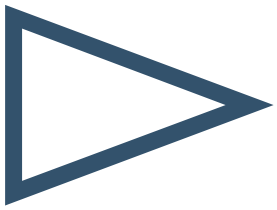


BMC Server Automation v8.3

Security Target

Version 0.15

March 17, 2015

bmcsoftware
BUSINESS RUNS ON I.T.™

Document Revision History

Date	Revision	Author	Changes made
23 August 2013	0.01	Chandra Bridges	Initial Draft.
28 August 2013	0.02	Chandra Bridges	Updates made to address comments from the Team.
16 September 2013	0.03	Chandra Bridges	Minor format and other corrections.
4 October 2013	0.04	Mark Gauvreau	Small change to TOE name/version
31 January 2014	0.05	TM	Addressed Evaluator's comments
13 February 2014	0.06	TM	Addressed Evaluator's comments
14 February 2014	0.07	TM	Modified Crypto claims
15 April 2014	0.08	TM	Version change
8 May 2014	0.09	TM	NSH Proxy service and audit clarifications
14 May 2014	0.10	TM	Clarifications to Section 7.5
7 July 2014	0.11	TM	Addressed certifier's comments
18 July 2014	0.12	TM	Addressed evaluator's comments
3 March 2015	0.13	TM	Updated TOE version
10 March 2015	0.14	TM	Response to additional evaluation
17 March 2015	0.15	TM	Addressed evaluator's comments

TABLE OF CONTENTS

1 SECURITY TARGET INTRODUCTION

This section presents Security Target (ST) identification information and an overview of the ST for *BMC Server Automation v8.3* (hereinafter referred to as *BSA*).

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. An ST principally defines:

- A security problem expressed as a set of assumptions about the security aspects of the environment, a list of threats that the product is intended to counter, and any known rules with which the product must comply (Security Problem Definition section).
- A set of security objectives and a set of security requirements to address the security problem (Security Objectives and Security Requirements sections, respectively).
- The IT security functions provided by the TOE that meet the set of requirements in the TOE Summary Specification section.

The structure and content of this ST comply with the requirements specified in Annex A Specification of Security Targets of [CCP1] and Section 11 Class ASE: Security Target evaluation of [CCP3].

1.1 ST Reference

ST Title:	BMC Server Automation v8.3 Security Target
ST Version:	Version 0.15
ST Date:	17 March 2015

1.2 TOE Reference

TOE Identification:	BMC Server Automation v8.3.03 build 190
TOE Developer	BMC Software, Inc.
TOE Type	Automated Server Configuration Management

1.3 Document References

The following references are used in this ST:

Abbreviation	Document
[CC]	Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 4, September 2012, CCMB-2012-09-(001 to 003)
[CCP1]	Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; CCMB-2012-09-001, Version 3.1, Revision 4, July 2012
[CCP2]	Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components; CCMB-2012-09-002, Version 3.1, Revision 4, September 2012
[CCP3]	Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Components; CCMB-2012-09-003, Version 3.1, Revision 4, September 2012
[CEM]	Common Methodology for Information Technology Security Evaluation; CCMB-2012-09-004, Version 3.1, Revision 4, September 2012
[FIPS140-2]	FIPS PUB 140-2. Security Requirements for Cryptographic Modules. May 2001
[FIPS180-3]	FIPS PUB 180-3. Secure Hash Standard (SHS). October 2008
[FIPS186-2]	FIPS PUB 186-2. Digital Signature Standard (DSS). January 2000
[FIPS197]	FIPS PUB 197. Advanced Encryption Standard. November 2001
[FIPS198-1]	FIPS PUB 198-1. The Keyed-Hash Message Authentication Code (HMAC). July 2008

Abbreviation	Document
[RFC 2313]	Request for Comments: 2313, PKCS #1: RSA Encryption, March 1998

1.4 Document Conventions

Section 8.1 in [CCP1] defines the approved set of operations that can be applied to the CC functional and assurance components: *assignment, refinement*, *selection,* and *iteration.* In this ST, these operations are indicated as follows:

1) The assignment operation is used to assign a specific value to an unspecified parameter, such as the length of a password. Showing the value in square brackets [assignment_value] indicates an assignment. In the case when an assignment operation is embedded in a selection operation, the operations will be denoted as follows: *selection value [assignment_value]*. Assignments in the Extended Components Definition are shown in square brackets and italics as follows: [assignment: *assignment details*].

2) The refinement operation is used to add detail or refine a requirement, and thus further restricts a requirement. Refinement of security requirements is denoted by **bold text** for new text and ~~strikethrough text~~ for deleted text.

3) The selection operation is used to select one or more options provided by the CC in stating a requirement. Selections are denoted by *underlined italicized text.*

4) Iterated security functional requirements will be identified by appending an additional identifier in round brackets next to their original identifier. For example: FMT_MTD.1(1) and FMT_MTD.1(2).

1.5 Document Terminology

1.5.1 CC Terminology

In the CC, many terms are defined in Section 4.1 of [CCP1]. The following terms are a subset of those definitions:

Term	Definition
Authentication data	The information used to verify the claimed identity of a user.
Authorized user	A TOE user who may, in accordance with the SFRs, perform an operation.
External entity	A human or IT entity possibly interacting with the TOE from outside of the TOE boundary.
Identity	A representation uniquely identifying entities (e.g. a user, a process or a disk) within the context of the TOE An example of such a representation is a string. For a human user, the representation can be the full or abbreviated name or a (still unique) pseudonym.
Object	A passive entity in the TOE, that contains or receives information, and upon which subjects perform operations.
Operation (on an object)	A specific type of action performed by a subject on an object.
Role	A predefined set of rules establishing the allowed interactions between a user and the TOE.
Security function policy	A set of rules describing specific security behaviour enforced by the TSF and expressible as a set of Security Functional Requirements (SFRs).
Security objective	A statement of an intent to counter identified threats and/or satisfy identified organisation security policies and/or assumptions.
Security requirement	A requirement, stated in a standardised language, which is meant to contribute to achieving the security objectives for a TOE.
Subject	An active entity in the TOE that performs operations on objects.
Target of evaluation	A set of software, firmware and/or hardware possibly accompanied by guidance.
TOE security functionality	The combined functionality of all hardware, software, and firmware of a TOE that must be relied upon for

Term	Definition
	the correct enforcement of the SFRs.
TSF interface	The means by which external entities (or subjects in the TOE but outside of the TSF) supply data to the TSF, receive data from the TSF and invoke services from the TSF.
User	See external entity defined above.

