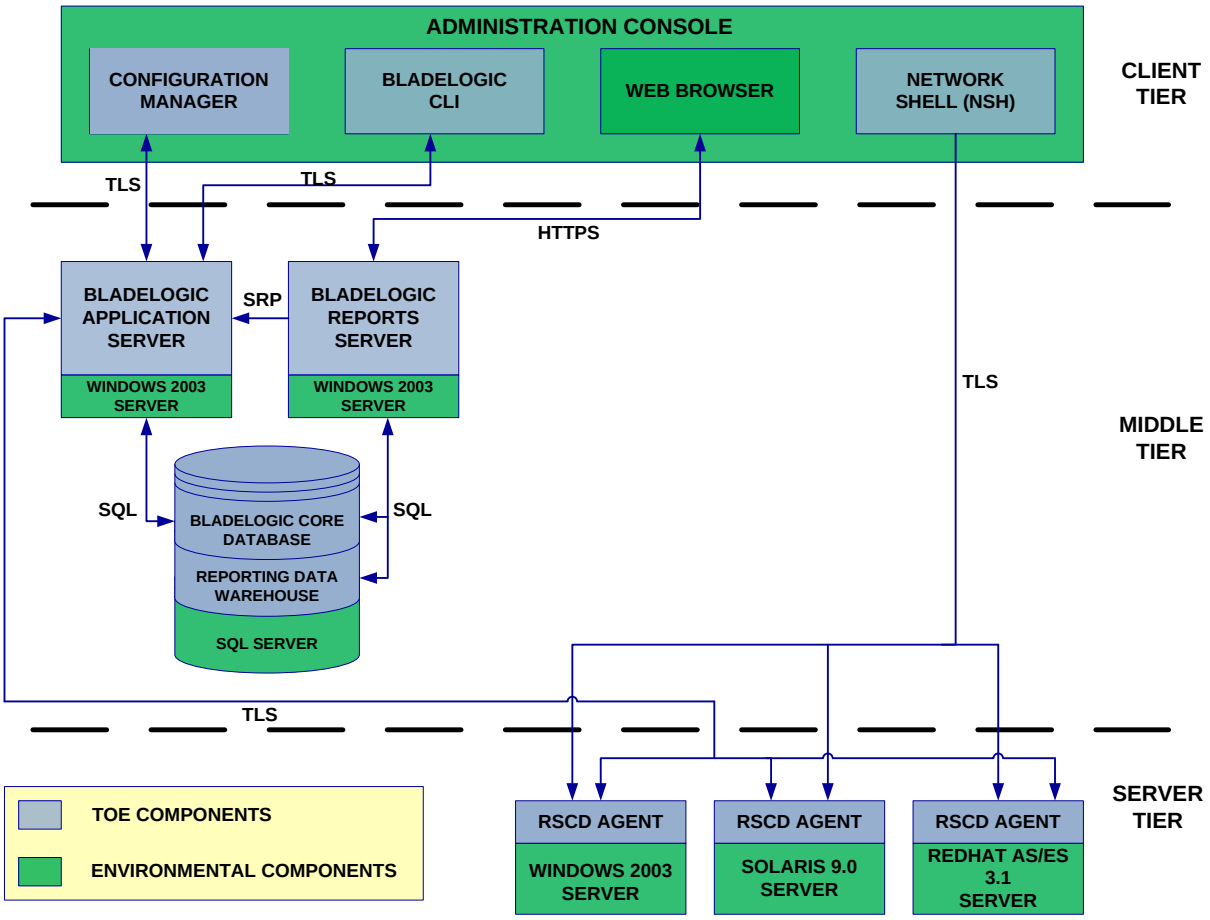

Figure 1 – BladeLogic Operations Manager Version 7.4.2 TOE Boundary

2.5.1 TOE Client Tier

The client tier provides three types of administrative management consoles: Configuration Manager, BladeLogic CLI, and Network Shell. In the evaluated configuration, all of BladeLogic's client-tier applications run on top of a Microsoft Windows 2003 server platform that allows for the management of the Middle and Server Tier components.

2.5.2 TOE Middle Tier

The middle tier provides the communications protocols and management of communications. In addition, the middle tier controls the Configuration Manager's interaction with the database and provides instructions on issuance of operating systems and applications to the RSCD Agent servers.

All clients and servers are set to communicate using secure communication based on Transport Layer Security (TLS). TLS automatically negotiates the strongest form of encryption that clients and servers can support. The cryptography used in this product has not been FIPS certified nor has it been analyzed or tested to conform to cryptographic standards during this evaluation. All cryptography has only been asserted as tested by the vendor.

2.5.3 TOE Server Tier

The TOE's server tier consists of RSCD Agents on remote servers that run Solaris 9.0 Server, RedHat AS/ES 4.0 Server, and Windows 2003 Server.

2.6 TOE Components

2.6.1 BladeLogic Configuration Manager

The Configuration Manager is a GUI that server administrators use for managing and automating data center procedures Once the user has been authenticated, their actions upon objects/resources are controlled via the BladeLogic Role Based Access Control Policy; all authorizations that allow access to these objects/resources is controlled by this policy.

2.6.2 BladeLogic CLI

BladeLogic's CLI allows BladeLogic users to perform most procedures available in Configuration Manager from a command line rather than using the Configuration Manager console. This interface requires the user to authenticate to the BladeLogic Application Server. Once the user has been authenticated, their actions upon objects/resources are controlled via the BladeLogic Role Based Access Control Policy; all authorizations that allow access to objects/resources is controlled by this policy.

2.6.3 BladeLogic Network Shell

Network Shell (NSH) is a network-aware shell that enables cross-platform access through a command line interface. In the evaluated configuration, the NSH directly connects to the RSCD Agent(s) using Self-signed, client-side certs over a TLS connection. The Self-signed, client-side certs enable agents to authenticate NSH clients. To accomplish this, agents are provisioned with SHA1 fingerprints of NSH clients' self-signed certificates. The user may then connect to any server with the BladeLogic agent installed that has been provisioned with the fingerprint of the user client's certificate. Once an NSH client has connected to an agent, the client's authorizations are enforced by configuration files (ACLs) stored on each server.

Note: A default installation of BladeLogic provides no authentication. Instead, this configuration relies on the host operating system of the Network Shell client to authenticate a user.

2.6.4 BladeLogic Application Server

The Application Server is the fundamental component in the BladeLogic Architecture. The Application Server allows users, through the Configuration Manager console to: browse configurations on servers in real time (i.e., snapshot), audit configurations on servers against other servers or against a baseline (i.e., audit job), run compliance policies against servers, install software and patches, and run scripts against servers (outside of the scope of the evaluated configuration). In addition to communicating between the Administrative console components and RSCD Agent servers, the Application Server also controls Configuration Manager's interaction with the BladeLogic Core Database. The BladeLogic Application Server utilizes TLS for session-layer security when communicating with the Configuration Manager and BladeLogic CLI in the client tier and RSCD agents in the server tier.

2.6.5 BladeLogic Core Database and the Reporting Data Warehouse

There are two distinct databases, the core database and the data warehouse. The AppServer accesses the core database, the reports server accesses the data warehouse. Data is aggregated and written to the data warehouse from the AppServer via NSH Script jobs that perform the ETL operations.

2.6.6 BladeLogic Reports Server

BladeLogic Reports Server is a web-based reporting utility that allows users to view pre-existing or ad-hoc reports created by running one of the many types of jobs that BladeLogic is capable of executing. Users access it using a local web browser over Hyper Text Transfer Protocol Secure Socket (HTTPS). BladeLogic Reports Server uses the Application Server to authenticate users, and it reads compliance data, job run data, and user log data (i.e., audit trails) from the Reporting Data Warehouse.

2.6.7 BladeLogic RSCD Agent

An RSCD Agent runs as a daemon (UNIX) or a service (Windows) on all servers managed by BladeLogic. The RSCD Agent software allows a client (BladeLogic Application Server or Network Shell computer) to establish contact with the RSCD Agent computer. The RSCD agent runs commands on behalf of the BladeLogic Application Server and sends the results back to the Application Server. It never initiates a connection to the Application Server, but only communicates when first contacted by the Application Server. The configuration of the RSCD Agent software determines whether a client can establish a connection to the RSCD Agent and what permissions the client will have.

2.7 BladeLogic Operations Manager Version 7.4.2 Evaluation Scope

This section provides information for the purpose of evaluating the TOE. This includes descriptions of the TOE's physical and logical boundaries for the purpose of evaluation.

2.7.1 Physical Boundaries

The physical boundary of the BladeLogic Operations Manager Version 7.4.2 TOE is identified in Figure 1. The components of the TOE have a three-tier architecture that consists of client, server, and middle tiers. The client tier contains the Administration Console that consists of the Configuration Manager, BladeLogic CLI, and Network Shell. The middle tier consists of the BladeLogic Application Server, BladeLogic Reports Server, BladeLogic Core Database, and Reporting Data Warehouse. The server tier consists of the RSCD Agents. Together, these components enforce access control, authorization, authentication, and audit as described in the logical boundary section.

In the evaluated configuration, the Administration Console, BladeLogic Application Server, and BladeLogic Reports Server are installed on Microsoft Windows 2003 servers. For remote access to the Reports Server, a Java enabled internet browser using JRE 1.5 is required. For evaluation purposes, the RSCD Agents are configured on a Windows 2003 Server, Solaris 9.0 Server, and Redhat AS/ES 4.0 Server.

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

BladeLogic Component	Server OS Requirements	Processor/Speed	Memory	Disk Space	Screen
Configuration Manager, Network Shell (NSH), BladeLogic CLI	Windows 2003 Server	• minimum - Intel Pentium III, 500 MHz • recommended - Intel Pentium IV, 2 GHz or better	• minimum - 256 MB • recommended - 512 MB or better	200 MB	• 1024 x 768 • minimum 256 colors

BladeLogic Component	Server OS Requirements	Processor/Speed	Memory	Disk Space	Screen
Reports Server	Windows 2003 Server	minimum - 1 Xeon, 1.5 GHz • recommended - 2 Xeon, 2 GHz or better	• minimum - 1 GB • recommended - 2 GB or better	10 GB	N/A
Application Server	Windows 2003 Server	• minimum - 2 Xeon, 2 GHz • recommended - 4 Xeon, 3 GHz or better	• minimum - 1 GB • recommended - 4 GB	50 GB	N/A
RSCD Agent	Windows 2003 Server, Solaris 9.0 Server, or Redhat AS/ES 4.0 Server		1 MB	10MB	

Table 1 – Server Operating System Requirements for the TOE

For the Operating System, a default installation is used with the standard options. In addition to these operating system requirements, the BladeLogic Core Database and Reporting Data Warehouse requires SQL Server 2000 Standard or Enterprise Editions with Service Pack 3a or more recent, or an Oracle database.

2.7.2 Logical Boundaries

The logical boundary of the TOE includes the following components:

- **Configuration Manager–** a GUI-based system for automating management of remote servers.
- **BladeLogic CLI –** A command line interface that allows access to the Configuration Manager to perform most procedures.
- **Network Shell –** A network-aware shell that allows cross-platform access through a command line interface.
- **BladeLogic Application Server –** A server that provides Configuration Manager access to RSCD Agents and can run ad-hoc and scheduled automation tasks against RSCD agents.
- **BladeLogic Reports Server –** A web-based reporting engine that supplies pre-created and ad-hoc reports on server inventory, compliance, activity, etc.
- **Remote System Call Daemon (RSCD) Agent –** The BladeLogic Agent that runs on all managed servers, providing the ability to manage them remotely.

- **BladeLogic Core Database –** Application interface between the Application Server and the SQL Database. The Core Database is used to store configuration and event data used by other BladeLogic Operations Manager components.

- **BladeLogic Reporting Data Warehouse –** Application interface between the BladeLogic Reports Server. The Reporting Data Warehouse is used to retrieve stored data that contains event information and storage of audit reports.

The following subsections describe the functions provided by the components making up the logical boundary of the TOE.

2.7.2.1 Identification and Authentication

The TOE requires users to provide unique identification and authentication data prior to being granted any administrative access to the system. The TOE enforces a BladeLogic Role Based Access Control Policy, which restricts access to the management functions of the TOE. This protection requires that users of the TOE be authenticated prior to any access to the management functions is granted.

The TOE provides the functionality of counting unsuccessful authentication steps, and when a user meets a specified number they are locked out for a specified amount of time. Additionally, the TOE verifies a user's password meets complexity requirements before it is changed.

The TOE provides two authentication mechanisms, Secure Remote Password and Active Directory / Kerberos. The Configuration Manager contains client- side implementations of these two authentication mechanisms, however, only Secure Remote Password is utilized in the TOE's evaluated configuration. After successfully authenticating a Configuration Manager user, the Application Server issues the client a BladeLogic Single Sign-on (SSO) credential. This SSO credential can be stored in process memory or on the local file system. Once the SSO credential is stored, it can be picked up and used by the Configuration Manager as well as BLCLI for future session establishment. The middle-tier's BladeLogic Application Server and BladeLogic Reports Server contain server-side implementations of these two authentication mechanisms.

SRP users are registered within a BladeLogic users table maintained within the BladeLogic Core Database. These user records include the authenticators used to authenticate users via the SRP authentication protocol.

All client-tier components, the Application Server and BladeLogic Reports Server are part of the TOE. The OS on which each runs is in the IT environment. While the BladeLogic Core Database is part of the TOE, the Database Management System (Microsoft SQL Server) in which the BladeLogic Core Database resides falls within the IT environment.

2.7.2.1.1 Protection of Cached Single Sign-on (SSO) Session Credentials in Process Memory

The Configuration Manager is a multithreaded GUI application that can maintain multiple connections to the BladeLogic Application Server. Whenever a new connection is established, the end user is authenticated to the Application Server using the previously acquired SSO session credential. The client GUI application maintains a copy of the user's SSO credential in process memory. When the process is terminated, the process memory is discarded.

The Configuration Manager relies on the host operating system to protect the confidentiality of the SSO session credential the GUI application caches in its process memory while the application is running and to prevent the leakage of those credentials to unauthorized entities after the application has terminated (e.g., ensuring process virtual memory is cleared).

2.7.2.1.2 Protection of Cached Single Sign-on (SSO) Session Credentials in OS File System

The BladeLogic CLI also supports the caching of a user's SSO session credential in a file stored within the host operating system's file system. The SSO session credential file can be found in the following location in a Windows environment: C:\Documents and Settings\<Windows_user_name>\Application Data\BladeLogic\bl_sesscc. When a CM user authenticates to the BladeLogic Authentication Service, he or she chooses whether or not to write the SSO session credential to a file. BladeLogic CLI will retrieve previously acquired SSO session credentials from the file system. An instance of the CM client application may also use a previously acquired SSO session credential that was written to the file system. BladeLogic relies on the host file system to restrict access to that file system to the user running the BladeLogic CLI application (SSO session credential cache files are written into users' home directories).

2.7.2.1.3 Protection of SRP User Records in Database

The BladeLogic Core Database stores the SRP authenticators in the SQL Server Database which the Application Server requires to authenticate client users. The TOE relies on the DBMS, an element of the IT environment, to protect the integrity and confidentiality of these client authenticators, as well as other user record attributes that may affect an SRP user's ability to log into the system (e.g., lockout status, password expiration time, etc.). Note that the SRP authenticator is essentially a cryptographic hash of the SRP password along with other account data. A user's SRP password cannot be retrieved from the authenticator.

2.7.2.2 Access Control

The TOE enforces a BladeLogic Role Based Access Control (RBAC) Policy, which works with Object based Permissions to restrict access to the management functions of the TOE based on roles and objects. (see section 6.1.2.2 for more information on Object based Permissions)

Once a client-tier user has authenticated to a middle-tier server, an active role must be established for the authenticated identity. The BladeLogic Core Database also records the set of roles in which each registered BladeLogic user is authorized to operate. Subsequent to successful user authentication, client and middle-tier entities negotiate an active role for the current session. Role negotiation is secured via user data protection (encryption of protocol exchanges), employing TLS.

Once access is granted and an authorized role has been negotiated, a user is limited to only management functions that are controlled by the authorizations assigned within the active role.

The TOE's RBAC system provides the ability to define levels of authority for users. A user operating within the RBACAdmins role has the ability to define roles and authorizations and have complete control over the TOE. It is through the use of roles, objects and assigning authorizations to various roles that users are granted access within BladeLogic Operations Manager Version 7.4.2 All discussion of "security privileges" within this document should be understood to mean Roles and their associated authorizations to access various parts of the TOE. Access to resources on managed servers is also under the control of the TOE's RBAC system. The TOE manages access control lists (ACLs) on each of the managed servers in the server tier. RSCD agents (which fall within the TOE) residing on each of these managed servers refer to their centrally managed ACL to determine whether an incoming management command is authorized and what local user to map to (i.e., impersonate) when servicing that management request. Each incoming command identifies the BladeLogic user identity and RBAC role for which the Application Server is issuing the management command. The ACL maps the pairing of incoming user identity and RBAC role to a local user recognized by the managed server's host operating system. The RSCD management agent then impersonates the mapped local user when servicing the particular request. In this way, the TOE leverages the user-based access control mechanisms provided by the managed server's operating system, which is part of the IT environment.