1.5.2 Abbreviations

The following acronyms are used in this ST:

Term	Definition
ACL	Access Control List
AES	Advanced Encryption Standard
BDSSA	BMC BladeLogic Decision Support for Server Automation
BLCLI	BMC Server Automation Command Line Interface
BSA	BMC Server Automation
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher-Block Chaining
CC	Common Criteria
CM	Configuration Management
CMVP	Cryptographic Module Validation Program
CLI	Command Line Interface
CVS	Comma-Separated Value
DBMS	Database Management System
DH	Diffie-Helman
EAL	Evaluation Assurance Level
ETL	Extract, transform and load
FIPS	Federal Information Processing Standard
GUI	Graphical User Interface
HMAC	Keyed-Hash Message Authentication Code
HTTP	Hyper Text Transfer Protocol
HTTPS	Hyper Text Transfer Protocol Secure
IT	Information Technology
JRE	Java RunTime Environment
NSH	Network Shell
OS	Operating System
PP	Protection Profile
RFC	Request For Comment

Term	Definition
RBAC	Role-Based Access Control
RSCD	Remote System Call Daemon
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SFP	Security Function Policy
SHS	Secure Hash Standard
SSL	Secure Socket Layer
ST	Security Target
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Functionality

1.5.3 BSA Terminology

The following additional terms are specific to this ST:

Term	Definition
Audit job	The function of comparing a managed server to a known configuration.
Audit trail	The audit trail maintained by the TOE.
Appserver/RSCD logs	These are log4j logs recorded in the Application Server and RSCD Agents.
Authorization	Authorization refers to permissions granted within BSA to perform a function. The terms 'authorization' and 'permissions' are used interchangeably throughout the ST.
BLAdmin	The built-in user which is assigned the BLAdmins role.
BLAdmins	A built-in role with authorizations granting the user in this role permission to change permissions for all system objects.
BLPackage	A BMC Server Automation package containing files, configuration entries, etc.
Client	The term 'client' or 'client application' refers to any of the administrative interfaces: BMC Server Automation Command Line Interface (BLCLI), Console, web interface to the Reports Server, or Network Shell.
Provisioning	The remote installation of operating systems or applications from the Application server to an RSCD Agent Server.
RBACAdmin	The built-in user which is assigned the RBACAdmins role.
RBACAdmins	A built-in role with authorizations granting the user in this role permission to read and modify Access Control List (ACL) authorizations for all system objects in BMC Server Automation.
Server	A server is a machine where Remote System Call Daemon (RSCD) Agent software was installed.

1.6 TOE Overview

BMC Server Automation (BSA) product is a comprehensive system for the initial provisioning and ongoing automated management of data center servers. Using this system, administrators can provision and configure servers by deploying operating systems, applications, files, and configuration

information. BSA lets administrators manage servers with a consistent user experience, regardless of whether the servers are physical or virtual. Administrators can also:

- View, manage, and store server configurations
- Deploy software, patches, and complex packages of files and other assets
- Compare servers to detect discrepancies in their configurations
- Measure and enforce compliance to organizational standards
- Apply configuration changes to servers
- Share routine server management tasks among functional teams in an organization

BSA v8.3 integrates configuration automation and compliance assurance, enabling the implementation of policy-based automation and providing a single platform for managing physical and virtual servers. The solution addresses three main areas: configuration, provisioning, and compliance.

1.6.1 Configuration

Configuration management tasks include patching, configuring, updating, and reporting on servers. BSA enables consistency in change and configuration management activities. Subject to security constraints, it exposes sufficient detail about servers under management to ensure effective and accurate administrative activities. Configuration management capabilities include:

- Providing visibility into configurations and changes across major operating systems, and virtualization and cloud platforms
- Policy definitions for automated updates to patches, packages, and configurations

1.6.2 Provisioning

BSA automates operating system (OS) installation and configuration with support for major operating systems, as well as virtualization and cloud platforms. It also provides the ability to choose the mechanism for delivering the OS: unattended install, image-based provisioning, or template-based provisioning on virtualization platforms. Provisioning capabilities include:

- Full-stack, automated provisioning in both traditional and cloud-based data centers
- Support for virtual template-based, image-based, and script-based provisioning

1.6.3 Compliance

BSA supports compliance maintenance by defining and applying configuration policies. When a server or application configuration deviates from policy, the necessary remediation instructions are automatically generated and packaged, and may be automatically or manually deployed on the server. Compliance capabilities include:

- Configuration remediation through repair, rollback, or configuration updates
- Integrated policy-exception documentation
- Pre-configured compliance policies for regulatory and security standards

1.6.4 Reporting

BSA reporting capabilities include:

- Reports for compliance, inventory, provisioning, patch, and deployment
- Consolidated virtualization reports across all hypervisor platforms
- Complex reporting capabilities, such as pivot tables, trending, and data correlation

- Role-based access control

1.7 TOE Description

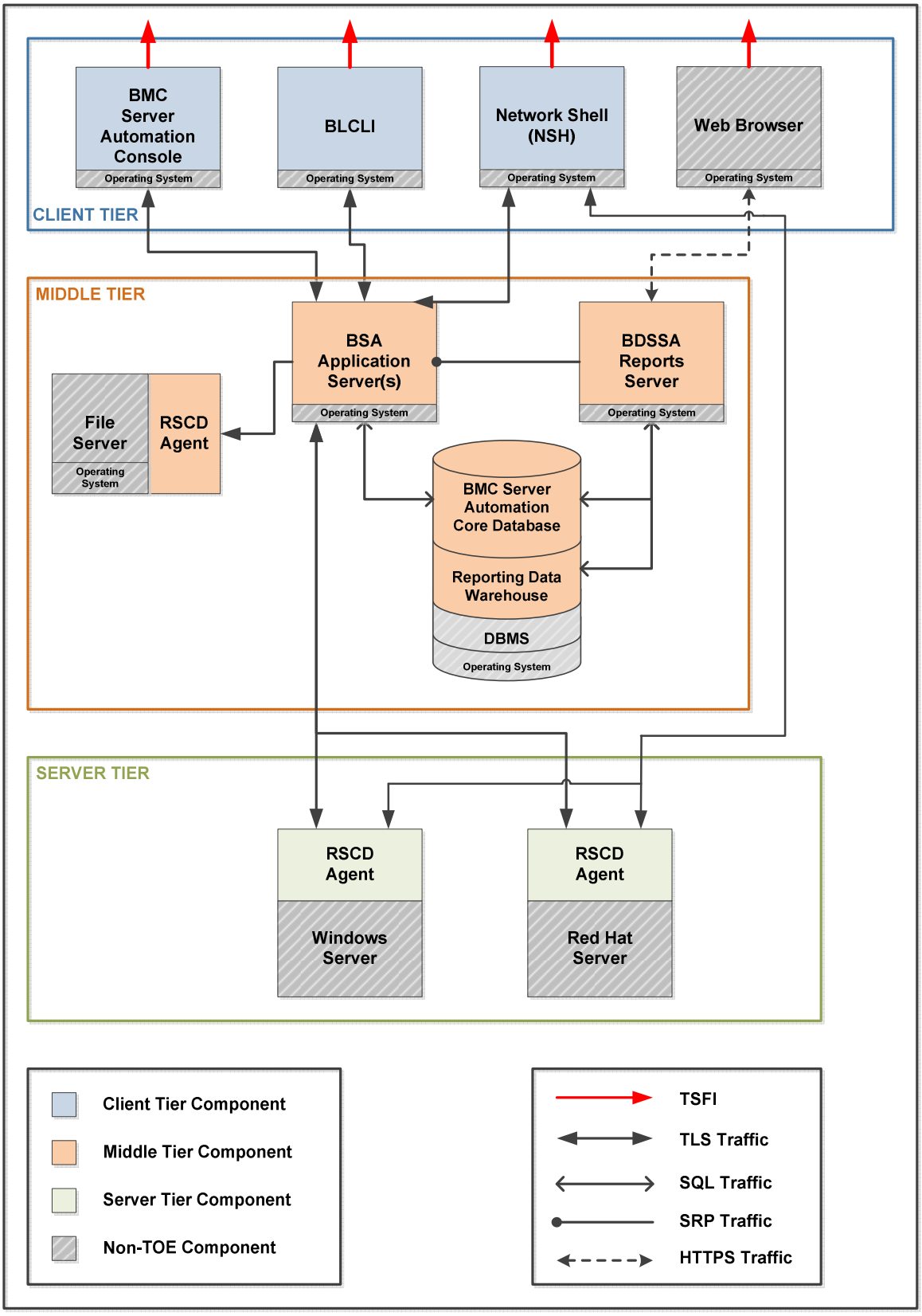
The following table identifies the BSA components and versions included in the evaluated configuration. The “abbreviated name” is used in this Security Target for discussion purposes.

Table 1 – BSA component names and versions

BSA component name and version	Version	Abbreviated name
BMC Server Automation Console	8.3.03.190	*BSA Console, Console*
Network Shell	8.3.03.190	*NSH(this is part of BSA Console, but is also installed separately with the BDSSA Reports Server)*
BMC Server Automation Application Server	8.3.03.190	*BSA Application Server, Application Server*
BMC BladeLogic Decision Support for Server Automation	8.3.03.1220*	*BDSSA Reports Server, Reports Server*
BMC Server Automation Remote System Call Daemon Agent	8.3.03.190	*RSCD Agent*

*With patch 190 and patch 194 to address the POODLE vulnerability

Figure 1– BMC Server Automation TOE Boundary



1.7.1 Physical scope and boundary

The BSA has a three-tier architecture that consists of client, server and middle tiers. Figure 1 illustrates the relationships between the major components of the three-tiered BSA system in the evaluated configuration of the TOE.

1.7.1.1 Client Tier

The BSA client tier describes the end user system access components and includes:

- The BSA console
- Access to the BLCLI
- Network Shell
- A web interface to the Reports server

1.7.1.1.1 BMC Server Automation Console (Console)

The BSA Console is a Graphical User Interface (GUI) that server administrators use for managing and automating data center procedures. Once users have been authenticated, their actions upon objects/resources are controlled via the BSA Role Based Access Control Policy; all authorizations that allow access to these objects/resources are controlled by this policy.

1.7.1.1.2 BMC Server Automation Command Line Interface (BLCLI)

The BLCLI allows BSA users to perform most procedures available in BSA Console from a command line rather than using the console itself. This interface requires the users be authenticated. Once users have been authenticated, their actions upon objects/resources are controlled via the BSA Role Based Access Control Policy; all authorizations that allow access to the objects/resources are controlled by this policy. The BLCLI software is part of the BSA Automation Server; however, the access to the CLI is conceptually included in the client tier.

1.7.1.1.3 Network Shell (NSH)

Network Shell (NSH) is a network-aware shell that enables cross-platform access through a command line interface. After successfully authenticating the Secure Remote Password (SRP) user, the Authentication Service issues the client a session credential. Once an authenticated NSH client has connected to an RSCD Agent, the client's authorizations are enforced by configuration files (ACLs) stored on each server.

1.7.1.1.4 Web Interface

The client tier includes the web interface to the Reports Server. The Reports Server is part of the middle tier; however, the access to the Reports Server is conceptually included in the client tier.

1.7.1.2 Middle Tier

The middle tier supports the BSA server components and supporting databases and includes the:

- BSA Application Server
- Data Warehouse
- Reports Server
- File Server

1.7.1.2.1 BMC Server Automation Application Server (Application Server)

The Application Server is the primary component in the BSA Architecture. It controls communication between the BSA console and remote servers, and controls interaction with the database and file servers. The Application Server utilizes TLS for session-layer security when communicating with the Console and the BLCLI in the client tier and RSCD Agents in the server tier.

1.7.1.2.2 BMC Server Automation Core Database and the Reporting Data Warehouse

There are two distinct databases, the Core Database and the Data Warehouse. The Application Server accesses the Core Database; the BDSSA Reports Server accesses the Data Warehouse.

The BSA Core Database stores the transactional data for the product which primarily includes management policies/jobs, their associated configuration/objects (e.g. depot items), list of enrolled servers, security information (including audit trail information), run/result data for previous executions of jobs and some infrastructure information (e.g. the application servers participating in load balancing).

The reporting Data Warehouse is the main store from which relevant data center reports (e.g. inventory reports, patch status) are generated by the reporting engine. The data is pushed to this database by performing an extract, transform and load (ETL) process on the transactional data stored in the Core Database. The custom reports information is also stored in the reporting warehouse.

1.7.1.2.3 BMC Decision Support for Server Automation (Reports Server)

The BMC Decision Support for Server Automation product is fundamentally a Data Warehouse. It provides the capability to store vast amounts of historical information over many years and allows users to create customized reports based on this information. Users access it using a local web browser over Hyper Text Transfer Protocol Secure Socket (HTTPS).

1.7.1.2.4 File Server

The BSA design requires a designated file server to host the files required by the Automation Server. The file server is simply a server (non-TOE component) running the RSCD Agent (TOE component). This may be combined with the Automation Server; however, these are separate servers in the evaluated configuration.

1.7.1.3 Server Tier

The BSA server tier consists of RSCD Agents on remote servers.

1.7.1.3.1 BMC Server Automation Remote System Call Daemon (RSCD) Agent

An RSCD Agent runs as a daemon (UNIX) or a service (Windows) on all servers managed by BSA. The Application Server communicates with RSCD Agents and initiates all communication to perform tasks. RSCD Agents never initiate communication with an Application Server or any other BSA component. The RSCD Agent runs commands on behalf of the Application Server and sends the results back to the Application Server.

1.7.1.4 Guidance Documentation

The TOE includes the following guidance documentation:

a. BMC Server Automation 8.3, 25-Feb-2014

b. BMC BladeLogic Decision Support for Server Automation 8.3, 05-Sep-2013

c. BMC Server Automation Command Line Interface 8.3, 28-Jul-2013

d. BMC Server Automation v8.3 Guidance Supplement, Version 0.03

1.7.2 Hardware and Software Requirements

The hardware requirements for any given environment depend on the size and amount of activity expected. This section describes the requirements used for the purposes of this evaluation.

The following table identifies the operating system and hardware required to support the TOE. The TOE components are part of the TOE. The operating system and hardware are part of the environment.

Table 2 – TOE Supporting Hardware and Software

TOE Component	Operating System	Hardware
BSA Console	Windows 2008 R2	General Purpose Computer Hardware dual-core x86, 2 GHz or greater

TOE Component	Operating System	Hardware
Network Shell	Windows 2008 R2	General Purpose Computer Hardware dual-core x86, 2 GHz or greater
BSA Application Server	Windows 2008 R2	General Purpose Computer Hardware 4 Xeon, 3 GHz or greater
BMC BladeLogic Decision Support for Server Automation	Windows 2008 R2	General Purpose Computer Hardware
RSCD Agent	Windows 2008 R2	General Purpose Computer Hardware 200 MB disk space
RSCD Agent	Red Hat Enterprise Linux 6	General Purpose Computer Hardware 200 MB disk space

The following table identifies other components in the operational environment required to support the TOE.

Table 3 – Operational Environment Hardware and Software

Environment Component	Supported Product
File Server	Windows 2008 R2 Operating System General Purpose Computer Hardware 72 GB disk space
Database	Microsoft SQL Server 2008 R2 on General Purpose Computer Hardware
Web Browser	Microsoft Internet Explorer 9.0 on any supported Windows Operating System on General Purpose Computer Hardware

1.7.3 Logical scope and boundary

The TOE provides the following security functions:

- Security Audit
- Cryptographic Support
- User Data Protection
- Identification and Authentication
- Security Management
- Protection of the TSF
- Trusted Path/Channels
- Compliance Management

1.7.3.1 Security Audit

The TOE maintains multiple records of events that occur within the TOE. The Security audit claims cover two of those facilities: the audit trail and the appserver/RSCD logs. The TOE allows authorized users to review records of events contained in the audit trail.

1.7.3.2 Cryptographic Support

The TOE uses TLS and HTTPS to support protected communications between TOE components and to the administrative interface. Cryptographic algorithms are implemented in support of those protections.