2.7.2.2.1 Protection of Users Role Authorizations and Permissions

The BladeLogic Core Database stores the RBAC roles for which each registered BladeLogic user is authorized to operate. In addition, it stores the collection of permissions granted to each role. The TOE relies on the DBMS, an element of the IT environment, to protect the integrity and confidentiality of this RBAC data.

2.7.2.2.2 Protection of Managed Servers' ACL Files

RSCD Agent ACLs are maintained in a "users" file residing on each managed server. The TOE relies on each managed server's host file system to protect the integrity of the ACL, essentially restricting access to the file to authorized users (typically the local user with Administrator privileges).

2.7.2.3 Audit

The TOE maintains three types of event logs that provide audit trails of system activities. These audit trails can be used to verify policy enforcement, and provide records for computer forensics.

The three sources of event log data are:
Job logs
Authorization logs
Agent logs

Job logs and Authorization logs are maintained in the BladeLogic core database. Agent log data is written to log files on the managed servers; on Unix systems, log data may be output directly to syslog. All three types of log data may be exported to the reporting data warehouse.

2.7.2.3.1 Protection of Audit Data Residing on the DBMS

Audit data is maintained within the SQL Server. The TOE relies on the DBMS hosting the data to protect the integrity and confidentiality of this audit trail data. These Database Management Systems are elements of the IT environment.

2.7.2.3.2 Protection of Agent Log Data Files

The TOE relies on agents' host operating systems (part of the IT environment, and outside of the TOE) to protect the integrity of confidentiality of this audit data prior to being transferred into the Reporting Data Warehouse,

2.7.2.4 Security Management

The TOE is managed through the Configuration Manager and BladeLogic CLI, through which the TOE management can be performed by providing users that are associated with the RBACAdmins role with the ability to manage user roles, authorizations, and manage audit functionality.

2.7.2.4.1 Protection of Persisted Security Data

All security data (class-level and object-level access controls, and user role authorizations) is stored in the SQL Server Database using the BladeLogic Core Database, an element of the TOE. The TOE relies on the SQL Server Database to protect the integrity and confidentiality of this security data residing within the database.

2.7.2.5 Partial TOE Self Protection

The TOE uses the BladeLogic Reports Server to support protection of external TOE communication via a web browser used by Reports Server by performing TLS v1.0 encryption through the Apache OpenSSL-based cryptographic module (mod_SSL). v0.9.7l). The BladeLogic Reports Server resides on a Tomcat Apache server v4.1.31. A username and password request is issued by the web server. The user provides an SRP username and

password to the web server which is passed to the Apache server via an industry standard web browser, either Internet Explorer v6 or Netscape v7. The Reports Server conducts an SRP authentication exchange with the Application Server, using the username and password provided over the web interface. The Application Server will validate the users claimed credentials against password and usernames stored in the SQL Server database. The TOE will return the success or failure of the authentication process. The Apache server is configured to use the strongest form of TLS security and relies on the user's web browser in the IT environment to negotiate the TLS protocol with its associated cryptography to perform the TLS handshake for authenticating the end points of the communication channel and to encrypt the data.

The TOE works with the IT environment (OS and DB) to provide protection of its security functions through non-bypassability and domain separation. All user operations are conducted in the context of an associated session. The TOE manages these sessions to prevent one session from compromising another session. The TOE provides only well-defined interfaces to these sessions, and the sessions allocated only after successful authentication, or when a session is requested from the physically protected local console which is under procedural control. The TOE relies on its platform to operate correctly and to prevent unauthorized access to TOE data and stored executables.

3 Security Environment

3.1 Threats to Security

This section defines the threats to security. The TOE security environment consists of the threats as they relate to the TOE and the IT environment protected by the TOE

3.1.1 Threats addressed by the TOE

The following are threats addressed by 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.

T.MISCONFIG Users, whether they be malicious or non-malicious, could attempt to modify the configuration of remote servers on a local network in an attempt to reduce the security posture of those remote servers.

T.MANAGE An administrator may incorrectly configure the TOE to mismanage user's accounts or adhere to noncompliant security and/or regulatory policies.

T.ACCESS An authorized user of the TOE could gain unauthorized access to resources or information protected by the TOE, or perform operations for which no access rights have been granted, via user error, system error, or other actions.

T.ADMIN_ERROR An administrator may incorrectly install or configure the TOE, or install a corrupted TOE resulting in ineffective security mechanisms.

T.MODIFY Users, whether they be malicious or non-malicious, could attempt to misconfigure or modify their user accounts in an attempt to tamper with TOE resources or modify security information relative to the TOE.

T.PROTECT A user or process may cause, through an unsophisticated attack, TSF data, or executable code to be inappropriately accessed (viewed, modified, or deleted).

T.MASK Users, whether they be malicious or non-malicious, could gain unauthorised access to the TOE by bypassing identification and authentication countermeasures.

T.EAVESDROPPING Malicious users could monitor (e.g., Sniff) network traffic in an unauthorized manner.

T.UNAUTH Users could gain unauthorised access to the web resources by bypassing identification and authentication requirements.

3.2 Secure Usage Assumptions

The specific conditions listed in this section are assumed to exist in the environment in which the TOE is deployed. These assumptions are necessary as a result of practical realities in the development of the TOE security requirements and the essential environmental conditions on the use of the TOE.

Personnel Assumptions:

A.ADMIN One or more authorised administrators will be assigned to install, configure and manage the TOE and the security of the information it contains.

A.NOEVIL Users of the TOE are not careless, wilfully negligent, or hostile and will follow and abide by the instructions provided by the guidance documentation.

A.PATCHES System Administrators exercise due diligence to update the TOE with the latest patches and patch the IT Environment (e.g., OS and database) so they are not susceptible to network attacks.

Logical Assumptions

A.LOCATE The network servers that the TOE will monitor and manage are isolated from any other network, either by physical separation or using logical protection such as a firewall.

Physical Assumptions
None

3.3 Organizational Security Policies

There are NO organizational security policies that apply to the TOE.

4 Security Objectives

This chapter provides a listing of security objectives to ensure that all of the security threats listed in Chapter 3 have been countered. The security objectives are divided into Security Objectives for the TOE (Section 4.1) and Security Objectives for the Environment (Section 4.2).

4.1 Security Objectives for the TOE

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

O.ACCESS The TOE will provide measures to authorize users to access specified TOE resources once the user has been authenticated. User authorization is based on roles configured by the authorized administrator of the TOE.

O.AUDIT The TOE will provide measures for recording security relevant events that will assist the authorized users in detecting misuse of the TOE and/or its security features that would compromise the integrity of the TOE and violate the security objectives of the TOE.

O.AUTH The TOE will provide measures to uniquely identify all users and will authenticate the claimed identity prior to granting a user access to the TOE.

O.COMPLIANCE The TOE will maintain remote server configuration consistent with local security policy including files, registry, and patches.

O.EAVESDROPPING The TOE will encrypt TSF data that traverses the network to prevent malicious users from gaining unauthorized access to TOE data.

O.MANAGE The TOE will provide authorized users with the resources to manage and monitor user accounts, TOE resources and security information relative to the TOE.

O.MONITOR The TOE will monitor remote server configurations to ensure the servers are configured according to the local security policy. The TOE will collect and analyze critical configuration data of remote servers in the IT environment.

O.ROBUST_ADMIN_GUIDANCE The TOE will provide administrators with the necessary information for secure delivery and management.

O.SELF_PROTECTION The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure.

4.2 Security Objectives for the Environment

The following security objectives for the IT environment of the TOE must be satisfied in order for the TOE to fulfill its security objectives.

OE.FILESYS The security features offered by the underlying Operating System and Database protect the files used by the TOE.

OE.TIMESTAMP The runtime environment for the audit mechanism of the TOE must provide a reliable time source for audit record generation.

OE.ADMIN One or more authorized administrators will be assigned to install, configure and manage the TOE and the security of the information it contains.

OE.MANAGE One or more competent individuals will be assigned to manage the TOE and the security of the information it contains.

OE.NOEVIL All users of the TOE are not careless, willfully negligent, or hostile and will follow and abide by the instructions provided by the guidance documentation.

OE.LOCATE The TOE will be located within controlled access facilities that will prevent unauthorized physical access.

OE.PROTECT The operating environment will provide a domain for its own execution that protects itself, its resources, and the TOE from external interference, tampering, or unauthorized disclosure.

5 IT Security Requirements

This chapter identifies the security requirements for the TOE and its environment. The operations performed on security functional and security assurance requirement components contained in this section adhere to the conventions as prescribed in Section 1.5.1 of this ST.

5.1 TOE Security Functional Requirements

The following table provides a summary of the Security Functional Requirements implemented by the TOE.

Security Functional Class	Security Functional Component
Security audit (FAU)	FAU_GEN.1 Audit data generation
	FAU_GEN_EXP.1(1) Audit data generation
	FAU_GEN_EXP.1(2) Audit data generation
	FAU_GEN_EXP.1(3) Audit data generation
	FAU_GEN.2 User identity association
	FAU_SAR.1 Audit review
	FAU_SAR_EXP.1(1) Audit review
	FAU_SAR_EXP.1(2) Audit review
	FAU_SAR_EXP.1(3) Audit review
	FAU_SAR.2 Restricted audit review
	FAU_SAR.3 Selectable audit review
	FAU_SEL.1 Selective audit
Cryptographic support (FCS)	FCS_CKM.1 (1) Cryptographic key generation
	FCS_CKM.1 (2) Cryptographic key generation
	FCS_CKM.4 Cryptographic key destruction
	FCS_COP.1 Cryptographic operation
User data protection (FDP)	FDP_ACC.1 Subset access control
	FDP_ACF.1 Security attribute based access control
Identification and authentication (FIA)	FIA_AFL.1 Authentication failure handling
	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
	FIA_SOS.1 Verification of Secret
Security management (FMT)	FMT_MOF.1 Management of security functions behavior
	FMT_MSA.1 Management of security attributes
	FMT_MSA.2 Secure security attributes
	FMT_MSA.3 Static attribute initialization
	FMT_MTD.1 (1) Management of TSF data
	FMT_MTD.1 (2) Management of TSF data
	FMT_SMF.1 Specification of management functions
	FMT_SMR.1 Security roles
Trusted Path (FTP)	FTP_TRP.1 Trusted Path
Protection of the TSF (FPT)	FPT_RVM_EXP_TOE.1 Non-bypassability of the TSP: TOE
	FPT_ITT.1 Basic internal TSF data transfer protection
	FPT_SEP_EXP_TOE.1 TSF Domain Separation: TOE
Compliance Management (FCM)	FCM_JOB_EXP.1(1) Compliance Management Jobs
	FCM_JOB_EXP.1(2) Compliance Management Jobs

Security Functional Class	Security Functional Component
	FCM_JOB_EXP.1(3) Compliance Management Jobs
	FCM_JOB_EXP.1(4) Compliance Management Jobs
	FCM_JOB_EXP.1(5) Compliance Management Jobs

Table 2 – TOE Security Functional Requirements

The following subsections present the details for each of the TOE Security Functional Requirement components.

5.1.1 Security audit (FAU)

5.1.1.1 FAU_GEN.1 Audit data generation

Hierarchical to: No other components.

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

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

b) All auditable events for the [not specific] level of audit; and

c) [login/logout, and all functions defined on table 3 below].

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

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

b) For each audit event type, based on the auditable event definitions of the functional components included in the ST, [role requesting access to objects/resources, object/resource for which access is being requested, authorization used, message (results of running reports against audited events)].

Dependencies: **FPT_STM.1** Reliable time stamps

Application Note: The above security functional requirement component contains the term "authorization used" The authorization used is more granular definition of functions from table 3. For example Blade.asset.registry.modify would indicate that a user has modified the registry setting of the object named "Blade.asset.object". The field "authorization used" is blank for login and logout attempts.

5.1.1.2 FAU_GEN_EXP.1(1) Audit data generation

Hierarchical to: No other components.

FAU_GEN_EXP.1.1(1) The TSF shall be able to generate a snapshot report based on server objects.

FAU_GEN_EXP.1.2(1) The TSF shall record within each server object snapshot report at least the following information: [Object name, Object type, Date modified].

Dependencies: FPT_STM.1 Reliable time stamps

Application Note : This requirement implements the concept of a snapshot job as defined in section 6.1.6.10.

5.1.1.3 FAU_GEN_EXP.1(2) Audit data generation

Hierarchical to: No other components.

FAU_GEN_EXP.1.1(2) The TSF shall be able to generate a audit report based on the following items listed in the snapshot report, server configuration file:
a) [All objects/resources that were detected to be added, removed, or modified from the baseline or server configuration file].

FAU_GEN_EXP.1.2(2) The TSF shall record within each entry of the compliance report at least the following information:
a) [Date and time of the report creation, remote server host name, remote server host ID, remote server account name responsible for the report creation]; and

b) For each audit report, based on the auditable event definitions of the functional components included in the ST, [object/resource scanned, status, compliance, number of differences, number of violations identified, total number of integrity errors, total number of objects/resources scanned, location of policy file, and location of configuration file].

Dependencies: FPT_STM.1 Reliable time stamps

Application Note : This requirement implements the concept of an audit job as defined in section 6.1.6.9, however the functionality of correcting discrepancies (i.e., a compliance job) is covered in the SFR FCM_JOB_EXP.1.1(1).

5.1.1.4 FAU_GEN_EXP.1(3) Audit data generation

Hierarchical to: No other components.

FAU_GEN_EXP.1.1(3) The TSF shall be able to generate a patch analysis job based on the following server objects:
a) [Application of hotfixes and patches on Windows 2003 servers].

FAU_GEN_EXP.1.2(3) The TSF shall record within each audit record at least the following information:
a) [name, QNumber, bulletin, product, and severity]

Dependencies: FPT_STM.1 Reliable time stamps

Application Note : This requirement implements the concept of a patch analysis job as defined in section 6.1.6.6, however this requirement does not apply the patch. Patches are applied through the use of a deploy job. Deploy jobs are captured in the SFR section FCM_ JOB_EXP.1.1.(3) defined in section 6.1.6.4

5.1.1.5 FAU_GEN.2 User identity association

Hierarchical to:	No other components.
FAU_GEN.2.1	The TSF shall be able to associate each auditable event with the identity of the user that caused the event.
Dependencies:	FAU_GEN.1 Audit data generation FIA_UID.1 Timing of identification

5.1.1.6 FAU_SAR.1 Audit review

Hierarchical to:	No other components.
FAU_SAR.1.1	The TSF shall provide [RBACAdmin, BLAdmin] with the capability to read [role requesting access to objects/resources, user role assignment, role based access authorization result, object/resource for which access is being requested, version of the object/resource 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.
Dependencies:	FAU_GEN.1 Audit data generation

5.1.1.7 FAU_SAR_EXP.1(1) Audit review

Hierarchical to:	No other components.
FAU_SAR_EXP.1.1(1)	The TSF shall provide [*RBACAdmins, BLAdmins*] with the capability to read [*all information collected in FAU_GEN_EXP.1(1)*] from the snapshot reports.
FAU_SAR_EXP.1.2(1)	The TSF shall provide the audit records in a manner suitable for the user to interpret the information.
Dependencies:	FAU_GEN_EXP.1.1(1) Audit data generation
Application Note :	This requirement implements the concept of a snapshot job as defined in section 6.1.6.10.

5.1.1.8 FAU_SAR_EXP.1(2) Audit review

Hierarchical to:	No other components.
FAU_SAR_EXP.1.1(2)	The TSF shall provide [*RBACAdmins, BLAdmins*] with the capability to read [*all information collected in FAU_GEN_EXP.1(2)*] from the compliance reports.
FAU_SAR_EXP.1.2(2)	The TSF shall provide the audit records in a manner suitable for the user to interpret the information.
Dependencies:	FAU_GEN_EXP.1.1(2) Audit data generation

Application Note :	This requirement implements the concept of an audit job as defined in section 6.1.6.9, however the functionality of correcting discrepancies (i.e., a compliance job) is covered in the SFR FCM_JOB_EXP.1.1(1).