1.7.3.3 User Data Protection

The TOE enforces a 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.

1.7.3.4 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 Role Based Access Control Policy, which restricts access to the management functions of the TOE. This protection requires that users of the TOE be identified and authenticated before any access to the management functions is granted. User attributes are maintained in support of authentication.

The TOE enforces the configured password rules, obscures feedback when entering passwords and will lock a user out after an administrator defined number of unsuccessful login attempts.

1.7.3.5 Security Management

The TOE is managed through the Console, BLCLI, Network Shell and BDSSA Reports Server Client. Functionality is provided to manage the security attributes that affect role based access control. Management functionality is provided through the management interfaces to perform user account and role management and to configure compliance management options.

1.7.3.6 Protection of the TSF

The TOE uses TLS to protect the internal communications between the client-tier applications and the middle-tier Application Server and between the Application Server and the RSCD Agents.

1.7.3.7 Trusted Path/Channels

The TOE uses HTTPS to protect communications between the web browser and the BDSSA Reports Server.

1.7.3.8 Compliance Management and Reporting

The TOE provides automated server management functions to enforce compliance to configured standards.

1.7.4 Functionality Excluded from the Evaluated Configuration

The following are excluded from this evaluation:

- Active Directory (AD)/Kerberos authentication mechanism.

2 CONFORMANCE CLAIMS

2.1 Common Criteria Conformance Claim

This Security Target claims to be conformant to version 3.1 of Common Criteria for Information Technology Security Evaluation according to:

- Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; CCMB-2012-09-001, Version 3.1, Revision 4, September 2012 [CCP1]
- Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components; CCMB-2012-09-002, Version 3.1, Revision 4, September 2012 [CCP2]
- Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements CCMB-2012-09-003, Version 3.1, Revision 4, September 2012 [CCP3]

As follows:

- CC Part 2 extended
- CC Part 3 conformant

The Common Methodology for Information Technology Security Evaluation, Version 3.1, Revision 4, September 2012 [CEM] has to be taken into account.

2.2 Protection Profile Claim

This Security Target does not claim conformance to a validated Protection Profile.

2.3 Assurance Package Claim

This Security Target claims conformance to Evaluation Assurance Level 2 augmented with ALC_FLR.2.

3 SECURITY PROBLEM DEFINITION

3.1 Threats

Table 4 lists threats to the resources to be protected by the TOE. The threat agents to the TOE are considered to be unauthorized users with public knowledge of how the TOE operates and who possess the skills and resources to alter TOE configuration settings, or parameters, or both. The threat agents do not have physical access to the TOE. Mitigation to the threats is through the objectives identified in Section 4.1 Security Objectives.

Table 4 –Threats

Threat	Description
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.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.MASK	Users, whether they be malicious or non-malicious, could gain undetected, 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 Organizational Security Policies

There are no Organizational Security Policies that apply to the TOE.

3.3 Assumptions

The assumptions are delineated in Table 5 are required to ensure the security of the TOE:

Table 5 – Assumptions

Assumption	Description
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, willfully negligent, or hostile and will follow and abide by the instructions provided by the guidance documentation.
A.TRAINED_ADMIN	TOE Administrators have reviewed the documentation provided by BMC for secure delivery and management of the TOE.
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.

Assumption	Description
A.STORAGE	The operational environment will protect audit data stored using the TOE's underlying operating system, files stored in the file server and TSF data (including audit trail data) stored in the database.
A.TIME	The operational environment will provide a reliable time source for audit record generation.
A.PATCH	Hotfix and patch files required to perform patching and other compliance activities will be available to the TOE from the operational environment.

4 SECURITY OBJECTIVES

The purpose of the security objectives is to address all of the security concerns, and to show which security concerns are addressed by the TOE, and which are addressed by the environment. Threats can be directed against the TOE or the security environment or both. Therefore, the CC identifies two categories of security objectives:

- Security objectives for the TOE
- Security objectives for the environment

4.1 Security Objectives for the TOE

This section identifies and describes the security objectives for the TOE, as shown in Table 6.

Table 6 – Security objectives for the TOE

Security Objective	Description
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 and viewing 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 the configured security policy including files, registry, and patches.
O.EAVESDROPPING	The TOE will encrypt TSF data that is transmitted between parts of the TOE and from the TOE to remote users 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 required security policy. The TOE will collect and analyze critical configuration data of remote servers in the IT environment.
O.IMPERSONATE	The TOE will verify the identity of remote users before allowing access to TSF functionality.

4.2 Security Objectives for the Environment

This section identifies and describes the security objectives for the environment, as shown in Table 7.

Table 7 – Security objectives for the environment

Objective	Description
OE.FILESYS	The security features offered by the underlying Operating System and the DBMS will protect the audit records used by the TOE, the stored audit records generated by the TOE, the files used by the TOE and the TSF data contained in the database.
OE.TIMESTAMP	The operational environment for the audit mechanism of the TOE must provide a reliable time source for audit record generation.

Objective	Description
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 and logical access.
OE.ROBUST_ADMIN_GUIDANCE	BMC will provide administrators with the necessary information for secure delivery and management.
OE.UPDATES	The operational environment will provide any required hotfix and patch files required by the TOE to perform compliance management activities.

4.3 Security Objectives Rationale

This section demonstrates that all security objectives for the TOE are traced back to aspects of the identified threats to be countered by the TOE, and that all objectives for the environment are traced back to assumptions for the environments.

Table 8 – Security objective to threats and assumptions correspondence

	Threats							Assumptions						
	T.MISCONFIG	T.MANAGE	T.ACCESS	T.MODIFY	T.MASK	T.EAVESDROPPING	T.UNAUTH	A.ADMIN	A.LOCATE	A.NOEVIL	A.STORAGE	A.TIME	A.TRAINED_ADMIN	A.PATCH
O.ACCESS			X											
O.AUDIT		X	X		X									
O.AUTH					X		X							
O.COMPLIANCE	X													
O.EAVESDROPPING						X								
O.MANAGE		X		X										
O.MONITOR	X													
O.IMPERSONATE					X									
OE.FILESYS											X			
OE.TIMESTAMP												X		
OE.ADMIN								X						
OE.MANAGE								X		X			X	
OE.NOEVIL										X				
OE.LOCATE									X					
OE.ROBUST_ADMIN_GUIDANCE													X	
OE.UPDATES														X

Table 9 – Security objectives rationale for the TOE

Objective	Threat	Rationale
O.ACCESS	**T.ACCESS**	O.ACCESS addresses T.ACCESS by providing the authorized users with the capability to specify access restrictions on the protected TOE resources to authenticated users.
O.AUDIT	**T.MASK**	O.AUDIT addresses T.MASK by providing the authorized users with tools necessary to monitor user activity to detect unauthorized access to the TOE.
	T.MANAGE	O.AUDIT addresses T.MANAGE by detecting mismanagement of user accounts or implementation of noncompliant policies.
	T.ACCESS	O.AUDIT addresses T.ACCESS by detecting unauthorized access by otherwise authorized users.
O.AUTH	**T.UNAUTH**	O.AUTH addresses T.UNAUTH by providing measures to uniquely identify and authenticate users prior to granting TOE access..
	T.MASK	O.AUTH addresses T.MASK by ensuring that users are identified and authenticated prior to being granted access to the TOE.
O.COMPLIANCE	**T.MISCONFIG**	O.COMPLIANCE mitigates this threat by having the TOE remotely patch and change configuration settings of remote servers on the local network.
O.EAVESDROPPING	**T.EAVESDROPPING**	O. EAVESDROPPING mitigates T.EAVESDROPPING by ensuring that all communication to and from the TOE or between components of the TOE are encrypted to prevent unauthorized access.
O.MANAGE	**T.MANAGE**	O.MANAGE addresses T.MANAGE by ensuring that authorized users are provided resources to correctly manage and monitor user accounts according to security and/or regulatory policies.
	T.MODIFY	O.MANAGE 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.
O.MONITOR	**T.MISCONFIG**	O.MONITOR mitigates this threat by having the TOE remotely monitor the configuration settings of remote servers on the local network.
O.IMPERSONATE	**T.MASK**	O.IMPERSONATE mitigates this threat by ensuring that the identity of remote users is verified before allowing access to TSF functionality.

Table 10 – Environment security objectives rationale for the TOE

Objective	Assumption	Rationale
OE.FILESYS	**A.STORAGE**	OE.FILESYS addresses A.STORAGE by ensuring that the underlying Operating System provides the capability to store and protect the audit records generated by the TOE and files used by the TOE, as well as the TSF data stored in the database.
OE.TIMESTAMP	**A.TIME**	OE.TIMESTAMP addresses A.TIME by ensuring that the underlying OS includes a system clock that provides reliable date and time which may be included in the audit records
OE.ADMIN	**A.ADMIN**	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.
OE.MANAGE	**A.ADMIN**	OE.MANAGE upholds A.ADMIN by ensuring that individuals are assigned to manage the TOE.
	A.NOEVIL	OE.MANAGE upholds A.NOEVIL by ensuring that the individuals assigned to manage the TOE are competent.
	A.TRAINED_ADMIN	OE.MANAGE upholds A.TRAINED_ADMIN by ensuring that the individuals

Objective	Assumption	Rationale
		assigned to manage the TOE are competent, and will therefore have reviewed the appropriate guidance.
OE.NOEVIL	**A.NOEVIL**	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.
OE.LOCATE	**A.LOCATE**	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 and logical access.
OE.ROBUST_ADMIN_GUIDANCE	**A.TRAINED_ADMIN**	OE.ROBUST_ADMIN_GUIDANCE directly maps to A.TRAINED_ADMIN to ensure that all administrators/users have reviewed the documentation provided by BMC for the BSA.
OE.UPDATES	**A.PATCH**	OE.UPDATES maps to A.PATCH to ensure that any hotfix or patch files required to perform compliance management are available to the TOE.

5 EXTENDED COMPONENTS DEFINITION

This section specifies the extended Security Functional Requirements (SFRs) used in this ST.

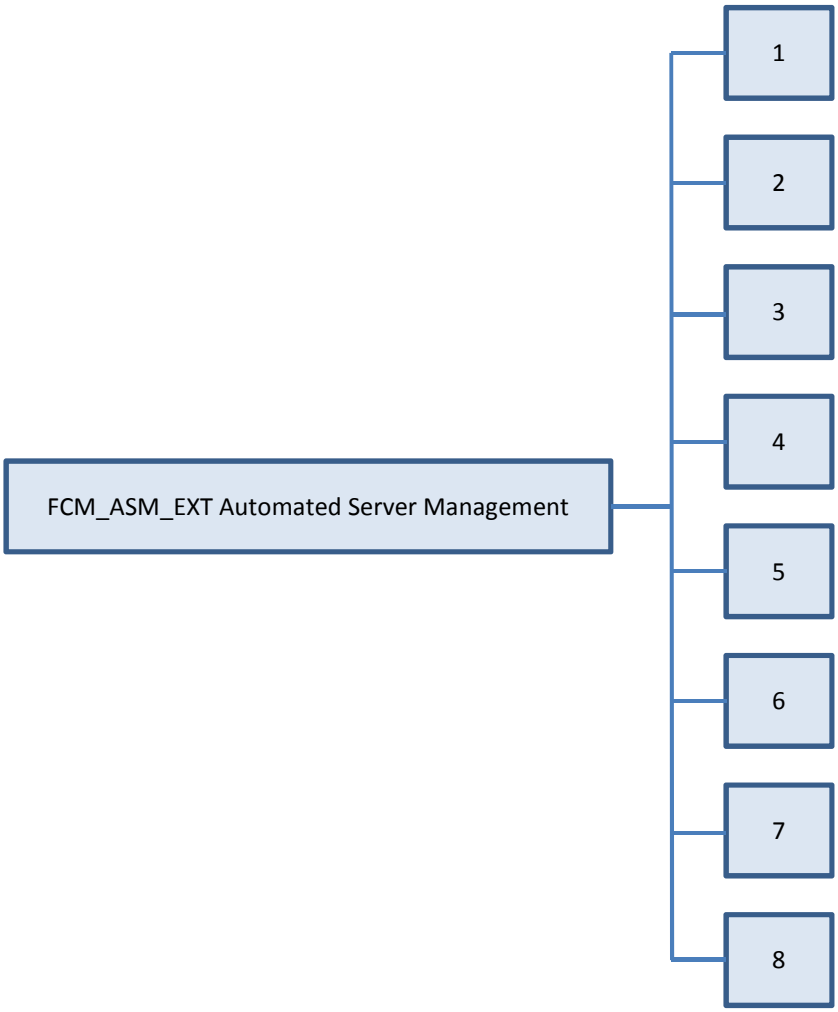
5.1 Class FCM Compliance management

The Compliance management class addresses automated server management functions required to enforce compliance requirements. The class is modeled on the FAU Security Audit class. Three families, Automated server management, Server management review and Compliance reporting, are defined for this class. The Automated server management family is modeled after FAU_GEN Security audit data generation. FCM_ASM_EXT.1, FCM_ASM_EXT.2 and FCM_ASM_EXT.3 were modeled after FAU_GEN.1. The Server management review family is modeled after FAU_SAR Security audit review. FCM_SMR_EXT.4, FCM_SMR_EXT.5, and FCM_SMR_EXT.6 were modeled after FAU_SAR.1. FCM_ASM_EXT.4 through FCM_ASM_EXT.8 were loosely based on FAU_GEN.1. The Compliance reporting family is modeled after FAU_SAR Security audit review, and FCM_CRP_EXT.1 is modeled after FAU_SAR.1 Audit review.

5.1.1 Automated Server Management Family Behaviour

This family defines the requirements for automated server management activities. These activities verify and enforce configuration policies.

Figure 2 – Component Leveling FCM_ASM_EXT



FCM_ASM_EXT.1 Snapshot reporting defines the requirements for generating a Snapshot report.

FCM_ASM_EXT.2 Audit job defines requirements for generating an Audit job.

FCM_ASM_EXT.3 Patch analysis job defines the Patch Analysis job.

FCM_ASM_EXT.4 Automated server management defines the requirements for defining a Compliance job.

FCM_ASM_EXT.5 Deployment of files addresses the deployment (push) of multiple files and directories to one or more managed servers.

FCM_ASM_EXT.6 Deployment of content addresses the deployment (push) of content to one or more managed servers.

FCM_ASM_EXT.7 Network shell script addresses the deployment and execution of previously saved network shell scripts.

FCM_ASM_EXT.8 Batch jobs addresses the deployment and execution of Batch jobs.

5.1.2 Server Management Review Family Behaviour

This family defines the requirements for reviewing the data produced by automated server management activities.

Figure 3 – Component Leveling FCM_SMR_EXT

FCM_SMR_EXT.1 Snapshot review defines the requirements for the capability to review the data resulting from Snapshot jobs.