5.1.1.9 FAU_SAR_EXP.1(3) Audit review

Hierarchical to:	No other components.
FAU_SAR_EXP.1.1(3)	The TSF shall provide [*RBACAdmins, BLAdmins*] with the capability to read [*all information collected in FAU_GEN_EXP.1(3)*] from the patch analysis jobs.
FAU_SAR_EXP.1.2(3)	The TSF shall provide the audit records in a manner suitable for the user to interpret the information.
Dependencies:	FAU_GEN_EXP.1.1(3) Audit data generation
Application Note :	This requirement implements the concept of a patch analysis job as defined in section 6.1.6.6, however this requirement does not apply the patch. Patches are applied through the use of a deploy job. Deploy jobs are captured in the SFR section FCM_ JOB_EXP.1.1.(3) defined in section 6.1.6.4.
Application Note:	For FAU_SAR_EXP requirements the authorization data is presented in reports through reporting applications.

5.1.1.10 FAU_SAR.2 Restricted audit review

Hierarchical to:	No other components.
FAU_SAR.2.1	The TSF shall prohibit all users read access to the audit records, except those users that have been granted explicit read-access.
Dependencies:	FAU_SAR.1 Audit review
Application Note:	The above security functional requirement component allows read access to the audit records to only those users that have been assigned to specific roles that are maintained by the TOE.

5.1.2 Cryptographic Support (FCS)

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

5.1.2.1 FCS_CKM.1 (1) Cryptographic key generation

Hierarchical to:	No other components.
FCS_CKM.1.1 (1)	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [RSA] and specified cryptographic key sizes [1024 bits] that meet the following: [RFC 2313].

Dependencies:	[FCS_CKM.2 Cryptographic key distribution or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction FMT_MSA.2 Secure security attributes
Application note:	This SFR supports key generation for TLS.

5.1.2.2 FCS_CKM.1 (2) Cryptographic key generation

Hierarchical to:	No other components.
FCS_CKM.1.1 (2)	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [*SRP*] and specified cryptographic key sizes [*2048 MODP group*] that meet the following: [*RFC 5054*].
Dependencies:	[FCS_CKM.2 Cryptographic key distribution or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction FMT_MSA.2 Secure security attributes

5.1.2.3 FCS_CKM.4 Cryptographic key destruction

Hierarchical to:	No other components.
FCS_CKM.4.1	The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [*overwrite method*] that meets the following: [*no standard*].
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FMT_MSA.2 Secure security attributes
Application note:	This SFR supports key destruction for TLS.

5.1.2.4 FCS_COP.1 Cryptographic operation

Hierarchical to:	No other components.
FCS_COP.1.1	The TSF shall perform [encryption and decryption] in accordance with a specified cryptographic algorithm [AES] and cryptographic key sizes [256 bits] that meet the following: [RFC 3268].

Dependencies:	[FDP_ITC.1 Import of user data without security attributes Or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
Application Note:	This SFR supports the symmetric key usage for TLS.

5.1.3 User data protection (FDP)

5.1.3.1 FDP_ACC.1 Subset access control

Hierarchical to:	No other components.
FDP_ACC.1.1	The TSF shall enforce the [BladeLogic Role Based Access Control Policy] on [subjects (BLAdmins, RBACAdmins, additional roles as configured by the RBACAdmins), objects and operations (Read and ModifyACL)].
Dependencies:	FDP_ACF.1 Security attribute based access control

5.1.3.2 FDP_ACF.1 Security attribute based access control

Hierarchical to:	No other components.
FDP_ACF.1.1	The TSF shall enforce the [*BladeLogic Role Based Access Control Policy*] to objects based on the following: [*role*].
FDP_ACF.1.2	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [*the user must be authenticated (via a SSO session credential, previously acquired through SRP) assigned to a role; the user's access to objects/resources is restricted (via an Access Control List) based upon their assigned role; and users can only be associated with one role at any given session*].
FDP_ACF.1.3	The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [*none*].
FDP_ACF.1.4	The TSF shall explicitly deny access of subjects to objects based on the [*none*].
Dependencies:	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialization
Application Note:	The BladeLogic Access Control Policy refers to Role-based Access Control (Roles) and Object based Permissions). See section 6.1.2.1 for Role-based Access Control and section 6.1.2.2 for Object based Permissions.

5.1.4 Identification and authentication (FIA)

5.1.4.1 FIA_AFL.1 Authentication failure handling

Hierarchical to:	No other components.
FIA_AFL.1.1	The TSF shall detect when [an administrator configurable positive integer within [***0-2147483647***] unsuccessful authentication attempts occur related to [***users attempting to authenticate***]].
FIA_AFL.1.2	When the defined number of unsuccessful authentication attempts has been met , the TSF shall [prevent the user from performing activities that require authentication until a defined time period set by the administrator has elapsed].
Dependencies:	FIA_UAU.1 Timing of authentication
Application Note:	The password policy that is part of the evaluated configuration is in section 6.1.7.1.

5.1.4.2 FIA_ATD.1 User attribute definition

Hierarchical to:	No other components.
FIA_ATD.1.1	The TSF shall maintain the following list of security attributes belonging to individual users: [*user identity, authentication data, object-level permissions and authorized role*].
Dependencies:	No dependencies

5.1.4.3 FIA_UAU.2 User authentication before any action

Hierarchical to:	FIA_UAU.1
FIA_UAU.2.1	The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.
Dependencies:	FIA_UID.1 Timing of identification

5.1.4.4 FIA_UAU.7 Protected authentication feedback

Hierarchical to:	No other components.
FIA_UAU.7.1	The TSF shall provide only [*the number of characters typed appearing as asterisks*] to the user while the authentication is in progress.
Dependencies:	FIA_UAU.1 Timing of authentication

5.1.4.5 FIA_UID.2 User identification before any action

Hierarchical to:	FIA_UID.1 Timing of identification
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.
Dependencies:	No dependencies

5.1.4.6 FIA_SOS.1 Verification of Secret

Hierarchical to:	No other Components.
FIA_SOS.1.1	The TSF shall provide a mechanism to verify that secrets meet [complexity requirements as defined in section 6.1.7.1: Password Policy]
Dependencies:	No dependencies

5.1.5 Security management (FMT)

5.1.5.1 FMT_MOF.1 Management of security functions behavior

Hierarchical to:	No other components.
FMT_MOF.1.1	The TSF shall restrict the ability to [**determine the behaviour of, disable, enable, modify the behaviour of**] the functions [event logging] to [users associated with the RBACAdmins role].
Dependencies:	FMT_SMF.1 Specification of management functions FMT_SMR.1 Security roles

5.1.5.2 FMT_MSA.1 Management of security attributes

Hierarchical to:	No other components.
FMT_MSA.1.1	The TSF shall enforce the [BladeLogic Role Based Access Control Policy] to restrict the ability to [**change_default, query, modify, delete,**] the security attributes [role authorization and object-level permissions] to [users associated with the RBACAdmins role].
Dependencies:	[FDP_ACC.1 Subset access control or FDP_IFC.1 Subset information flow control] FMT_SMF.1 Specification of management functions FMT_SMR.1 Security roles FMT_SMR.1 Security roles

5.1.5.3 FMT_MSA.2 Secure security attributes

Hierarchical to:	No other components.

FMT_MSA.2.1	The TSF shall ensure that only secure values are accepted for security attributes.
Dependencies:	ADV_SPM.1 Informal TOE security policy model [FDP_ACC.1 Subset access control or FDP_IFC.1 Subset information flow control] FMT_MSA.1 Management of security attributes

Application Note: This SFR has been added in support of FCS_CKM.1 and FCS_CKM.4 for key generation and destruction. To perform these cryptographic functions, a trusted local admin must connect to the Application Server or the RSCD Agent to run one of the utilities to update the keys stored in the local files.

5.1.5.4 FMT_MSA.3 Static attribute initialization

Hierarchical to:	No other components.
FMT_MSA.3.1	The TSF shall enforce the [*BladeLogic Role Based Access Control Policy*] to provide [***restrictive*****]** default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2	The TSF shall allow the [*users associated with the RBACAdmins role*] to specify alternative initial values to override the default values when an object or information is created.
Dependencies:	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles

5.1.5.5 FMT_MTD.1 (1) Management of TSF data

Hierarchical to:	No other components.
FMT_MTD.1.1(1)	The TSF shall restrict the ability to [**delete]** the [audit records and user job runs] to [users associated with the RBACAdmins role, and other roles as specified].
Dependencies:	FMT_SMF.1 Specification of management functions FMT_SMR.1 Security roles

5.1.5.6 FMT_MTD.1 (2) Management of TSF data

Hierarchical to:	No other components.
FMT_MTD.1.1(2)	The TSF shall restrict the ability to [**perform functions specified in table 3]** the [TSF data specified in table 3] to [default roles specified in table 3, and other roles as specified].
Dependencies:	FMT_SMF.1 Specification of management functions FMT_SMR.1 Security roles

Table 3 - System Management Functions

RBACAdmin	Create	User Roles
RBACAdmin	Modify	User Roles
RBACAdmin	Delete	User Roles
RBACAdmin,/ BLAdmin	Read*	User Roles
RBACAdmin	Create	User
RBACAdmin	Modify (including changing passwords)	User
RBACAdmin	Delete**	User
RBACAdmin,/ BLAdmin	Read*	User
RBACAdmin	Assign Role	User
RBACAdmin	Unassign Role	User
RBACAdmin	Add Authorization	Role
RBACAdmin	Delete Authorization	Role
RBACAdmin	Create	Role
RBACAdmin	Delete**	Role
RBACAdmin	Modify Properties of	Role
RBACAdmin / BLAdmin /Object Level Permissions	Create	Objects
RBACAdmin / BLAdmin / Object Level Permissions	Modify Properties of*	Objects
RBACAdmin / BLAdmin / Object Level Permissions	Delete*	Objects
RBACAdmin / BLAdmin / Object Level Permissions	Read*	Objects
RBACAdmin / BLAdmin / Object Level Permissions	Modify ACL's	Objects

Application Note: Objects are those of which the TOE can manipulate where the RSCD Agent has been installed.

Application Note: Operations with a ** in the center column are audited but not reviewable via the TOE. For review of these audit records an administrator must access the database locally.

Application Note: Operations with a * in the center column are not audited by the TOE.

5.1.5.7 FMT_SMF.1 Specification of management functions

Hierarchical to: No other components.

FMT_SMF.1.1 The TSF shall be capable of performing the following security management functions: [as specified in table 3]

Dependencies: No dependencies

Application Note: Objects include audit records. Modify users includes changing user passwords.

5.1.5.8 FMT_SMR.1 Security roles

Hierarchical to:	No other components.
FMT_SMR.1.1	The TSF shall maintain the roles [RBACAdmins, BLAdmins, additional roles as configured by the RBACAdmins].
FMT_SMR.1.2	The TSF shall be able to associate users with roles.
Dependencies:	FIA_UID.1 Timing of identification
Application Note:	The above security functional requirement component has two roles: RBACAdmins and BLAdmins role, which are default roles that cannot be deleted by anyone and additional roles can be configured by the RBACAdmins on an as needed basis.

5.1.6 Protection of the TSF (FPT)

5.1.6.1 FPT_RVM_EXP_TOE.1 Non-bypassability of the TSP: TOE

Hierarchical to:	No other components.
FPT_RVM_EXP_TOE.1.1	The TSF, when invoked by the underlying host OS, shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.
Dependencies:	No dependencies.

5.1.6.2 FTP_TRP.1 Trusted Path

Hierarchical to:	No other components.
FTP_TRP.1.1	The TSF shall provide a communication path between itself and [remote] users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from [disclosure].
FTP_TRP.1.2	The TSF shall permit [the remote users] to initiate communication via the trusted path.
FTP_TRP.1.3	The TSF shall require the use of the trusted path for [initial user authentication, user management].
Dependencies:	No dependencies
Application note:	This SFR is included to capture TLS encryption functionality.

5.1.6.3 FPT_ITT.1 Basic internal TSF data transfer protection

Hierarchical to: No other components.

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

Dependencies: No dependencies

Application Note: This SFR is included to capture SRP and TLS encryption functionality.

5.1.6.4 FPT_SEP_EXP_TOE.1 TSF domain separation: TOE

Hierarchical to: No other components.

FPT_SEP_ EXP_TOE.1.1 The TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects initiating actions through its own TSFI.

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

Dependencies: No dependencies.

5.1.7 Compliance Management (FCM)

5.1.7.1 FCM_JOB_EXP.1 (1) Compliance Management Jobs

Hierarchical to: No other components.

FCM_JOB_EXP.1.1(1) The TSF shall be able to forcibly correct any deviations between an audit report and a snapshot report.

Dependencies: FAU_GEN_EXP.1.1(2)
FCM_JOB_EXP.1.1(3)

Application Note: This requirement is used to capture the functionality of Compliance Job described in section 6.1.6.8. Compliance Jobs examines the template parts specified in the compliance rules and compares them to the rules that have been defined.

5.1.7.2 FCM_JOB_EXP.1 (2) Compliance Management Jobs

Hierarchical to: No other components.

FCM_JOB_EXP.1.1(2) The TSF shall be able to deploy (or push) multiple files and directories to one or more managed servers.

Dependencies: No Dependencies

Application Note: This requirement is used to capture the functionality of File Deploy described in section 6.1.6.1. File Deploy Jobs rely on the ability to push files to remotely managed servers therefore making the requirement a dependency to the requirement above.

5.1.7.3 FCM_JOB_EXP.1 (3) Compliance Management Jobs

Hierarchical to: No other components.

FCM_JOB_EXP.1.1(3) The TSF shall be able to execute deployed (or pushed) content to one or more managed servers unattended.

Dependencies: FCM_JOB_EXP.1.1(2)

Application Note: This requirement is used to capture the functionality of Deploy Jobs described in section 6.1.6.4. Deploy Jobs rely on the ability to push files to remotely managed servers therefore making the requirement a dependency to the requirement above. Users cannot remotely execute files that are not a part of the package of files sent to the remotely managed server.

5.1.7.4 FCM_JOB_EXP.1 (4) Compliance Management Jobs

Hierarchical to: No other components.

FCM_JOB_EXP.1.1(4) The TSF shall be able to allow for the deployment and execution of previously saved network shell scripts.

Dependencies: FCM_JOB_EXP.1.1(2)

Application Note: This requirement is used to capture the functionality of Network Shell Jobs described in section 6.1.6.5. Network Shell Jobs run scripts or commands on one or more servers.

5.1.7.5 FCM_JOB_EXP.1 (5) Compliance Management Jobs

Hierarchical to: No other components.

FCM_JOB_EXP.1.1(5) The TSF shall be able to concatenate a series of deploy jobs, file deploy jobs, and network shell jobs, as well as deploy a series of BLPackages.

Dependencies: FCM_JOB_EXP.1.1(2)

Application Note: This requirement is used to capture the functionality of Batch Jobs described in section 6.1.6.3. Batch Jobs are the common way the TOE will perform patch management.

5.2 TOE Security Assurance Requirements

This section identifies the Security Assurance Requirement components met by the TOE. These assurance components meet the requirements for EAL3 augmented with ALC_FLR.1.

5.2.1 Configuration Management (ACM)

5.2.1.1 Authorisation controls (ACM_CAP.3)

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

5.2.1.2 TOE CM Coverage (ACM_SCP.1)

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

5.2.2 Delivery and operation (ADO)

5.2.2.1 Delivery procedures (ADO_DEL.1)

ADO_DEL.1.1D	The developer shall document procedures for delivery of the TOE or parts of it to the user.
ADO_DEL.1.2D	The developer shall use the delivery procedures.
ADO_DEL.1.1C	The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.
ADO_DEL.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

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

ADO_IGS.1.1D	The developer shall document procedures necessary for the secure installation, generation, and start-up of the TOE.
ADO_IGS.1.1C	The installation, generation and start-up documentation shall describe all the steps necessary for secure installation, generation, and start-up of the TOE.
ADO_IGS.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADO_IGS.1.2E	The evaluator shall determine that the installation, generation, and start up procedures result in a secure configuration.