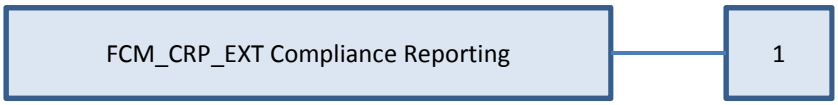
FCM_SMR_EXT.2 Audit job review defines the requirements for the capability to review the data resulting from Audit jobs.

FCM_SMR_EXT.3 Patch analysis job review defines the requirements for the capability to review the data resulting from Patch Analysis jobs.

5.1.3 Compliance Reporting Family Behaviour

This family defines the requirements for reporting on the results of compliance activities.

Figure 4 – Component Leveling FCM_CRP_EXT



FCM_CRP_EXT.1 Compliance reports addresses the creation of reports in support of compliance reporting.

5.1.4 Snapshot reporting (FCM_ASM_EXT.1)

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

- Snapshot jobs may be created, configured and scheduled to run periodically.

Audit Activity: The following actions should be auditable (FAU):

- Execution of snapshot jobs and the results of each execution (i.e., the delta between the latest snapshot and the previous one).

FCM_ASM_EXT.1	**Snapshot reporting**
Hierarchical to:	No other components

Dependencies:	FPT_STM.1 Reliable time stamps
FCM_ASM_EXT.1.1	The TSF shall be able to generate a snapshot report based on server objects.
FCM_ASM_EXT.1.2	The TSF shall record within each server object snapshot report at least the following information: [assignment: *report information fields*].

5.1.5 Audit job (FCM_ASM_EXT.2)

Management Activity: The following actions should be considered for FMT:

- Audit jobs can be created, configured and scheduled to run periodically.

Audit Activity: The following actions should be auditable (FAU):

- Execution of audit jobs and the results of each execution (i.e., the data between the latest server snapshot and the master server/master snapshot).

FCM_ASM_EXT.2	**Audit job**
Hierarchical to:	No other components
Dependencies:	FPT_STM.1 Reliable time stamps
FCM_ASM_EXT.2.1	The TSF shall be able to compare snapshots to determine whether servers match a standard configuration.
FCM_ASM_EXT.2.2	The TSF shall be able to generate an 'Audit Report' based on the following items listed in the snapshot report, server configuration file: a) [assignment: *Audit Report information*].
FCM_ASM_EXT.2.3	The TSF shall record within each entry of the 'Audit Report' at least the following information: a) [Assignment: *Audit Report details*]; and b) For each 'Audit Report', based on the auditable event definitions of the functional components included in the ST, [assignment: *other relevant report information*].

Application Note: 'Audit Report' refers to the report resulting from an 'Audit Job' and should not be confused with the audit logs.

5.1.6 Patch analysis job (FCM_ASM_EXT.3)

Management Activity: The following actions should be considered for FMT:

- Patch Analysis jobs can be created, configured and scheduled to run periodically.

Audit Activity: The following actions should be auditable (FAU):

- Execution of Patch Analysis jobs and the results of each execution (e.g., the missing patches on each analyzed server).

FCM_ASM_EXT.3	**Patch analysis job**
Hierarchical to:	No other components
Dependencies:	FPT_STM.1 Reliable time stamps
FCM_ASM_EXT.3.1	The TSF shall be able to check and record the configuration of patches on specific servers.

5.1.7 Automated server management (FCM_ASM_EXT.4)

Management Activity: The following actions should be considered for FMT:

- Compliance jobs can be created, configured and scheduled to run periodically.

Audit Activity: The following actions should be auditable (FAU):

- Execution of Compliance jobs and the success/failure of each execution on each target server.

FCM_ASM_EXT.4	**Compliance job**
Hierarchical to:	No other components
Dependencies:	No dependencies.
FCM_ASM_EXT.4.1	The TSF shall be able to determine whether configuration elements of a server adhere to, or violate administrative user-defined rules.

5.1.8 Deployment of files (FCM_ASM_EXT.5)

Management Activity: The following actions should be considered for FMT:

- File Deploy jobs can be created, configured and scheduled to run periodically, or on ad hoc basis (e.g., in response to a compliance rule violation, the user may remediate the situation by running a File Deploy job).

Audit Activity: The following actions should be auditable (FAU):

- Execution of File Deploy jobs and the success/failure of each execution on each target server.

FCM_ASM_EXT.5	**Deployment of files**
Hierarchical to:	No other components
Dependencies:	No dependencies.
FCM_JOB_EXT.5.1	The TSF shall be able to deploy (or push) multiple files and directories to one or more managed servers.

5.1.9 Deployment of content (FCM_ASM_EXT.6)

Management Activity: The following actions should be considered for FMT:

- Software Deploy jobs can be created, configured and scheduled to run periodically, or on ad hoc basis (e.g., in response to a compliance rule violation, the user may remediate the situation by running a Software Deploy job).

Audit Activity: The following actions should be auditable (FAU):

- Execution of Software Deploy jobs and the success/failure of each execution on each target server.

FCM_JOB_EXT.6	**Deployment of content**
Hierarchical to:	No other components
Dependencies:	No dependencies.
FCM_ASM_EXT.6.1	The TSF shall be able to execute deployed (or pushed) content to one or more managed servers unattended.

5.1.10 Network shell script (FCM_ASM_EXT.7)

Management Activity: The following actions should be considered for FMT:

- NSH Script jobs can be created, configured and scheduled to run periodically, or on ad hoc basis (e.g., in response to a compliance rule violation, the user may remediate the situation by running an NSH Script job).

Audit Activity: The following actions should be auditable (FAU):

- Execution of NSH Script jobs and the success/failure of each execution on each target server.

FCM_ASM_EXT.7	**Network shell script**
Hierarchical to:	No other components
Dependencies:	No dependencies.
FCM_ASM_EXT.7.1	The TSF shall be able to allow for the deployment and execution of previously saved network shell scripts.

5.1.11 Batch Jobs (FCM_ASM_EXT.8)

Management Activity: The following actions should be considered for FMT:

- Batch jobs can be created, configured and scheduled to run periodically, or on ad hoc basis (e.g., in response to a compliance rule violation, the user may remediate the situation by running a Batch job that in turn contains a series of jobs).

Audit Activity: The following actions should be auditable (FAU):

- Execution of Batch jobs and the success/failure of each execution on each target server.

FCM_ASM_EXT.8	**Batch Jobs**
Hierarchical to:	No other components
Dependencies:	No dependencies.
FCM_ASM_EXT.8.1	The TSF shall be able to concatenate a series of deploy jobs, file deploy jobs, and network shell script jobs.

5.1.12 Snapshot review (FCM_SMR_EXT.1)

Management Activity: The following actions should be considered for FMT:

- Configuring access to Snapshot jobs (and therefore job results) using a Role-based access control mechanism.

Audit Activity: The following actions should be auditable (FAU):

- Granting / revocation of access to job results.

FCM_SMR_EXT.1	**Snapshot review**
Hierarchical to:	No other components
Dependencies:	FCM_ASM_EXT.1 Snapshot reporting
FCM_SMR_EXT.1.1	The TSF shall provide [assignment: *authorized administrators*] with the capability to read information collected in the snapshot reports.
FCM_SMR_EXT.1.2	The TSF shall provide the snapshot report in a manner suitable for the user to interpret the information.

5.1.13 Audit job review (FCM_SMR_EXT.2)

Family Behavior: This family defines the requirements for tools that should be available to authorized users to assist in the review of Audit jobs.

Management Activity: The following actions should be considered for FMT:

- Configuring access to Audit jobs (and therefore job results) using a Role-based access control mechanism.

Audit Activity: The following actions should be auditable (FAU):

- Granting / revocation of access to job results.

FCM_SMR_EXT.2	**Audit job review**
Hierarchical to:	No other components
Dependencies:	FCM_ASM_EXT.2 Audit job
FCM_SMR_EXT.2.1	The TSF shall provide [assignment: *authorized administrators*] with the capability to read all information collected from the results of the Audit job.
FCM_SMR_EXT.2.2	The TSF shall provide the results of the Audit job in a manner suitable for the user to interpret the information.

5.1.14 Patch analysis job review (FCM_SMR_EXT.3)

Management Activity: The following actions should be considered for FMT:

- Configuring access to patch analysis jobs (and therefore job results) using a Role-based access control mechanism.

Audit Activity: The following actions should be auditable (FAU):

- Granting / revocation of access to job results.

FCM_SMR_EXT.3	**Patch analysis job review**
Hierarchical to:	No other components
Dependencies:	FCM_ASM_EXT.3 Patch analysis job
FCM_SMR_EXT.3.1	The TSF shall provide [assignment: *authorized administrators*] with the capability to read all information collected from the patch analysis jobs.
FCM_SMR_EXT.3.2	The TSF shall provide the patch analysis records in a manner suitable for the user to interpret the information.

5.1.15 Compliance reports (FCM_CRP_EXT.1)

Management Activity: The following actions should be considered for FMT:

- There are no management activities foreseen.

Audit Activity: The following actions should be auditable (FAU):

- There are no auditable events foreseen.

FCM_CRP_EXT.1	**Compliance Reports**
Hierarchical to:	No other components

Dependencies:	No dependencies.
FCM_CRP_EXT.1.1	The TSF shall provide [assignment: *authorized administrators*] with the capability to create compliance reports.
FCM_CRP_EXT.1.2	The TSF shall provide the report in a manner suitable for the user to interpret the information.

5.2 Rationale for the Extended TOE Security Functional Components

FCM_ASM_EXT.1 through FCM_ASM_EXT.8 were created to capture the basic functionality provide by the TOE, specifically the functionality associated with creating snapshot jobs, audit jobs, patch analysis jobs, file deploy jobs, software deploy, NHS script jobs, and batch jobs. FCM_SMR_EXT.1 through FCM_SMR_EXT.3 were created to capture the functionality to review the data produced by the compliance activities. FCM_CRP_EXT.1 was created to capture the compliance reporting capabilities of the TOE.

5.3 Extended TOE Security Assurance Components

There are no extended TOE security assurance components defined for this evaluation.

6 SECURITY REQUIREMENTS

6.1 Security Functional Requirements

The table provided below is a summary of the operations performed on the security functional requirements selected for the TOE. The operations will be identified as follows: A = Assignment, S = Selection, R = Refinement and I = Iteration.

Table 11 – TOE security functional requirements

Class	Functional component	A	S	R	I
Security Audit (FAU)	**FAU_GEN.1 (1)** Audit data generation (Appserver/RSCD logs)	X	X	X	X
	FAU_GEN.2 User identity association (Appserver/RSCD logs)			X	
	FAU_GEN.1 (2) Audit data generation (Audit Trail)	X	X	X	X
	FAU_SAR.1 Audit review (Audit Trail)	X		X	
	FAU_SAR.2 Restricted audit review (Audit Trail)			X	
Cryptographic Support (FCS)	**FCS_CKM.1 (1)** Cryptographic key generation	X			X
	FCS_CKM.1 (2) Cryptographic key generation	X			X
	FCS_CKM.4 Cryptographic key destruction	X			
	FCS_COP.1 (1) Cryptographic operation	X			X
	FCS_COP.1 (2) Cryptographic operation	X			X
	FCS_COP.1 (3) Cryptographic operation	X		X	X
	FCS_COP.1 (4) Cryptographic operation	X		X	X
User Data Protection (FDP)	**FDP_ACC.1** Subset access control	X			
	FDP_ACF.1 Security attribute based access control	X			
Identification and Authentication (FIA)	**FIA_AFL.1** Authentication failure handling	X	X		
	FIA_ATD.1 User attribute definition	X			
	FIA_SOS.1 Verification of secrets	X			
	FIA_UAU.2 User authentication before any action				
	FIA_UAU.7 Protected authentication feedback	X			
	FIA_UID.2 User identification before any action				
Security Management (FMT)	**FMT_MSA.1** Management of security attributes	X	X		
	FMT_MSA.3 Static attribute initialization	X	X		
	FMT_MTD.1 Management of TSF data	X	X		
	FMT_SMF.1 Specification of management functions	X			
	FMT_SMR.1 Security roles	X			
Protection of the TSF (FPT)	**FPT_ITT.1** Basic internal TSF data transfer protection		X	X	
Trusted Path/Channels (FTP)	**FTP_TRP.1** Trusted path	X	X		
Compliance Management	**FCM_ASM_EXT.1** Snapshot reporting	X			
	FCM_ASM_EXT.2 Audit job	X			
	FCM_ASM_EXT.3 Patch analysis job				

Class	Functional component	A	S	R	I
	FCM_ASM_EXT.4 Compliance job				
	FCM_ASM_EXT.5 Deployment of files				
	FCM_ASM_EXT.6 Deployment of content				
	FCM_ASM_EXT.7 Network shell script				
	FCM_ASM_EXT.8 Batch jobs				
	FCM_SMR_EXT.1 Snapshot review	X			
	FCM_SMR_EXT.2 Audit job review	X			
	FCM_SMR_EXT.3 Patch analysis job review	X			
	FCM_CRP_EXT.1 Compliance Reports	X			

6.1.1 Security Audit (FAU)

FAU_GEN.1(1)	**Audit data generation (Appserver/RSCD logs)**
Hierarchical to:	No other components.
Dependencies:	FPT_STM.1 Reliable time stamps
FAU_GEN.1.1(1)	The TSF shall be able to generate an audit record **in the Appserver/RSCD logs** of the following auditable events: a) Start-up and shutdown of the audit functions; b) All auditable events for the *not specified* level of audit; and c) [login/logout, and all functions defined in Table 12 below].
FAU_GEN.1.2(1)	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 (if applicable), and the outcome (success or failure) of the event; and b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [no other information].

Table 12 – Auditable Events

Action	Target
Create, Modify, Delete, Assign Role, Unassign Role	Users
Create, Delete, Modify, Add Authorization, Delete Authorization	Roles

FAU_GEN.2	**User identity association (Appserver/RSCD logs)**
Hierarchical to:	No other components.
Dependencies:	FAU_GEN.1 Audit data generation FIA_UID.1 Timing of identification
FAU_GEN.2.1	For audit events **in the Appserver/RSCD logs** resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

FAU_GEN.1(2)	**Audit data generation (Audit Trail)**
Hierarchical to:	No other components.

Dependencies:	FPT_STM.1 Reliable time stamps
FAU_GEN.1.1(2)	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 specified* level of audit; and c) [object access].
FAU_GEN.1.2(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 (if applicable)~~ **Role trying to access the object, User who has assumed a role in the current session**, **Authorization requested** and the outcome (success or failure) of the event; and b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [none].

Application Note: For the purpose of the audit trail, 'type of event' is always object access.

FAU_SAR.1	**Audit review (Audit Trail)**
Hierarchical to:	No other components.
Dependencies:	FAU_GEN.1 Audit data generation
FAU_SAR.1.1	The TSF shall provide [users in the RBACAdmins, and BLAdmins roles] with the capability to read [Date when a user requests authorization to access the object, Role trying to access the object, User who has assumed a role in the current session, Authorization requested and Status (success or failure)] from the audit **trail** records.
FAU_SAR.1.2	The TSF shall provide the audit records in a manner suitable for the user to interpret the information.