5.2.3 Development (ADV)

5.2.3.1 Informal functional specification (ADV_FSP.1)

ADV_FSP.1.1D	The developer shall provide a functional specification.
ADV_FSP.1.1C	The functional specification shall describe the TSF and its external interfaces using an informal style.
ADV_FSP.1.2C	The functional specification shall be internally consistent.
ADV_FSP.1.3C	The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing details of effects, exceptions and error messages, as appropriate.
ADV_FSP.1.4C	The functional specification shall completely represent the TSF.
ADV_FSP.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_FSP.1.2E	The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

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

ADV_HLD.2.1D	The developer shall provide the high-level design of the TSF.
ADV_HLD.2.1C	The presentation of the high-level design shall be informal.
ADV_HLD.2.2C	The high-level design shall be internally consistent.
ADV_HLD.2.3C	The high-level design shall describe the structure of the TSF in terms of subsystems.
ADV_HLD.2.4C	The high-level design shall describe the security functionality provided by each subsystem of the TSF.
ADV_HLD.2.5C	The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.
ADV_HLD.2.6C	The high-level design shall identify all interfaces to the subsystems of the TSF.
ADV_HLD.2.7C	The high-level design shall identify which of the interfaces to the subsystem of the TSF are externally visible.
ADV_HLD.2.8C	The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing details of effects, exceptions and error messages, as appropriate.
ADV_HLD.2.9C	The high-level design shall describe the separation of the TOE into TSP-enforcing and other subsystems.
ADV_HLD.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_HLD.2.2E	The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

5.2.3.3 Informal correspondence demonstration (ADV_RCR.1)

ADV_RCR.1.1D	The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided.
ADV_RCR.1.1C	For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.
ADV_RCR.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.4 Guidance documents (AGD)

5.2.4.1 Administrator guidance (AGD_ADM.1)

AGD_ADM.1.1D	The developer shall provide administrator guidance addressed to system administrative personnel.
AGD_ADM.1.1C	The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.
AGD_ADM.1.2C	The administrator guidance shall describe how to administer the TOE in a secure manner.
AGD_ADM.1.3C	The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.
AGD_ADM.1.4C	The administrator guidance shall describe all assumptions regarding user behaviour that are relevant to secure operation of the TOE.
AGD_ADM.1.5C	The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.
AGD_ADM.1.6C	The administrator guidance shall describe each type of security relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.
AGD_ADM.1.7C	The administrator guidance shall be consistent with all other documentation supplied for evaluation.
AGD_ADM.1.8C	The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.
AGD_ADM.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.4.2 User guidance (AGD_USR.1)

AGD_USR.1.1D	The developer shall provide user guidance.
AGD_USR.1.1C	The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.
AGD_USR.1.2C	The user guidance shall describe the use of user-accessible security functions provided by the TOE.
AGD_USR.1.3C	The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.
AGD_USR.1.4C	The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behaviour found in the statement of TOE security environment.
AGD_USR.1.5C	The user guidance shall be consistent with all other documentation supplied for evaluation.
AGD_USR.1.6C	The user guidance shall describe all security requirements for the IT environment that are relevant to the user.
AGD_USR.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.5 Life cycle support (ALC)

5.2.5.1 Identification of security measures (ALC_DVS.1)

ALC_DVS.1.1D	The developer shall produce development security documentation.
ALC_DVS.1.1C	The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.
ALC_DVS.1.2C	The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.
ALC_DVS.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ALC_DVS.1.2E	The evaluator shall confirm that the security measures are being applied.

5.2.5.2 Basic flaw remediation (ALC_FLR.1)

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

5.2.6 Tests (ATE)

5.2.6.1 Analysis of coverage (ATE_COV.2)

ATE_COV.2.1D	The developer shall provide an analysis of the test coverage.
ATE_COV.2.1C	The analysis of the test coverage shall demonstrate the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.
ATE_COV.2.2C	The analysis of the test coverage shall demonstrate that the correspondence between the TSF as described in the functional specification and the tests identified in the test documentation is complete.
ATE_COV.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.6.2 Testing: High-level design (ATE_DPT.1)

ATE_DPT.1.1D	The developer shall provide the analysis of the depth of testing.
ATE_DPT.1.1C	The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design.
ATE_DPT.1.2E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.6.3 Functional testing (ATE_FUN.1)

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

ATE_FUN.1.2D The developer shall provide test documentation.

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

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

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

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

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

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

5.2.6.4 Independent testing - sample (ATE_IND.2)

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

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

ATE_IND.2.2C The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

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

ATE_IND.2.2E The evaluator shall test a subset of the TSF as appropriate to confirm that the TOE operates as specified.

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

5.2.7 Vulnerability assessment (AVA)

5.2.7.1 Examination of guidance (AVA_MSU.1)

AVA_MSU.1.1D The developer shall provide guidance documentation.

AVA_MSU.1.1C The guidance documentation shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AVA_MSU.1.2C The guidance documentation shall be complete, clear, consistent and reasonable.

AVA_MSU.1.3C The guidance documentation shall list all assumptions about the intended environment.

AVA_MSU.1.4C The guidance documentation shall list all requirements for external security measures (including external procedural, physical and personnel controls).

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

AVA_MSU.1.2E The evaluator shall repeat all configuration and installation procedures to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.

AVA_MSU.1.3E The evaluator shall determine that the use of the guidance documentation allows all insecure states to be detected.

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

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

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

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

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

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

5.2.7.3 Developer vulnerability analysis (AVA_VLA.1)

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

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

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

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

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

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

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

5.3 Environment Security Functional Requirements

This section identifies the security functional requirements that have been levied to the IT environment and must be enforced in order for the TOE to securely enforce its stated functional claims. These Security Functional Requirements are identified in the following table.

Security Functional Class	Security Functional Component
Security audit (FAU)	FAU_STG.1 Protected audit trail storage
Protection of the TSF (FPT)	FPT_RVM_EXP.1 Non-bypassability of the TSP: IT Environment
	FPT_SEP_EXP.1 TSF domain separation: IT Environment
	FPT_STM.1 Reliable time stamps

Table 4 – Environment Security Functional Requirements

The following subsections present the details for each of the IT Environment Security Functional Requirement components.

5.3.1 Security audit (FAU)

5.3.1.1 FAU_STG.1 Protected audit trail storage

Hierarchical to: FAU_STG.1

FAU_STG.1.1 The **IT Environment** shall protect the stored audit records from unauthorized deletion.

FAU_STG.1.2	The **IT Environment** shall be able to [**prevent**] unauthorized modifications to the audit records in the audit trail.
Dependencies:	FAU_GEN.1 Audit data generation
Application Note:	This requirement represents the data (snapshot records, audit records, audit trails, patch analysis reports, configuration settings, etc. stored in the SQL Server database.

5.3.2 Protection of the TSF (FPT)

5.3.2.1 FPT_RVM_EXP.1 Non-bypassability of the TSP: IT Environment

Hierarchical to:	No other components.
FPT_RVM_EXP.1.1	The security functions of the TOE server OS shall ensure that TOE server OS security policy enforcement functions are invoked and succeed before each function within the scope of control of the TOE server OS is allowed to proceed.
Dependencies:	No dependencies.

5.3.2.2 FPT_SEP_EXP.1 TSF domain separation: IT Environment

Hierarchical to:	No other components.
FPT_SEP_EXP.1.1	The security functions of the TOE server OS shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects in the scope of control of the TOE server OS.
FPT_SEP_EXP.1.2	The security functions of the TOE server OS shall enforce separation between the security domains of subjects in the scope of control of the TOE server OS.
Dependencies:	No dependencies

5.3.2.3 FPT_STM.1 Reliable time stamps

Hierarchical to:	No other components.
FPT_STM.1.1	The **IT Environment** shall be able to provide reliable time-stamps for **use by the TOE**.
Dependencies:	No dependencies

6 TOE Summary Specification

6.1 TOE Security Functions

This section describes the security functions provided by the TOE.

6.1.1 Audit

Operations Manager provides a GUI interface to select an object to see an audit trail of all changes to that object that is restricted to authorized roles. This allows for searching to be performed by having the Reports Server filter on the level of objects that are produced in the audit trails. An audit trail is a record of who has sought authorization for specific actions. Users can also run ad hoc reports in order to view the audit trail. Ad hoc reports create custom report definitions and are run from the Reports Server. The user can specify in the RBAC Manager workspace whether an audit trail entry is recorded every time a user is successfully authorized for an action, every time a user is denied authorization, or both. Audit trail settings apply globally. The audit trail also records the role that was trying to access the object, and the user using the role in the current session. The audit trails are then stored in the SQL Server Database via the BladeLogic Core Database or the Reporting Data Warehouse. Audit trails are meant to capture user activity using the TOE which is different than the output of a job executed by a user of the TOE.

6.1.1.1 Audit Trail

All actions in BladeLogic must be authorized, and an audit trail entry can be generated for each authorization request. The user can specify whether an audit trail entry is recorded every time a user is successfully authorized for an action, every time a user is denied authorization, or both. It must be noted that an audit trail entry is also recorded when there are changes to the role definitions themselves. In this way, BladeLogic can maintain an audit trail for changes to all system objects maintained within the Application Server. Each audit trail entry records the following information:

Role trying to access a resource
User who has assumed a role in the current session
Authorization requested
Status (success or failure)
A time stamp (reliable time-stamps are provided by the IT environment).

Note that audit trail records only capture information about users seeking authorization; they do not capture changes to the object itself. Employing the CM GUI, authorized users can view audit trails for changes to all system objects.

Authorization log data within the core database gets moved to the data warehouse, where it is available to reports users. This is done via scripts that are run from the Application Server.

6.1.1.2 Agent Logs

BladeLogic RSCD Agents implement a rolling event log mechanism for recording all activities conducted by it in response to requests from BladeLogic Application Servers and NSH clients.

As agent logs are typically written to files on the managed servers. Agent log data may be copied to the reports server where it is used to generate server usage reports.

6.1.2 Access Control

The TOE uses a system of role-level and object-level authorizations that grant permissions to perform actions on objects. Users are assigned to roles and granted the permissions defined for each respective role. A user can have many roles, but a user can only assume one role at a time.

In addition to role-based authorizations, the Configuration Manager is used to grant authorizations to roles to perform specific actions on objects. To take an action on any object, the user must have role level authorization as well as object-level authorization, which allows for fine-grained access to objects throughout the system.

The TOE enforces access to objects via the Configuration Manager, BLCLI, and the Network Shell enabled consoles. The Configuration Manager and BLCLI provide an interface for setting role-level authorizations and server permissions. Once these authorizations and permissions have been defined, the Application Server converts the role and object-level authorizations to Agent ACLs. The content within the Agent ACLs is then pushed to the RSCD Agents through an ACL push job, where the information is converted into configuration file entries. The Agent configuration file entries restrict all incoming connections and user access to the RSCD Agent.

Configuration files are updated by local admins through the various admin utilities. Additionally, the admin utilities are used for access control to the RSCD Agent. The utilities used in the evaluated configuration include the following:

Bl_gen_ssl - This utility is used by the Network Shell client and the Application Server clients to create an id.pem file, which contains the client's self-signed certificate and the private key associated with the certificate. These are used for communication with the RSCD Agent. Once the id.pem file is created, the passphrase that is used to encrypt the private key must be added to the securecert file on the Application Server and on the Network Shell clients. This is done through the secadmin utility. For more information on the bl_gen_ssl utility, see pg. 119 of the BladeLogic Administrative Guide v7.4.2.

Secadmin – This utility is used for updating the secure and securecert files on the Application Server and Network Shell clients, the secure file on the Reports Server and the secure file on the RSCD Agent. Administrators can create, modify, or delete entries in the secure file for a host. They can also create, modify, or delete default or rscd entries in the secure file. The secure file can be edited by hand, but BladeLogic recommends that the secadmin utility is always used to ensure the file is formatted correctly. Administrators can modify entries in the securecert file. The securecert file is updated to include an encoded copy of the passphrase in the id.pem file.

Putcert - This command is used from the Network Shell and Application Server clients to create or update a fingerprint file on the targeted Agent. On a Windows machine, the fingerprint file for a Window Application Server is: C:\Program Files\BladeLogic\RSC\certs\SYSTEM.

Default and implicit behaviors for each RSCD Agent include the following:

- All clients are granted read/write access to all servers running the RSCD Agent.
- All clients and servers are set to communicate using secure communication based on TLS. TLS uses AES encryption.
- Users are granted permissions on servers through a process of user privilege mapping. When a user attempts to connect to an agent, he or she is mapped to a local user on the

agent host, typically the administrator or root account. The local user to which the end user is mapped must have the local permissions necessary to conduct the configuration management activities initiated by the end user.

By default there are two built-in roles as follows (see Table 3 for System Management functions):

- **RBACAdmins**—Built-in authorizations grant the RBACAdmins role the authorizations to read and modify the Access Control List (ACL) for all objects. This allows the RBACAdmins role to always have access to any object, and even if all roles that are granted access to an object are deleted, the RBACAdmins role can still modify that object's authorizations so other roles can then access the object. In addition, out-of-box authorizations grant the RBACAdmins role the authority to perform any actions relating to roles, users, ACL, and assigning authorizations to roles. The RBACAdmins cannot be deleted.

- **BLAdmins**—The default authorizations for the BLAdmins role allows the user to change the default objects, modify and delete any object except for roles, assigning authorizations to roles, and event logging configuration settings. The BLAdmins role cannot be deleted.

The following table summarizes the authorizations granted to RBACAdmins and BLAdmins:

Default Role	Built-in Authorizations	Out-of-box Authorizations
RBACAdmins	• Granted Read authorization on all objects (to include audit records) • Granted ability to modify the ACL authorization on all objects in the Operations Manager • The above authorizations are built-in and cannot be modified.	• Granted authorization for all objects relating to RBAC. • Granted Server. Authorized to push ACLs to servers. • The above authorizations can be modified as necessary.
BLAdmins	• Granted Read authorization on all objects (to include audit records). • Granted ability to Read the ACL authorization on all objects in the Operations Manager • The above authorizations are built-in and cannot be modified.	• Granted authorization on all classes of objects, except the following: > only has read-only access to roles > only has read-only access to authorization profiles >cannot modify the download locations for Windows patch analysis configurations • The above authorizations can be modified as necessary.

Table 5 – RBACAdmins and BLAdmin Authorizations

The TOE supports authorization via Access Control Lists (ACL) and a Role-Based Access Control (RBAC) model. Authorizations are defined for each respective role to which users are assigned. In addition, objects (i.e., servers, jobs, system packages, etc.) also receive permissions and must therefore have role and object level authorizations. The TOE defines the policy for access to objects throughout the TOE interfaces.

6.1.2.1 Role Based Access Control

The TOE supports authorization via ACLs and a Role Based Access Control (RBAC) model. Role-based access control (RBAC) limits the actions users can perform on a system-wide basis. RBAC administrators grant access permissions to users by defining a set of authorizations combined in a role, and assigning users to those roles. A user

can have many roles, but a user can only assume one role at a time. A role can grant access to specified severs, authorize users to perform actions, and manage authorizations. After using RBAC to define roles and users, administrators can push those authorizations to agents on remote servers, thereby restricting access to those servers. On remote servers, RBAC converts permission information into entries in the TOE's configuration files. The configuration files are ACLs that define user access to an agent.

6.1.2.2 Object Based Permissions

BladeLogic allows users to assign permissions directly to objects. BladeLogic's object-based permissions are a form of discretionary access control (DAC). DAC mechanisms allow users to grant or revoke access to objects under their control without the intercession of a system administrator. In BladeLogic, object-level permissions are granted to specific roles, not users. To accomplish this an ACL is defined for every system object created in BladeLogic. The ACL specifies which roles are granted access to the object and what types of actions those roles can perform on the object. Object-level permissions allow BladeLogic users to delegate authority for managing different objects within BladeLogic to different roles.

The combination of role-based and object-based authorizations determines a user's *effective authorizations* – that is, the actions a role can actually perform on a system object. A role can only perform an action that is authorized at both the role level and the object level.

6.1.3 Security Management

The TOE provides multiple levels of security. The TOE supports authorization via a RBAC model and a set of ACLs. Authorizations are assigned to a role, and then users are assigned to that role. The respective role then grants the users the permissions defined for the role.

6.1.3.1 Configuration manager

The Configuration Manager provides the tools to manage the activities throughout the data center. The Configuration Manager allows the system administrator to create, modify, delete, read, and rename objects used to perform tasks. The Configuration Manager also allows the system administrator to manage applications, software packages, compliance, and remote administration.

6.1.4 Self Protection

The Self-protection function is responsible for providing an execution domain that is protected from interference and tampering by unauthorized users. The TOE is an application running on a dedicated device that executes all of its processes internally and is only accessible through the defined interfaces. In addition, only authorized users and the host IT environment are able to modify the functionality of the TOE. The RSCD Agent interface enforces domain separation in that any data sent to or from this interface is logically separated from all other TOE data. Therefore, the data is never executed but rather parsed for analysis. At all physical interfaces, the TOE intercedes to ensure domain separation. Traffic can only come into the TOE via the Configuration Manager, BLCLI, or the Network Shell console interfaces which require users to authenticate. Once the user has authenticated their actions upon objects/resources are controlled via the BladeLogic access control policy; all access to objects/ resources are controlled by this policy.

The Network Shell can be launched as a standalone shell on an administrators work station or through the Configuration Manager console. In the evaluated configuration, the Network Shell is launched as a standalone shell. The Reports Server interface is accessed using local web browsers and protected via username and password. The SQL Server Database underlying Operating System interfaces are (only accessible by internal TOE processes). Unauthorized users cannot bypass the identification and authentication mechanisms, preventing interference and tampering by untrusted subjects and thereby maintaining a domain for its own execution. The self protection

function of the TOE and the self protection features of the host IT environment work together to satisfy the self protection requirements. The reliable time value is received upon boot up or modified via a trusted channel from the host IT Environment. The host IT Environment mediates its interfaces to only allow authorized modifications while protecting those interfaces from interference and tampering. For example, once the clock is initialized or modified via the trusted interface the IT Environment ensures those interfaces are free from interference and tampering.

6.1.5 Trusted Communications

The TOE uses TLS for remote access to the Reports Server, internal communications between the client tier applications (i.e., Configuration manager and BladeLogic CLI) and the middle-tier Application Server, and for internal communications between the Application Server and the RSCD Agents. The TLS v1.0 protocol includes functionality for data encryption, server authentication, message integrity, and client authentication. The TOE provides for the secure passage of data between a client and a server running Hypertext Transfer Protocol (HTTP), sometimes referred to as HTTPS. For remote access to the Reports Server, a Java enabled internet browser using JRE 1.5 is required. Currently the TOE supports Internet Explorer v6.0 (for Windows clients) and Netscape v7.0 (for UNIX and Windows clients).

6.1.5.1 Secure Remote Password

Secure Remote Password (SRP) is used by the TOE as the authentication mechanism in the evaluated configuration. Secure Remote Password (SRP) is a protocol used for integrating secure password authentication into networked applications by using a password and key exchange for authentication. Additionally, the SRP protocol resists dictionary attacks by network intruders and protects past sessions and passwords against future compromise. This is achieved by elaborate hashes of the user's password, known as authenticators, being stored in the database. Therefore, an attacker who captures the password database cannot use it to compromise security.

SRP is one of the authentication mechanisms supported by the BladeLogic Authentication Service. After successfully authenticating an end user, the Authentication Service issues the client application a single sign-on session credential, which the client then uses to establish authenticated application layer sessions with the Application Server. The Authentication Service listens on port 9840; the Application Service listens on port 9841.Communications to both of these ports is protected via TLS.

6.1.5.2 BladeLogic SRP Protocol Implementation

The TOE uses the SRP protocol for authenticating users to the BladeLogic Authentication Service.

The SRP protocol (see RFC 2945) specifies two parameters, N and g (modulus and generator, respectively), to be used within the Diffie Hellman exchange. The TOE meets the following requirements for N and g:

The values of N and g used in this protocol must be agreed upon by the two parties in question. They are set in advance, on the server application and supplied to the client tier. The host sends the parameters in the first message along with the salt. For maximum security, N is a safe prime (i.e. a number of the form N = 2q + 1, where q is also prime). Also, g is a generator modulo N, which means that for any X where 0< X < N, there exists a value x for which g^x % N == X.

For the SRP implementation, the TOE uses the modulus and generator (N,g) the MODP DH group 14 defined in RFC 3526. This is a well-known, 2048-bit MODP DH group defined for use with IKE, and appropriate for 128-bit symmetric key agreement.

The SRP protocol incorporates the use of a hash algorithm. The RFC's protocol description references SHA1, but states that an alternative hash algorithm may be used. The TOE employs SHA1.

The SRP protocol also calls for the client and server to generate random values, a and b, to be used as their respective private exponents. Per the recommendation of RFC 3526 (if the strength of the selected MODP Diffie Hellman group is 128-bits, then the size of the randomly selected exponent should be a least 256 bits), the TOE uses a length of 256 bits for each of these random exponents.

6.1.5.3 BladeLogic TLS Protocol Implementation

The TOE uses TLS for securing communications between clients and the Application Server, between Application Server and RSCD agents, and between web browsers and the BladeLogic Reports Server.

For TLS, the TOE employs the TLS_RSA_WITH_AES_256_CBC_SHA cipher suite; that is…
RSA key negotiation
256-bit AES block encryption algorithm
CBC (Cipher Block Chaining) block cipher mode
SHA1 HMAC construction for integrity protection

6.1.6 Operations Manager

Operations Manager lets IT organizations automate the management of enterprise-class data centers. Using Operations Manager, administrators can inventory servers and applications, manage patches, measure and enforce compliance to organizational standards, administer all configuration changes to servers and applications, provision servers with applications and operating systems, and perform many other data center automation tasks. Together these capabilities allow for the management of the complete life cycle of servers and applications in the data center.

Operations Manager consists of three modules:

- Discover – Read-only view that allows for the determination of current server software and system configurations (i.e. snapshot jobs).

- Compliance – Read-only view of server configurations used to compare compliance with an existing standard or "gold" server or to compare configurations to some type of security, operations, build, patch, or regulatory policy (i.e. audit jobs, patch analysis jobs, compliance jobs).

- Configuration - Provisions new servers or propagates configuration changes throughout a server environment. This module can be used for bare metal server provisioning, structured configuration changes (such as a packaged software installations), and ad hoc system configuration changes (such as editing a file on a live server). The Configuration module can be valuable for adding new server capacity, installing or upgrading new software, remediating or synchronizing a server with a compliance policy or "gold server" configuration, and performing any type of ad hoc configuration change that a server's operating system requires (i.e. deploy jobs, file deploy jobs, network shell script jobs).

6.1.6.1 File Deploy Job

File Deploy Jobs allow the user to deploy or push multiple files and directories to one or more servers. When the user deploys a directory, the contents of the directory are copied recursively, meaning that all sub-directories and their contents are also deployed. In order to deploy files already stored in the depot, the file must be bundled as a BLPackage and the al BLPackage Deploy Job must be used to deploy the BLPackage. Deploying files as

BLPackages provides far more control over a job, including the ability to simulate its deployment, automatically roll the job back when a failure occurs, and manually undo the job.

6.1.6.2 BLPackages

A BLPackage is an aggregation of many types of server objects that are packaged together so they can be deployed unattended on multiple remote hosts. A BLPackage can be made from bundling server objects or audit reports. BLPackages can consist of server objects from Windows or UNIX, but the server objects from both Operating Systems cannot be combined.

6.1.6.3 Batch Job

A Batch Job is a concatenated series of Deploy Jobs, File Deploy Jobs, and Network Shell Jobs. Batch Jobs are useful when administrators must perform multiple discrete jobs. For example, a Batch Job can deploy a series of BLPackages to update a distributed application that consists of components running on database, application and Web servers. A Batch Job can include a Network shell Script that reboots Windows servers. Batch Jobs are the common way the TOE will perform patch management.

6.1.6.4 Deploy Job

Deploy Jobs is the deployment of software packages or a BLPackage to one or more remote servers. Both software packages and BLPackages are executable packages that can be deployed unattended. A Deploy job can also be used to uninstall software packages. The Deploy Job pushes a software package to servers where the uninstall should occur and then runs an uninstall command. A File Deploy Job can be used if just deploying files and directories. However, bundling files and directories as BLPackages and using a Deploy Job to deploy them gives more control over a job, including the ability to simulate its deployment, automatically roll the job back when a failure occurs, and manually undo the job.

6.1.6.5 Network Shell Jobs

The Network Shell is a network scripting language that enables cross platform access through a command line interface. Network Shell Script Jobs allows the deployment and execution of previously saved Network Shell Scripts. A Network Shell Script runs scripts or commands on one or more severs.

6.1.6.6 Patch Management

Patch Management determines the Windows patches and service packs that should be installed on a server, including patches and service packs for both Microsoft operating systems and other Microsoft products such as SQL Server and Office. Patch Management uses File Deploy Jobs, Deploy jobs, Batch Jobs and Network Shell Script Jobs to deploy and execute patches and hotfixes. When a Patch Analysis Job is performed, the administrator downloads the latest patch information files (XML files) from Shavlik servers and stores the files on the Application Server's local file system. The administrator also specifies the location on the file system for the TOE to check for the patch information files. These files are then used to define the correct patch configuration of any servers that are monitored for compliance. After running a Patch Analysis Job, the results will be provided in a human readable format which is suitable for an administrator to review. This Patch Analysis information (including all patch properties) is displayed in a tabbed dialog.

6.1.6.7 Patch Management and Assurance Continuity

Patch Management possibly will change the IT Environment that the TOE will operate in, thus leaving the TOE in an unknown state. To mitigate this risk, BladeLogic will use the assurance continuity program to assert that hotfixes and service packs do not interfere with the TOE's ability to continue to meet the security functional requirements as stated in this ST.

6.1.6.8 Compliance Jobs

Compliance specifies a subset of the component template parts and defines compliance rules that these parts must satisfy. Each compliance rule can include a set of instructions explaining what actions to take if a rule is not satisfied. One possible action is the deployment of a BLPackage to correct the problematic configuration. After a component template is complete, a Compliance Job can be run to monitor the configuration of a component. For each component, the Compliance Job examines the template parts specified in the compliance rules, comparing them to the rules that have been defined. When a component fails to satisfy the compliance rules and remediation is enabled, a BLPackage can be deployed to correct the problem. Remediation can occur automatically, as part of a Compliance Job, or Compliance Job results can be examined and an administrator can manually remediate compliance rule failures. Typically, Operations Manager users employ Compliance Jobs to ensure consistency with compliance policies.

6.1.6.9 Audit Jobs

When users run jobs to that generate reports, the Core BladeLogic Database stores that data to the SQL Server Database. An Audit Job compares servers or snapshots to determine whether their configurations match a standard configuration. Audit jobs can quickly identify discrepancies between server configurations. When a discrepancy is identified the necessary changes detected in the Audit Report can be deployed to a server so its configuration matches the standard. Audits Jobs can also perform a security function by quickly identifying unauthorized changes to server configurations. Performing an Audit Job requires a "master" server with a standard configuration that is used as the basis of comparison. The procedure for identifying a master server depends on how the Audit Job is defined. If the Audit Job is defined by selecting live server objects a server or a snapshot must be identified as the "master". If the Audit Job is defined by selecting one or more components or snapshots, then one or more components or snapshots, together, act as a "master". An Audit job can also be defined as an entire server or components of a server. Components are managed objects that provide a higher level of abstraction than other server objects in the TOE. A component can specify the files, configuration entries, and registry values needed to support a server. Server objects are any manageable configuration option of the remote servers; they consist of files, directories, registry keys, configuration settings, etc. After running an Audit Job, the results will be provided in a human readable format which is suitable for an administrator to review. Audit Job results are displayed in the form of a hierarchical tree beneath the Audit Job in the Jobs workspace. Selecting nodes in the left pane displays audit information in the contents pane on the right.

6.1.6.10 Snapshot

A Snapshot is a record of how a server is configured at a point in time. Snapshots allow administrators to audit the configuration of servers. Snapshots can be based on components (i.e., specified server objects) or the all the server objects. A Patch analysis can also be conducted through the comparisons of the snapshots. The Patch Analysis determines which Windows service packs or hotfixes are needed on the server or components being audited.

To take a snapshot, an administrator must be granted SnapshotJob.Execute and SnapshotJob.Read (a role cannot execute a job without being able to read the job). Through the Configuration Manager, an administrator can run a snapshot job by drilling down through Server workspace/Component workspace/Component Template/Jobs workspace/Snapshot Job wizard. When a Snapshot Job is defined, a snapshot of components or live server objects can be taken.

Snapshot Jobs are stored in the Jobs workspace. Snapshot Jobs which have run at least once are also accessible from the Snapshot Results nodes for the associated server in the Servers workspace. After running a Snapshot Job, the results will be provided in a human readable format which is suitable for an administrator to review. The results display in a hierarchical tree beneath a job in the Jobs workspace and under the Snapshot Results node for a server in the Servers workspace. Results show each run of the job, date and time of the run, and the servers where the job ran.

Using snapshot results, an administrator can:
Base an audit on a snapshot

Bundle snapshot results into a BLPackage or software package
Export the results to a snapshot report

When a snapshot is taken of most types of server objects, only attribute information is saved, such as the name of a patch, its version, and the date of installation. Attribute information is always saved in the database. Snapshots of file systems, however, can contain actual content. Content is saved in the file server. Snapshot results can be exported into the following formats**:**

HTML format, which is suitable for printing or viewing with a web browser
Comma-separated value (CSV) format, which can be imported into databases or spreadsheets.

6.1.6.11 Audit Report

Audit results compare the master (which can be a server, snapshot, or component) to other hosts, including hosts in other snapshots. Audit Reports can be viewed in two ways; server or object. An Audit Report would be viewed using an object to determine which servers have a configuration that is inconsistent with server objects on the master server. An Audit Report would be viewed using a server to determine if a server object is consistent with a corresponding server object on the master server. Audit reports are created by running an Audit Job.

6.1.7 Identification and Authentication

BladeLogic employs a two-stage procedure for authenticating client application users to their respective middle-tier servers. First, client users authenticate with a BladeLogic Authentication Service, a service hosted by a BladeLogic Application Server, and acquire a BladeLogic single sign-on (SSO) session credential. Then, having acquired a credential, the client application establishes a TLS session with a middle tier service—either an Application Service or Network Shell Proxy Service. Once the TLS session is established, the client presents its SSO session credential to the service, which validates the credential and uses it to establish the identity of the client user. Readers familiar with HTTP cookies may view BladeLogic SSO session credentials as analogous to cookies used to communicate an authenticated identity to a BladeLogic service. The following figure depicts this two-stage authentication procedure.

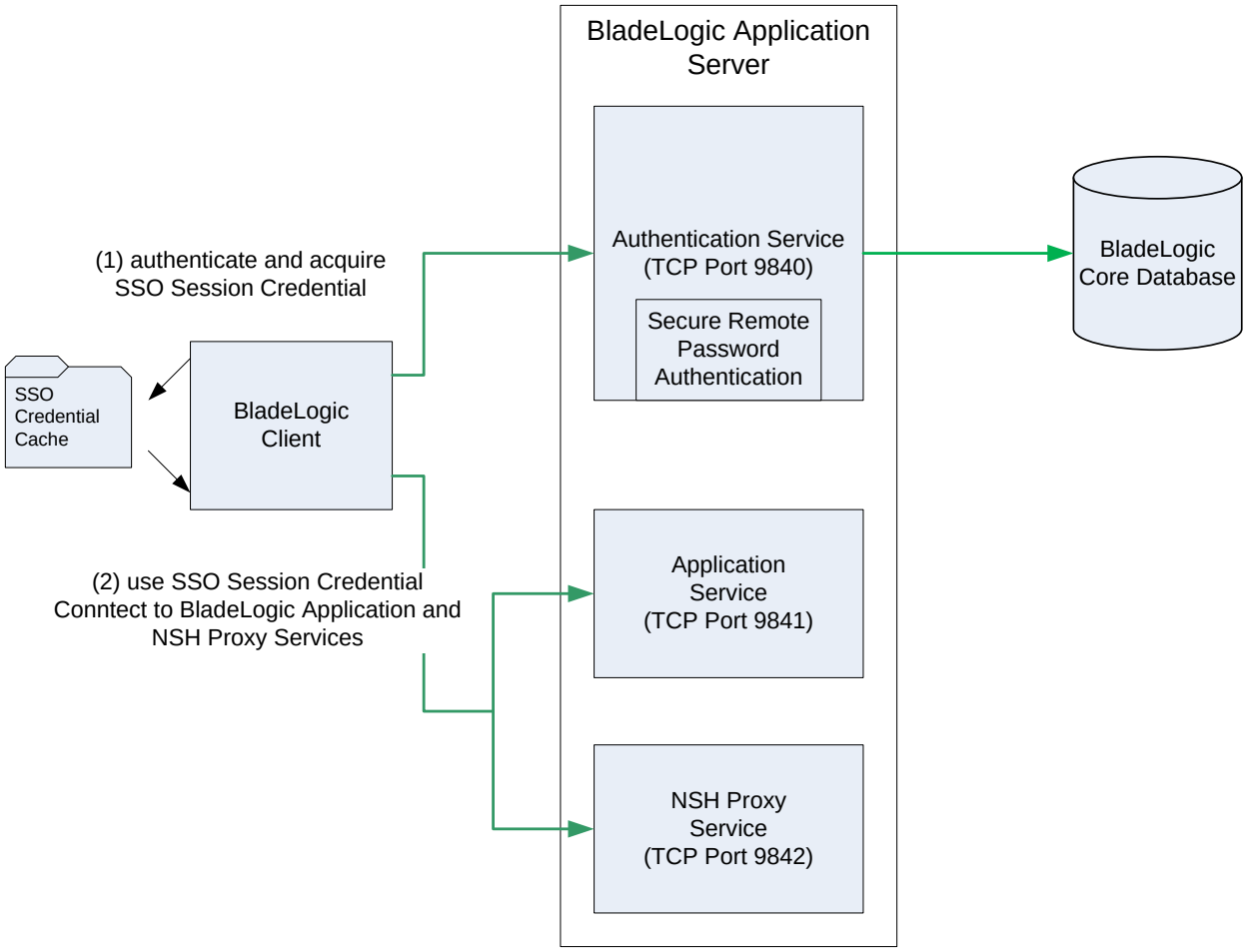

Figure 6-1: BladeLogic Authentication with SSO

SSO session credentials have a finite lifetime and can be cached in the file system of the client's host. The SSO credential lifetime is set at ten hours by default; however, this is a configurable parameter. If a client application's credential cache contains an unexpired session credential, that credential can be used to establish a new client/server session without requiring the user to re-authenticate. All BladeLogic client applications can share the same session credential.

The BladeLogic Authentication Service supports the authentication of end users registered in an external enterprise-level identity management system (Active Directory / Kerberos) as well as users registered within Operations Manager's own application-level user directory (SRP – Secure Remote Password).

Operation Manager's internally registered users authenticate via username and password, using the Secure Remote Password protocol (IETF RFC 2945).

Users can authenticate and obtain SSO session through authentication clients built into the configuration manager console GUI, or a command line utility called ***blcred***. However, the blcred command line utility is not in the evaluated configuration. This is because users are able to obtain an SSO credential by authenticating via the Configuration Manager. Once the SSO credential is received, the user is able to execute tasks via the BLCLI or the

Configuration Manager. When authenticating via the Configuration Manager, users are prompted for their identity and authentication credentials. In this TOE, the credentials requested are a user's SRP username and password.

BladeLogic Reports does not employ SSO credentials. Users present their I&A credentials to the Reports Server across a web interface. The client/webserver communications run over https, guarding against an eavesdropper acquiring the user's I&A credentials.

The default BladeLogic installation sets up security in the following way:

- Users of Operations Manager must authenticate by providing a username and password via Secure Remote Password (SRP).
- Network Shell is launched through the configuration manager or the administrator's workstation and then interacts directly with RSCD Agents, after being authenticated by the Application Manager. In the evaluated configuration, Network Shell is launched through an administrator workstation and interacts directly with the RSCD Agent.
- Authorization system-wide is based on RBAC and agent ACLs

In addition, BladeLogic uses the following algorithm to determine whether a user has permissions for accessing an RSCD Agent:

- First, the client reads its secure file to determine whether it includes an entry for a particular server. If an entry for that server exists, the client uses the information in the entry to establish a connection with the server.
- Then, the system checks the exports configuration file, where users connecting from specified machines can be mapped to a particular user, such as Administrator on Windows.
- The system then checks the local user and the users.local configuration files to determine if these files include any entries that supersede the definitions set in the exports file. Permissions are granted on a user-by-user basis.

6.1.7.1 Password Policy

The RBACAdmins' configure the password policies through the Configuration Manager depending on the organizations need for a low, medium, or high security policy.

- **Password minimum length –** by default or by entering "0" there is no minimum length for passwords.
- **Maximum password age –** by setting a maximum password age, users will be required to change their passwords at specified intervals. By entering "0" the passwords will never expire.
- **Account lockout –** by setting a threshold and duration for account lockouts, it is specified how many failed logins cause a user to be locked out and how long that lockout lasts. By configuring "0" for lockout on failed logins, the users can't be locked out because of login failures. By configuring "0" under length of lockout will cause the user to require the RBACAdmins to unlock the account

- The password policy for the evaluated configuration of the TOE consists of 8 character minimum passwords, a 15 minute delay after 3 failed login attempts, and a 90 day expiration period for passwords.

6.2 TOE Security Assurance Measures

This section identifies the assurance measures provided by the developer in order to meet the security assurance requirement components for EAL3 augmented with ALC_FLR.1. These measures are identified in the following table.

Component	Document(s)	Rationale
ACM_CAP.3: Authorisation controls	[1] Configuration Management Plan for Operations Manager v7.4.2 [2] Configuration Management Evidence of Coverage for Operations Manager v7.4.2 [3] 7_4_2_list.txt [4] configuration_item_list.doc [5] (folder) func_spec_dump	[1] This document describes the configuration management method for the TOE. [2] This document shows system records and other assurance that the technologies and procedures in the CM Plan are being used. [3] This document lists the configuration items for the TOE's implementation representation (source code). [4] This document lists the configuration items used as evidence for the evaluation of the SARs defined in the ST. [5] The contents of this folder are a series of images which list the configuration items for the TOE's document repository. These documents represent the CI List, CM Plan, and evidence that the CM Plan is applied to the items in the CI List.
ACM_SCP.1: TOE CM coverage	[1] 7_4_2_list.txt [2] configuration_item_list.doc [3] (folder) func_spec_dump	[1] This document lists the configuration items for the TOE's implementation representation (source code). [2] This document lists the configuration items used as evidence for the evaluation of the SARs defined in the ST. [3] The contents of this folder are a series of images which list

Component	Document(s)	Rationale
		the configuration items for the TOE's document repository. Together, these three items comprise the CI List for the TOE and determine the scope of items maintained by the CM system.
ADO_DEL.1: Delivery procedures	[1] BMC/BladeLogic Electronic Product Distribution (EPD)	This document describes product delivery for BladeLogic and a description of all procedures used to ensure objectives are not compromised in the delivery process.
ADO_IGS.1: Installation, generation, and start-up procedures	[1] BladeLogic Administration Guide v7.4.2 [2] BladeLogic Users Guide v 7.4.2 [3] BladeLogic Installation Guide v7.4.2 [4] Using BladeLogic Reports [5] BladeLogic Operations Manager Install Guide.doc	These documents together document the procedures necessary and describe the steps required for the secure installation, generation, and start-up of the TOE.
ADV_FSP.1: Informal functional specification	[1] Functional Specification Document for BladeLogic v7.4.2 (Document version 1.2)	This document describes the TOE security functions (TSF) from the viewpoint of the behaviour of the external interfaces.
ADV_HLD.2: Security enforcing high-level design	[1] Functional Specification Document for BladeLogic v7.4.2 (Document version 1.2) [2] High Level Design Document for BladeLogic Operations Manager v7.4.2 (Document version 1.2)	These documents together describe the TSF structure and interfaces in terms of subsystems.
ADV_RCR.1: Informal correspondence demonstration	[1] Functional Specification Document for BladeLogic v7.4.2 (Document version 1.2) [2] High Level Design Document for BladeLogic Operations Manager v7.4.2 (Document version 1.2)	These documents provide an analysis on relations between security functions in the summary specification and the subsystem in the functional specification/high-level design for the ST. They show the correspondence of the representation of the TOE between SFRs, external interfaces, and subsystems/internal interfaces.

Component	Document(s)	Rationale
AGD_ADM.1: Administrator guidance	[1] BladeLogic Administration Guide v7.4.2 [2] BladeLogic Users Guide v 7.4.2 [3] BladeLogic Installation Guide v7.4.2 [4] BladeLogic Network Shell Command Reference v 7.4.2 [5] Using BladeLogic Reports [6] Booz Allen_BMC_v7+4+2_1-3_AdminGuideSupp_3_20091106.doc	These documents together describe the processes to be used for proper administration of the TOE.
AGD_USR.1: User guidance	[1] Using BladeLogic Reports [2] BladeLogic Users Guide v 7.4.2	These documents together describe the proper use of the TOE from a user standpoint.
ALC_DVS.1: Identification of security measures	[1] (folder) acquisition [2] AssuranceLifecycle-dff-20090615.doc [3] audit_practices.pdf [4] Booz Allen_BMC_BLDE_ALC_DVS_onsite report_20090722.doc [5] Configuration_Management_Plan-v7_4_2.doc [6] Document Confidentiality and Intellectual Property Assigment Agreement.pdf [7] Floor Plans with Access and CCTV details.pdf [8] India RM Retention Schedule.doc [9] Infrastructure Backups.pdf [10] LexingtonAccessRecord.pdf [11] LexingtonFloorPlan.pdf [12] policy_lost_laptop.doc [13] Pre-Employment Screening Policy for BMC Employees.pdf [14] proprietary information handling.pdf [15] rules of the road.pdf [16] Sec scan.jpg [17] shredding evidence.pdf [18] unit test process.pdf	[1] The contents of this folder are a number of documents which describe the procedures used by BMC (the current owner of the TOE) to vet BladeLogic (the developer of the TOE), who was acquired by BMC in 2009. [2] This document is a high-level description of the physical, procedural, and personnel security of the TOE development. [3] This document is a summary of BMC's self-auditing policies. [4] This document is a report summarizing the development security processes, including interviews with key personnel who support the development and infrastructure of the TOE to confirm the accuracy of the documentation. [5] The CM Plan defines some aspects of development security including project lifecycle and CM system records (auditing). [6] This document is what new employees of BMC must sign when joining the company. It defines proprietary information

Component	Document(s)	Rationale
	[19] US RM Retention Schedule.doc	and makes the signer aware of potential disciplinary action for divulging this information. [7] This document provides floor plans of one of the two facilities used to develop, test, and maintain the TOE, including access controlled entry points and the location of surveillance cameras. [8] This document provides a definition of how long different types of records are retained at the India facility where TOE work takes place. [9] This document defines the backup schedule for all of the master components of the TOE's implementation representation. [10] This document provides an example of the ability for physical security to audit badge access for personnel accessing a development site. [11] This document provides floor plans of one of the two facilities used to develop, test, and maintain the TOE, including access controlled entry points and the location of surveillance cameras. [12] This document describes security best practices for employees to safeguard their laptops and remediation procedures to perform if it is lost or stolen. [13] This document describes the background check policies that candidate employees must undergo before being considered eligible to be hired. [14] This document describes the policies which should be used to identify and protect proprietary information. [15] This document describes

Component	Document(s)	Rationale
		the acceptance criteria for QA to determine when a revision of the TOE is sufficiently developed to begin testing. [16] This document is a scan of the visitor log for a development facility which shows that the defined visitor policy is being followed. [17] This document is a bill to commercial document destruction services showing that proprietary information is disposed of in a secure manner. [18] This document details the testing process of revisions of the TOE, providing assurance that most potential defects should be addressed before release. [19] This document provides a definition of how long different types of records are retained at the US facility where TOE work takes place. These documents provide information about the organizational policies and procedures that apply to the secure development of the TOE and security of the infrastructure used to develop it.
ALC_FLR.1: Basic flaw remediation	[1] FlawRemediation.doc	This document provides a description of procedures used to track all reported security flaws in each release of the TOE.
ATE_COV.2: Analysis of coverage	[1] BladeLogic Operations manager 7.4.2 Functional Test Plan (Document version 1.2) [2] Test Evidence of Coverage.xlsx	These documents provide documented evidence that all product security functions have been tested against functional specification of security elements.
ATE_DPT.1 Testing: High-level design	[1] BladeLogic Operations manager 7.4.2 Functional Test Plan (Document version 1.2)	These documents provide evidence that the tests identified in the test documentation

Component	Document(s)	Rationale
	[2] Test Evidence of Coverage.xlsx	(ATE_FUN.1) are sufficient to demonstrate that the TSF operates in accordance with its high-level design and that all tests were conducted.
ATE_FUN.1: Functional testing	[1] BladeLogic Operations manager 7.4.2 Functional Test Plan (Document version 1.2) [2] Test Evidence of Coverage.xlsx [3]Booz_Allen_BMC_Blade7+4+2_ TSS-TestCase_Matrix_20090717.xls [4]BoozAllen_BMC_v7.4.2_ASE_3_ 20090727_TestCase-TSS_highlighted.doc [5]Server Authorizations.docx	These documents together provide an overview of the tests performed by the developer against the claims contained in the Security Target, describe the security functional test items and specific test procedures, inputs and outputs used for proving the TSF functions are as specified, the expected test results, the actual test results while executing each test case, and the evidence these tests were conducted.
ATE_IND.2: Independent testing - sample	[1] Evaluation Team Test Plan for BladeLogic Operations Manager Version 7.4.2 (Document version 1.1) [2] Evaluation Team Test Report for BladeLogic Operations Manager Version 7.4.2 (Document Version 1.1)	These documents provide introduction of tests that were performed by the evaluators in order to verify the accuracy of the developer testing.
AVA_MSU.1: Examination of guidance	[1] BladeLogic Administration Guide v7.4.2 [2] BladeLogic Users Guide v 7.4.2 [3] BladeLogic Installation Guide v7.4.2 [4] BladeLogic Network Shell Command Reference v 7.4.2 [5] Using BladeLogic Reports [6] Booz Allen_BMC_v7+4+2_1-3_AdminGuideSupp_3_20091106.doc	These documents describe the procedures to help the TOE user's security install the product and software and perform operations.
AVA_SOF.1: Strength of TOE security function evaluation	[1] SOF Rationale.doc	This document describes the analysis result of strength of function for security mechanisms which have probabilistic or permutational mechanisms excluding cryptographic mechanism for

Component	Document(s)	Rationale
		the TOE.
AVA_VLA.1: Developer vulnerability analysis	[1] BladeLogic Operations manager 7.4.2 Functional Test Plan (Document version 1.2) [2] Vulnerability Analysis BladeLogic Operations Manager v7.4.2 (Document version 0.6) [3] Evaluation Team Test Plan for BladeLogic Operations Manager Version 7.4.2 (Document version 1.1) [4] Evaluation Team Test Report for BladeLogic Operations Manager Version 7.4.2 (Document Version 1.1)	These documents together describes the activities performed by the developer against the TOE and how they will be mitigated to ensure that obvious security vulnerabilities found will not be wrongfully used in the TOE environments.

Table 6 – TOE Security Assurance Measures

7 Protection Profile Claims

There are no PP claims for this evaluation.

8 Rationale

8.1 Security Objectives Rationale

The following table provides a mapping with rationale to identify the security objectives that address the stated assumptions and threats.

Assumption	Objective	Rationale
A.ADMIN There will be one or more authorized administrators assigned to install, configure, and manage the TOE and the security information it contains..	OE.ADMIN One or more authorized administrators will be assigned to install, configure and manage the TOE and the security of the information it contains.	OE.ADMIN maps to A.ADMIN in order to ensure that authorised administrators install, manage and operate the TOE in a manner that maintains its security objectives.
A.PATCHES System Administrators exercise due diligence to update the TOE with the latest patches and patch the IT Environment (e.g., OS and database) so they are not susceptible to network attacks.	OE.ADMIN One or more authorized administrators will be assigned to install, configure and manage the TOE and the security of the information it contains.	OE.ADMIN maps to A.PATCHES in order to ensure that the authorized administrators properly patch the IT environment in a manner that maintains its security objectives. Furthermore, this objective ensures the authorized administrator will download the latest service packs and hotfixes to the application server of the TOE and update the associated XML file as needed.
A.NOEVIL All users of the TOE are not careless, willfully negligent, or hostile, and will follow and abide by the instructions provided by the TOE documentation.	OE.NOEVIL All users of the TOE are not careless, willfully negligent, or hostile and will follow and abide by the instructions provided by the guidance documentation.	OE.NOEVIL directly maps to A.NOEVIL and ensures that all users of the TOE are properly trained in the configuration and usage of the TOE and will follow the guidance provided.
A.LOCATE The processing resources of the TOE will be located within controlled access facilities, which will prevent unauthorized physical access.	OE.LOCATE The TOE will be located within controlled access facilities that will prevent unauthorized physical access.	OE.LOCATE directly maps to A.LOCATE to ensure that those responsible for the TOE locate the TOE in a controlled access facility that will prevent unauthorized physical access.

Table 7 – Assumption to Objective Mapping

Threat	Objective	Rationale
T.ACCESS Authorized users could gain electronic access to protected network resources by attempting to establish a connection that they are not permitted to perform.	O.ACCESS The TOE will provide measures to authorize users to access specified TOE resources once the user has been authenticated. User authorization is based on access rights configured by the authorized users of the TOE.	O.ACCESS (FDP_ACC.1, FDP_ACF.1) addresses T.ACCESS by providing the authorized users with the capability to specify access restrictions on the protected TOE resources to authenticated users.
	OE. FILESYS The Security features offered by the underlying Operating	OE.FILESYS (FAU_STG.1) addresses T.ACCESS by ensuring that

Threat	Objective	Rationale
	system protect the files used by the TOE.	the underlying Operating System provides the capability to store and protect the files used by the TOE.
T.ADMIN_ERROR An administrator may incorrectly install or configure the TOE, or install a corrupted TOE resulting in ineffective security mechanisms.	O.ROBUST_ADMIN_GUIDANCE The TOE will provide administrators with the necessary information for secure delivery and management.	O.ROBUST_ADMIN_GUIDANCE (ADO_DEL.1, ADO_IGS.1, AGD_ADM.1, AGD_USR.1, AVA_MSU.1) helps to mitigate this threat by ensuring the TOE administrators have guidance that instructs them how to administer the TOE in a secure manner and to provide the administrator with instructions to ensure the TOE was not corrupted during the delivery process. Having this guidance helps to reduce the mistakes that an administrator might make that could cause the TOE to be configured in a way that is unsecure.
	O.MANAGE The TOE will provide authorized users with the resources to manage and monitor user accounts, TOE resources and security information relative to the TOE.	O.MANAGE (FMT_MOF.1, FMT_MSA.1, FMT_MSA.2, FMT_MSA.3, FMT_MTD.1(1), FMT_MTD.1 (2), FMT_SMF.1, FMT_SMR.1) addresses T.ADMIN_ERROR by ensuring only authorized administrators can use the provided resources for managing and monitoring user accounts, TOE resources and security information relative to the TOE.
T.MASK Users whether they be malicious or non-malicious, could gain unauthorised access to the TOE by bypassing identification and authentication countermeasures.	O.AUDIT The TOE will provide measures for recording security relevant events that will assist the authorized users in detecting misuse of the TOE and/or its security features that would compromise the integrity of the TOE and violate the security objectives of the TOE.	O.AUDIT (FAU_GEN.1, FAU_GEN.2, FAU_SAR.1, FAU_SAR.2) addresses T.MASK by providing the authorized users with tools necessary to monitor user activity to ensure that misuse of the TOE does not occur.
	OE.TIMESTAMP The runtime environment for the audit mechanism of the TOE must provide a reliable time source for audit record generation.	OE.TIMESTAMP (FPT_STM.1) addresses this T.MASK by providing an audit mechanism in the underlying Operating System that includes the current date and time in each audit record.

Threat	Objective	Rationale
T.MODIFY Users could attempt to misconfigure or modify their user accounts in an attempt to tamper with TOE resources or modify security information relative to the TOE.	O.MANAGE The TOE will provide authorized users with the resources to manage and monitor user accounts, TOE resources and security information relative to the TOE.	O.MANAGE (FMT_MOF.1, FMT_MSA.1, FMT_MSA.2, FMT_MSA.3, FMT_MTD.1 (1), FMT_MTD.1 (2), FMT_SMF.1, FMT_SMR.1) addresses T.MODIFY by ensuring that only authorized users can use the provided resources for managing and monitoring user accounts, TOE resources and security information relative to the TOE.
T.EAVESDROPPING A malicious user could eavesdrop on network traffic to gain unauthorized access to TOE data.	O.EAVESDROPPING The TOE will provide measures to assist the authorized users in detecting unauthorized monitoring of networks or information systems that would compromise the integrity of the TOE and violate the security objectives of the TOE.	O. EAVESDROPPING (FCS_CKM.1(1), FCS_CKM.1(2), FCS_CKM.4, FCS_COP.1, FPT_TRP.1, FPT_ITT.1) mitigates T.EAVESDROPPING by ensuring that all communication to and from the TOE or between components of the TOE are not sent unless they are encrypted.
T.MISCONFIG Users, whether they be malicious or non-malicious, could attempt to modify the configuration of remote servers on a local network in an attempt to reduce the security posture of those remote servers.	O.MONITOR The TOE will monitor remote server configurations to ensure the servers are configured according to the local security policy. The TOE will collect and analyze critical configuration data of remote servers in the IT environment.	O.MONITOR (FAU_GEN_EXP.1(1), FAU_GEN_EXP.1(2), FAU_GEN_EXP.1(3), FAU_SAR_EXP.1(1), FAU_SAR_EXP.1(2), FAU_SAR_EXP.1(3), FAU_STG.1) mitigates this threat by having the TOE remotely monitor the configuration settings of remote servers on the local network.
	O.COMPLIANCE The TOE will maintain remote server configuration consistent with local security policy including files, registry, and patches.	O.COMPLIANCE (FCM_JOB_EXP.1) mitigates this threat by having the TOE remotely patch and change configuration settings of remote servers on the local network.
T.PROTECT A user or process may cause, through an unsophisticated attack, TSF data, or executable code to be inappropriately accessed (viewed, modified, or deleted).	OE.PROTECT The operating environment will provide a domain for its own execution that protects itself, its resources, and the TOE from external interference, tampering, or unauthorized disclosure.	OE.PROTECT (FPT_RVM_EXP.1, FPT_SEP_EXP.1) maps to T.PROTECT to ensure that the environment is configured in a manner that protects the TOE components and operation.
	O.SELF_PROTECTION The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure. The TSF will provide a High Availability configuration which allows for continued operation of the TOE in the event of a single unit failure.	O.SELF_PROTECTION (FPT_RVM_EXP_TOE.1, FPT_SEP_EXP_TOE.1) contributes to countering this threat by ensuring that the TSF can protect itself from users. If the TSF could not maintain and control its domain of execution, it could not be trusted to control access to the resources under its control.

Threat	Objective	Rationale
T.UNAUTH Users could gain unauthorized access to TOE resources by bypassing identification and authentication requirements.	**O.AUTH** The TOE will provide measures to uniquely identify all users and will authenticate the claimed identity prior to granting a user access to the TOE.	O.AUTH (FIA_UAU.2, FIA_UID.2, FIA_SOS.1, FIA_UAU.7, FIA_AFL.1, FIA_ATD.1 addresses T.UNAUTH by providing measures to uniquely identify and authenticate users through SRP username/password and SSO credential.

Table 8 – Threat to Objective Mapping

8.2 Security Functional Requirements Rationale

The following table provides a mapping with rationale to identify the security functional requirement components that address the stated TOE and environment objectives.

Objective	Security Functional Components	Rationale
O.ACCESS The TOE will provide measures to authorize users to access specified TOE resources once the user has be authenticated. User authorization is based on access rights configured by the authorized users of the TOE.	FDP_ACC.1 Subset access control	FDP_ACC.1 requires the TOE to prevent unauthorized access to TOE resources by enforcing the BladeLogic Access Control Policy.
	FDP_ACF.1 Security attribute based access control	FDP_ACF.1 requires the TOE to enforce the BladeLogic Access Control Policy on the protected TOE resources and requires the authorized administrators to configure user access rights accordingly.
O.AUDIT The TOE will provide measures for recording security relevant events that will assist the authorized users in detecting misuse of the TOE and/or its security features that would compromise the integrity of the TOE and violate the security objectives of the TOE.	FAU_GEN.1 Audit data generation	FAU_GEN.1 defines the security relevant events that will be recorded by the TOE along with the details of the event that will be recorded.
	FAU_GEN.2 User identity association	FAU_GEN.2 requires the TOE to associate each auditable event with the identity of the user that caused the event.
	FAU_SAR.1 Audit Review	FAU_SAR.1 provides the capability to read information from the audit records.
	FAU_SAR.2 Restricted audit review	FAU_SAR.2 requires that there are no other users except those that have been identified in FAU_SAR.1 Audit review that can read the information.

Objective	Security Functional Components	Rationale
O.MONITOR The TOE will monitor remote server configurations to ensure the servers are protected according to the local security policy. The TOE will collect and analyze critical data to report on the performance, capacity, availability, and response of the protected servers, and applications.	FAU_GEN_EXP.1(1) Audit data generation	FAU_GEN_EXP.1(1) defines the server objects that will be recorded by the TOE along with the details of the event that will be recorded for each snapshot job.
	FAU_GEN_EXP.1(2) Audit data generation	FAU_GEN_EXP.1(2) defines the items listed in the baseline/server configuration file that will be recorded by the TOE along with the details of the event that will be recorded for each audit job.
	FAU_GEN_EXP.1(3) Audit data generation	FAU_GEN_EXP.1(3) defines the server objects that will be recorded by the TOE along with the details of the event that will be recorded for each patch analysis job.
	FAU_SAR_EXP.1(1) Audit review	FAU_SAR_EXP.1(1) provides the capability to read information from the snapshot jobs.
	FAU_SAR_EXP.1(2) Audit review	FAU_SAR_EXP.1(2) provides the capability to read information from the audit jobs.
	FAU_SAR_EXP.1(3) Audit review	FAU_SAR_EXP.1(3) provides the capability to read information from the patch analysis jobs.
	FAU_STG.1 Audit storage	FAU_STG.1 provides protection of the various job report output (i.e., remote server configuration settings).
O.AUTH The TOE will provide measures to uniquely identify all users and will authenticate the claimed identity prior to granting a user access to the TOE.	FIA_ATD.1 User attribute definition	FIA_ATD.1 specifies the security attributes that should be maintained at the level of the user. This means that the security attributes listed are assigned to and can be changed at the level of the user. In other words, changing a security attribute in this list associated with a user should have no impact on the security attributes of any other user.
	FIA_UAU.2 User authentication before any action	FIA_UAU.2 requires a user be authenticated before any access to the TOE and resources protected by the TOE is allowed.
	FIA_UAU.7 Protected authentication feedback	FIA_UAU.7 requires that only limited feedback information is provided to the user during the authentication.

Objective	Security Functional Components	Rationale
	FIA_UID.2 User identification before any action	FIA_UID.2 requires a user be identified before any access to the TOE and resources protected by the TOE is allowed.
	FIA_AFL.1 Authentication Failure Handling	FIA_AFL.1 ensures unauthorized users cannot endlessly attempt to authenticate after some number of failed attempts defined by an authorized administrator the user becomes locked out.
	FIA_SOS.1 Verification of Secret	FIA_SOS.1 The strength of the authentication mechanism is sufficient to ensure that unauthorized users cannot easily impersonate an authorized user.
O.MANAGE The TOE will provide authorized administrators with the resources to manage and monitor user accounts, resources, and security information relative to the TOE.	FMT_MOF.1 Management of security functions behaviour	FMT_MOF.1 allows the authorized users (roles) to manage the behavior of functions in the TSF that use rules or have specified conditions that may be manageable.
	FMT_MSA.1 Management of security attributes	FMT_MSA.1 restricts the assignment of security attributes of users and resources to the authorized administrators.
	FMT_MSA.2 Secure security attributes	FMT_MSA.2 ensures only secure values are accepted for security attributes.
	FMT_MSA.3 Static attribute initialization	FMT_MSA.3 allows the authorized administrators to override the default values set for security attributes when creating user accounts.
	FMT_MTD.1(1) Management of TSF data	FMT_MTD.1 (1) allows authorized users (roles) control over the management of TSF data. Examples of TSF data include audit information, clock, system configuration and other TSF configuration parameters.
	FMT_MTD.1(2) Management of TSF data	FMT_MTD.1 (2) restricts the ability to [**perform functions specified in table 3]** the [TSF data specified in table 3] to [default roles specified in table 3, and other roles as specified].
	FMT_SMF.1 Specification of management functions	FMT_SMF.1 requires that the TOE provide the ability to manage its security functions including the management of user accounts and user access rights, TOE resources

Objective	Security Functional Components	Rationale
		and security information recorded in the audit logs.
	FMT_SMR.1 Security Roles	FMT_SMR.1 requires the TOE to provide the ability to set roles for security relevant authority as well as to restrict the ability to define and assign roles to authorized administrators.
O.ROBUST_ADMIN_GUIDANCE The TOE will provide administrators with the necessary information for secure delivery and management.	ADO_DEL.1 Delivery procedures	ADO_DEL.1 describes product delivery and a description of all procedures used to ensure objectives are not compromised in the delivery process.
	ADO_IGS.1 Installation, generation, and startup procedures	ADO_IGS.1 documents the procedures necessary and describe the steps required for the secure installation, generation, and start-up of the TOE.
	AGD_ADM.1 Administrator guidance	AGD_ADM.1 describes the processes to be used for proper administration of the TOE.
	AGD_USR.1 User guidance	AGD_USR.1 describes the proper use of the TOE from a user standpoint.
	AVA_MSU.1 Examination of guidance	AVA_MSU.1 describes the procedures to help the TOE user's security install the product and software and perform operations.
O.EAVESDROPPING The TOE will provide measures to assist the authorized users in detecting unauthorized monitoring of networks or information systems that would compromise the integrity of the TOE and violate the security objectives of the TOE.	FCS_CKM.1(1) Cryptographic key generation	FCS_CKM.1(1) requires key generation for RSA in support of TLS sessions of the TOE.
	FCS_CKM.1(2) Cryptographic key generation	FCS_CKM.1(2) requires key generation with SRP protocol in support of TLS sessions of the TOE.
	FCS_CKM.4 Cryptographic key destruction	FCS_CKM.4 requires destruction of encryption keys used to manage remote sessions of the TOE.
	FCS_COP.1 Cryptographic operation	FCS_COP.1 requires encryption and decryption with AES in support of TLS sessions of theTOE.
	FTP_TRP.1 Trusted Path	FPT_TRP.1 ensures communications between the TOE and the user are encrypted.
	FPT_ITT.1 Basic Internal TSF Data Transfer Protection	FPT_ITT.1 Ensures communications between components of the TOE are encrypted.

Objective	Security Functional Components	Rationale
OE.FILESYS The security features offered by the underlying Operating System and Database protect the files used by the TOE. .	FAU_STG.1 Protected audit trail storage	FAU_STG.1 requires that the underlying SQL Server protect the audit records generated by the TOE that are stored in the BladeLogic Core Database and Reporting Data Warehouse.
OE.TIMESTAMP The runtime environment for the audit mechanism of the TOE must provide a reliable time source for audit record generation.	FPT_STM.1 Reliable Time Stamps	FPT_STM.1 requires that an accurate time source will be available to the TOE for use in applying timestamps in the audit trail.
OE.PROTECT The IT Environment will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure.	FPT_RVM_EXP.1 Non-bypassability of the TSP	FPT_RVM_EXP.1 requires that the operating environment provide mechanisms to prevent bypassing the security features provided and enforced by the TOE.
	FPT_SEP_EXP.1 TSF domain separation	FPT_SEP_EXP.1 requires that the operating environment provide an isolated domain for the TOE to operate.
O.SELF_PROTECTION The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure.	FPT_RVM_EXP_TOE.1 Non-bypassability of the TSP	FPT_RVM_EXP_TOE.1 requires that the operating environment provide mechanisms to prevent bypassing the security features provided and enforced by the TOE.
	FPT_SEP_EXP_TOE.1 TSF domain separation	FPT_SEP_EXP_TOE.1 requires that the operating environment provide an isolated domain for the TOE to operate.
O.COMPLIANCE The TOE will maintain remote server configuration consistent with local security policy including files, registry, and patches.	FCM_JOB_EXP.1 (1) Compliance management Jobs	FCM_JOB_EXP.1(1), requires the TOE to have the ability to forcibly correct any deviations between an audit report and a snapshot report or server configuration file. This maintains compliance with local security policies.
	FCM_JOB_EXP.1 (2) Compliance management Jobs	FCM_JOB_EXP.1 (2) requires the TOE to have the ability to deploy (or push) multiple files and directories to one or more managed servers. This maintains compliance with local security policies.

Objective	Security Functional Components	Rationale
	FCM_JOB_EXP.1 (3) Compliance management Jobs	FCM_JOB_EXP.1 (3) requires the TOE to have the ability to execute deployed (or pushed) content to one or more managed servers unattended. This maintains compliance with local security policies.
	FCM_JOB_EXP.1 (4) Compliance management Jobs	FCM_JOB_EXP.1 (4) requires the TOE to have the ability to allow for the deployment and execution of previously saved network shell scripts. This maintains compliance with local security policies.
	FCM_JOB_EXP.1 (5) Compliance management Jobs	FCM_JOB_EXP.1 (5) requires the TOE to have the ability to concatenate a series of deploy jobs, file deploy jobs, and network shell jobs, as well as deploys a series of BLPackages. This maintains compliance with local security policies.

Table 9 – Security Functional Requirements Rationale

8.3 Security Assurance Requirements Rationale

The rationale for the Security Assurance Requirements is discussed in Table 6 – TOE Security Assurance Measures of Section 6.2.

8.4 Requirement Dependency Rationale

All Security Functional Requirement component dependencies have been met by the TOE with the exception of FPT_STM.1, and ADV_SPM.1. FPT_STM.1, Reliable Time Stamps is met by the IT environment and is a dependency on FAU_GEN.1, FAU_GEN_EXP.1(1), FAU_GEN_EXP.1(2), and FAU_GEN_EXP.1(3). An accurate time source will be available to the TOE for use in determining the timestamp for the audit trail. ADV_SPM.1 is a dependency of FMT_MSA.2. FMT_MSA.2 is included to satisfy the dependency for FCS requirements. ADV_SPM.1 is not applicable with regards to supporting cryptographic functionality and therefore is not included in this ST.

8.5 Explicitly Stated Requirements Rationale

This TOE contains the following explicit security functions:

- FAU_GEN_EXP.1(1)
- FAU_GEN_EXP.1(2)
- FAU_GEN_EXP.1(3)
- FAU_SAR.EXP.1(1)
- FAU_SAR.EXP.1(2)
- FAU_SAR.EXP.1(3)
- FCM_JOB_EXP.1.1(1)
- FCM_JOB_EXP.1.1(2)

- FCM_JOB_EXP.1.1(3)
- FCM_JOB_EXP.1.1(4)
- FCM_JOB_EXP.1.1(5)
- FPT_RVM_EXP_TOE.1
- FPT_SEP_EXP_TOE.1
- FPT_RVM_EXP.1
- FPT_SEP_EXP.1

FAU_GEN_EXP.1(1), FAU_GEN_EXP.1(2), FAU_GEN_EXP.1(3), FCM_JOB_EXP.1.1(1), FCM_JOB_EXP.1.1(2), FCM_JOB_EXP.1.1(3), FCM_JOB_EXP.1.1 (4), FCM_JOB_EXP.1.1 (5) were created to capture the basic functionality provide by the TOE. FAU_GEN_EXP.1(1) allows for TOE to perform snapshot jobs and record the configuration of a group of severs objects at a point in time. FAU_GEN_EXP.1(2) allows for the TOE to perform audit jobs which can be used to determine whether server configurations match a standard configuration (i.e. select components, snapshots, or live server objects) FAU_GEN_EXP.1(3) allows for the TOE to perform patch analysis jobs which can be used to compare the patch configuration on servers. FCM_JOB_EXP.1.1(2) allows the TOE to perform File Deploy Jobs that allows the TOE to push multiple files and directories to one or more managed servers. FCM_JOB_EXP.1.1(3) allows the TOE to perform Deploy Jobs to deploy software packages to one or more remote servers. FCM_JOB_EXP.1.1(1) allows the TOE to correct deficiencies by comparing an audit report and a snapshot report. FCM_JOB_EXP.1.1 (4) allows the TOE to have the ability to allow for the deployment and execution of previously saved network shell scripts. FCM_JOB_EXP.1.1 (5) allows the TOE to have the ability to concatenate a series of deploy jobs, file deploy jobs, and network shell jobs, as well as deploy a series of BLPackages.

FAU_SAR.EXP.1(1), FAU_SAR.EXP.1(2), and FAU_SAR.EXP.1(3) were created to provide additional capabilities of the TOE to read information collected in FAU_GEN_EXP.1(1), FAU_GEN_EXP.1(2), and FAU_GEN_EXP.1(3).

FPT_RVM_EXP_TOE.1, FPT_SEP_EXP_TOE.1, FPT_RVM_EXP.1, and FPT_SEP_EXP_.1 had to be explicitly stated because the TOE is software only and therefore can only provide partial TOE self-protection while relying on the OS and Hardware platforms to provide the full protection. The approach used for these requirements is according to the NIAP policy requiring software-only TOEs to use explicit requirements to specify the aspects provided by the TOE and those provided by the platform. The current reference for this policy is documents: 'TOE Protection, March 12, 2005', and 'CCEVS Policy on Accepting Security Target, April 8, 2005'. As with FPT_RVM.1, and FPT_SEP.1, on which they were based, these explicit requirements have no dependencies.

This Security Target does not include any explicitly stated Security Assurance Requirements.

8.6 TOE Summary Specification Rationale

This section identifies the security functions provided by the TOE mapped to the security functional requirement components contained in this ST. This mapping is provided in the following table.

Security Function	Security Functional Components
Access Control	FDP_ACC.1 Subset access control
	FDP_ACF.1 Security attribute based access control
Identification and Authentication	FIA_ATD.1 User attribute definition
	FIA_UAU.2 User authentication before any action
	FIA_UAU.7 Protect authentication feedback

Security Function	Security Functional Components
	FIA_UID.2 User identification before any action
	FIA_AFL.1 Authentication Failure Handling
	FIA_SOS.1 Verification of Secret
Audit	FAU_GEN.1 Audit data generation
	FAU_GEN_EXP.1(1) Audit data generation
	FAU_GEN_EXP.1(2) Audit data generation
	FAU_GEN_EXP.1(3) Audit data generation
	FAU_GEN.2 User identity association
	FAU_SAR.1 Audit review
	FAU_SAR.1(1) Audit review
	FAU_SAR.1(2) Audit review
	FAU_SAR.1(3) Audit review
	FAU_SAR.2 Restricted audit review
	FAU_SAR.3 Selectable audit review
	FAU_SEL.1 Selective audit
	FAU_STG.1 Protected audit trail storage
Security Management	FMT_MOF.1 Management of security functions behaviour
	FMT_MSA.1 Management of security attributes
	FMT_MSA.2 Secure security attributes
	FMT_MSA.3 Status attribute initialization
	FMT_MTD.1(1) Management of TSF data
	FMT_MTD.1(2) Management of TSF data
	FMT_SMF.1 Specification of management functions
	FMT_SMR.1 Security roles
Trusted Communications	FCS_CKM.1(1) Cryptographic key generation
	FCS_CKM.1(2) Cryptographic key generation
	FCS_CKM.4 Cryptographic key generation
	FCS_COP.1 Cryptographic operation
Protection of the TSF	FPT_RVM_EXP_TOE.1 Non-bypassability of the TSP: TOE
	FPT_TRP.1 Trusted Path

Security Function	Security Functional Components
	FPT_ITT.1 Basic Internal TSF data transfer protection
	FPT_SEP_EXP_TOE.1 TSF Domain Separation: TOE
	FPT_RVM_EXP.1 Non-bypassability of TSP: IT Environment
	FPT_SEP_EXP.1 TSF domain separation: IT Environment
	FPT_STM.1 Reliable time stamps
Compliance Management	FCM_JOB_EXP.1 (1) Compliance Management Jobs
	FCM_JOB_EXP.1 (2) Compliance Management Jobs
	FCM_JOB_EXP.1 (3) Compliance Management Jobs
	FCM_JOB_EXP.1 (4) Compliance Management Jobs
	FCM_JOB_EXP.1 (5) Compliance Management Jobs

Table 10 – SF to SFR Mapping

8.6.1 Access Control

The access control function of the BladeLogic Operations Manager v 7.4.2 TOE enforces the FDP_ACC.1 and FDP_ACF.1 requirements.

BladeLogic Operations Manager v 7.4.2 enforces a BladeLogic Role Based Access Control Policy, which restricts access to the management functions of the TOE. This protection requires that users of the TOE be authenticated prior to any access to the management functions is granted. BladeLogic Operations Managers uses SRP to authenticate the user, and ACLs to restrict that user to certain functionality and access within the TOE. It further restricts a user from assuming more than one role at a time in order to ensure strict compliance with the BladeLogic Role Based Access Control Policy.

The BladeLogic Application Server enforces access to objects throughout the Configuration Manager/ console. After role-level authorizations and server permissions have been defined in Configuration Manager the Application Server converts the role and object-level authorizations to Access Control Lists that are then pushed down to RSCD Agents. There the information is converted into entries in a configuration file that restricts user access to the RSCD Agent, which will control all incoming connections to RSCD Agents, including connections from Network Shell.

In addition to role-based authorizations, the Configuration Manager is used to grant authorizations to roles to perform specific actions on controlled objects. To take an action on any object, the user must have role level authorization as well as object-level authorization (see section 6.1.2.2 Object Based Permissions), which allows for fine-grained access to objects throughout the system.

BladeLogic uses a system of role-level and object-level authorizations throughout the Configuration Manager that grant permissions to perform actions on objects.

8.6.2 Identification and Authentication

The identification and authentication function of the BladeLogic Operations Manager v 7.4.2 TOE enforces the FIA_ATD.1, FIA_UAU.2, FIA_UAU.7, FIA_AFL.1, FIA_SOS.1and FIA_UID.2 requirements.

The BladeLogic Operations Manager v 7.4.2 enforces not only role-based authorizations, but also a list of security attributes belonging to individual users. In addition, BladeLogic Operations Manager enforces permissions that must be defined for every object in the Configuration Manager and requires each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user. When users authenticate, the system provides only limited feedback in the form of the number of characters typed appearing as asterisks during authentication.

The TOE supports BladeLogic Role Based Access Control policy through Internet Protocol (IP)-based authentication between the Application Server and RSCD Agents. By defining a setting in the exports file, incoming traffic can be limited to communication from specified BladeLogic Application Servers.

The TOE defines a number of failed login attempts which cannot be exceeded before an account is locked out. If an account is locked out, the TOE invokes a delay or timeframe which prevents the user from attempting to gain access until the timeframe is exceeded or a system administrator sets the account. The TOE ensures the user provides a password with a minimal character length in order to gain access to the TOE.

The user uses the client to authenticate to the Authentication Service, which listens on port 9840. In the evaluated configuration, the user specifies Secure Remote Password (SRP). The Authentication Service uses information in its database including a password authenticator to validate user's protocol exchanges, which proves the user has knowledge of the password. The Authentication Service uses the Diffie-Helman exchange to authenticate the user's password and then issues an SSO credential and sends it back to the client software. This is done over a TLS connection.

8.6.3 Audit

The security audit function of the BladeLogic Operations Manager v 7.4 TOE enforces the FAU_GEN.1, FAU_GEN_EXP.1(1), FAU_GEN.1(2), FAU_GEN.1(3), FAU_GEN.2, FAU_SAR.1, FAU_SAR_EXP.1(1), FAU_SAR_EXP.1(2), FAU_SAR_EXP.1(3), FAU_SAR.2, FAU_SAR.3, FAU_STG.1 and FAU_SEL.1 requirements. FAU_GEN.1 requires a reliable time-stamp, which is provided by FPT_STM.1.1 (provided by the IT environment).

The primary purpose of the audit security function is to generate auditable events as configured by authorized users. This is achieved by capturing events into a record that is stored in the SQL Server Database.

Operations Manager provides a GUI interface to select an object to see auditable events. An audit trail is a record of who has sought authorization for specific actions. The user can specify whether an audit trail entry is recorded every time a user is successfully authorized for an action, every time a user is denied authorization, or both. Audit trail settings apply globally throughout Operations Manager. The audit trail also records the role that was trying to access the object, and the user using the role in the current session. The Application Server processes the audit trail, although it can be configured to process on a different server by the RBACAdmins. The audit trails are then stored in the SQL Server Database and accessed via the BladeLogic Core Database. When users run jobs to report audit activities, the BladeLogic Reports Server reads data from the SQL Server Database via the Reporting Data Warehouse.

The audit security function also is responsible for storing reports about remote server configuration settings in the SQL Server Database. The reports are generated via running either, snapshot jobs, audit jobs, or patch analysis jobs. The information captured as a result of these jobs running are used by the compliance management function to correct any deficiencies against a predetermined policy. See the configuration management section for more details on this process.

8.6.4 Security Management

The security management function of the BladeLogic Operations Manager v 7.4.2 TOE enforces the FMT_MOF.1, FMT_MSA.1, FMT_MSA.2, FMT_MSA.3, FMT_MTD.1(1), FMT_MTD.1 (2), FMT_SMF.1, and FMT_SMR.1 requirements.

The BladeLogic Operations Manager v 7.4.2 enforces security management through the BladeLogic Role Based Access Control Policy by restricting the ability to determine the behavior of, disabling, enabling, or modifying event logging for audit functions to only authorized users. It further restricts the ability to change the default, query, modify, or delete the security attributes such as event logging, roles, and authorizations to only authorized users, and provides restrictive default values for security attributes. The system also restricts the ability to delete the audit records and user job runs to authorized users, and is capable of performing security management functions, while associating users with roles. Users are only allowed to use one role at a time.

8.6.5 Self Protection

The self protection function of the BladeLogic Operations Manager v 7.4.2 TOE enforces the FPT_TRP.1, FPT_ITT.1, FPT_STM.1, FPT_RVM_EXP.1, FPT_SEP_EXP.1, FPT_RVM_EXP_TOE.1 and FPT_SEP_EXP_TOE.1 requirements.

FPT_RVM_EXP_TOE.1 is implemented by the Self-Protection Function. The TOE makes sure that security enforcing functions are invoked and succeed before allowing any other mediated action to occur.

FPT_SEP_EXP_TOE.1 is implemented by the Self-Protection Function. The Self-Protection Function provides a protected execution domain and a separation of traffic streams traversing the TOE. The TOE is a dedicated device, with no general purpose operating system, or programming interface. No untrusted processes are permitted on the TOE.

FTP_TRP.1 ensures communications between the TOE and the users are encrypted. The trusted path is initiated by the remote user and is used for initial authentication and user management.

FPT_ITT.1 Ensures communications between components of the TOE are encrypted.

FAU_STG.1 requires that the underlying SQL Server protect the audit records generated by the TOE that are stored in the BladeLogic Core Database and Reporting Data Warehouse.

FPT_SEP_EXP.1 requires that the operating environment provide an isolated domain for the TOE to operate.

FPT_RVM_EXP.1 requires that the operating environment provide mechanisms to prevent bypassing the security features provided and enforced by the TOE.

FCS_CKM.1 (1) and FCS_CKM.4 are implemented by the Self Protection Function. The TOE implements the cryptographic key generation and destruction functions of the SSL protocols to protect the communication channel for remote administration. FCS_COP.1 is implemented by the Self Protection Function. The TOE implements SSL crypto operations to protect the communication channel for remote administration. All cryptography has been asserted as tested by BladeLogic. Cryptographic keys used by the TOE live 1000 days and only relevant to a specific

session. Keys are also destroyed permanently through admin guidance which describes the proper use of the keytool utility as an available method of destruction.

8.6.6 Compliance

The compliance function of the BladeLogic Operations Manager v 7.4.2 TOE enforces the FCM_JOB_EXP.1 (1), FCM_JOB_EXP.1 (2), FCM_JOB_EXP.1 (3), FCM_JOB_EXP.1 (4), FCM_JOB_EXP.1 (5), requirements.

The primary purpose of the Compliance function is to provide the TOE and the IT environment with applicable patches, hotfixes, and service packs needed according to a snapshot. A snapshot is a record of the configuration of a group of servers or server objects at a point in time. The administrator can specify what objects or remote servers are included in the snapshot. After a patch analysis configuration comparison is performed, any updates or changes can be made through File Deploy, Deploy Jobs, or Network Shell Script Jobs.

The compliance function is also responsible for storing snapshots and reports about remote server configuration settings in the SQL Server Database. These reports are generated via running either, snapshot jobs, audit jobs, or patch analysis jobs. The information captured as a result of these jobs running are used by the compliance management function to correct any deficiencies against a predetermined policy.

8.6.7 Trusted Communications

The trusted communications function of the BladeLogic Operations Manager v 7.4 TOE enforces the FCS_CKM.1(1) , FCS_CKM.1(2), FCS_CKM.4, and FCS_COP.1 requirements.

FCS_CKM.1(1), and FCS_CKM.4 are implemented by the Trusted Communications Function. The TOE implements the cryptographic key generation and destruction functions of the TLS protocol to protect the communication channel for either remote administration (via an internet browser) or internal TOE communications between the Application Servers and the RSCD Agents. FTP_TRP, FTP_ITT and FCS_COP.1 ensures the TOE implements TLS crypto operations to protect the communication channel for remote administration.

FCS_CKM.1(2) and FCS_CKM.4 are implemented by the Trusted Communications Function. The TOE implements the cryptographic key generation and destruction functions of the SRP protocol to protect the internal TOE communication channels passing user credentials. FTP_ITT and FCS_CKM.1(2) ensure the TOE implements TLS crypto operations to protect the communication channel for internal TOE communications between the Configuration Manager and BladeLogic CLI clients; and the BladeLogic Application Server.

For SRP and TLS, the FMT_MSA.2 ensures the TOE implements crypto operations using valid parameter values. The cryptography used in this product has not been FIPS certified nor has it been analyzed or tested to conform to cryptographic standards during this evaluation. All cryptography has only been asserted as tested by the vendor."

8.7 Strength of Function Rationale

The strength of function requirement applies to FIA_SOS.1. The SOF claim for FIA_SOS.1 is SOF-basic. The strength of the “secrets” mechanism is consistent with the objective of the password quality parameters (section 6.1.7.1 Password Policy). Strength of Function shall be demonstrated for the non-certificate based authentication mechanisms to be SOF-basic, as defined in Part 1 of the CC. Specifically, the authentication mechanism must demonstrate adequate protection against attackers possessing a low-level attack potential. The environment is one where the potential attacker is unsophisticated, with access to only standard equipment and public information about the product, consistent with a Common Criteria Level of Evaluation of EAL3. Specifically, AVA_VLA.1 requires that the TOE be resistant to an attacker with a low to moderate attack potential, this is consistent with SOF-basic. Consequently, the metrics (password) chosen for inclusion in this ST for this TOE were determined to be acceptable

for SOF-basic and would adequately protect information in a Basic Robustness Environment. Specifically, this environment is protected, and isolated network (see A.LOCATE in section 3.2).

As stated in the TSS, there is one security function (FIA_UAU.2) based on probabilistic methods. A policy for selecting a strong password for user authentication to meet this claim is described in the administrator guidance. While the TOE can enforce this policy through the I&A security function via the FIA_AFL.1 and FIA_SOS.1.

8.8 PP Claims Rationale

This Security Target does not claim Protection Profile conformance.