Application Note: Note that the term 'users in the RBACAdmins and BLAdmins roles' also includes users in other roles that may be created with the required permissions to perform this task.

FAU_SAR.2	**Restricted audit review (Audit Trail)**
Hierarchical to:	No other components.
Dependencies:	FAU_SAR.1 Audit review
FAU_SAR.2.1	The TSF shall prohibit all users read access to the audit **trail** records, except those users that have been granted explicit read-access.

6.1.2 Cryptographic Support (FCS)

FCS_CKM.1(1)	**Cryptographic key generation**
Hierarchical to:	No other components.
Dependencies:	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction
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].

FCS_CKM.1(2)	**Cryptographic key generation**
Hierarchical to:	No other components.
Dependencies:	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction
FCS_CKM.1.1(2)	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [RNG] and specified cryptographic key sizes [128 bits] that meet the following: [ANSI X.9.31 and FIPS 186-2].

Application Note: The OpenSSL FIPS Object Modules uses a Random Number Generator (RNG) that adheres to ANSI X.9.31 standards. The RSA BSAFE® Crypto-J JCE Provider Module uses an RNG that adheres to FIPS 186-2 standards.

FCS_CKM.4	**Cryptographic key destruction**
Hierarchical to:	No other components.
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]
FCS_CKM.4.1	The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [zeroization] that meets the following: [FIPS 140-2].

FCS_COP.1 (1)	**Cryptographic operation**
Hierarchical to:	No other components.
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] FCS_CKM.4 Cryptographic key destruction
FCS_COP.1.1 (1)	The TSF shall perform [encryption and decryption] in accordance with a specified cryptographic algorithm [AES] and cryptographic key sizes [128 bits] that meet the following: [FIPS197].

FCS_COP.1 (2)	**Cryptographic operation**
Hierarchical to:	No other components.
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] FCS_CKM.4 Cryptographic key destruction
FCS_COP.1.1(2)	The TSF shall perform [cryptographic signature services] in accordance with a specified cryptographic algorithm [RSA] and cryptographic key sizes [1024 bits] that meet the following: [FIPS186-2].

FCS_COP.1 (3)	**Cryptographic operation**
Hierarchical to:	No other components.
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] FCS_CKM.4 Cryptographic key destruction
FCS_COP.1.1 (3)	The TSF shall perform [cryptographic hashing services] in accordance with a specified cryptographic algorithm [SHA-1] and ~~cryptographic key~~ **message digest** sizes [160 bits] that meet the following: [FIPS180-3].

FCS_COP.1 (4)	**Cryptographic operation**
Hierarchical to:	No other components.
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] FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1 (4)	The TSF shall perform [keyed-hash message authentication] in accordance with a specified cryptographic algorithm [HMAC-SHA-1]**,** ~~and cryptographic~~ key sizes [160 bits]**, and message digest sizes [160 bits]** that meet the following: [FIPS198-1 and FIPS180-3].

6.1.3 User Data Protection (FDP)

FDP_ACC.1	**Subset access control**
Hierarchical to:	No other components.
Dependencies:	FDP_ACF.1 Security attribute based access control
FDP_ACC.1.1	The TSF shall enforce the [BSA Role Based Access Control Policy] on [subjects: administrative users in the roles BLAdmins, RBACAdmins objects: system objects operations: system functions].

Application Notes: Note that 'administrative users in the roles RBACAdmins and BLAdmins ' also includes users in other roles that may be created with the required permissions to perform these tasks. A system object is any item within BSA that may be accessed. It includes all functions, such as batch job management and discovery job management functions, and all physical resources such as servers. System functions include all operations that may be appropriate for that object, such as read, write, modify, cancel, delete, modify priority, pause, resume, execute, open, etc.

FDP_ACF.1	**Security attribute based access control**
Hierarchical to:	No other components.
Dependencies:	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation
FDP_ACF.1.1	The TSF shall enforce the [BSA Role Based Access Control Policy] to objects based on the following: [Subjects: administrative users subject attributes: role objects: system objects object attributes: ACL].
FDP_ACF.1.2	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [an authenticated user may access a system object if the user's current role has been given authorization to access the object, and the object has an ACL which grants the user's current role access to the object].
FDP_ACF.1.3	The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [users in the RBACAdmins role may access all objects].
FDP_ACF.1.4	The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [none].

Application Notes: Note that 'authorization' to access an object is based upon having the required 'permission' to perform the task requiring access.

6.1.4 Identification and Authentication (FIA)

FIA_AFL.1	**Authentication failure handling**
Hierarchical to:	No other components.
Dependencies:	FIA_UAU.1 Timing of authentication
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].

FIA_ATD.1	**User attribute definition**
Hierarchical to:	No other components.

Dependencies:	No dependencies.
FIA_ATD.1.1	The TSF shall maintain the following list of security attributes belonging to individual users: [user identity, authentication data, and authorized role].

FIA_SOS.1	**Verification of secrets**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FIA_SOS.1.1	The TSF shall provide a mechanism to verify that secrets meet [an administrator defined minimum length, maximum password age, may not contain the user's account name and must contain characters from three of the following categories: uppercase letters, lowercase letters, digits, non-alphanumeric characters, other Unicode characters].

FIA_UAU.2	**User authentication before any action**
Hierarchical to:	FIA_UAU.1 Timing of authentication
Dependencies:	FIA_UID.1 Timing of identification
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.

FIA_UAU.7	**Protected authentication feedback**
Hierarchical to:	No other components.
Dependencies:	FIA_UAU.1 Timing of authentication
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.

FIA_UID.2	**User identification before any action**
Hierarchical to:	FIA_UID.1 Timing of identification
Dependencies:	No dependencies.
FIA_UID.2.1	The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

6.1.5 Security Management (FMT)

FMT_MSA.1	**Management of security attributes**
Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions
FMT_MSA.1.1	The TSF shall enforce the [BSA Role Based Access Control Policy] to restrict the ability to *change_default, query, modify, delete* the security attributes [role permissions and users assigned to roles] to [users in the RBACAdmins role].

Application Note: Note that the term 'users in the RBACAdmins role' also includes users in other roles that may be created with the required permissions to perform this task.

FMT_MSA.3	Static attribute initialisation
Hierarchical to:	No other components.
Dependencies:	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles
FMT_MSA.3.1	The TSF shall enforce the [BSA 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 in the RBACAdmins role] to specify alternative initial values to override the default values when an object or information is created.

Application Note: Note that the term 'users in the RBACAdmins role' also includes users in other roles that may be created with the required permissions to perform this task.

FMT_MTD.1	Management of TSF data
Hierarchical to:	No other components.
Dependencies:	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions
FMT_MTD.1.1	The TSF shall restrict the ability to *delete, [create, read, modify properties of]* the [object level permissions] to [users in the RBACAdmins and BLAdmins roles].

Application Note: Note that the term 'users in the RBACAdmins and BLAdmins roles' also includes users in other roles that may be created with the required permissions to perform this task.

FMT_SMF.1	Specification of Management Functions
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FMT_SMF.1.1	The TSF shall be capable of performing the following management functions: [as specified in Table 13].

Table 13 – System Management Functions

SFR	Management Function
FDP_ACF.1	Management of the object level ACLs and RBAC permissions used to make explicit access or denial based decisions.
FIA_AFL.1	Management of the threshold for unsuccessful authentication attempts.
	Management of actions to be taken in the event of an authentication failure.
FIA_SOS.1	Management of the password policy.
FIA_UAU.2, FIA_UID.2	Management of user accounts, including user identity and authentication type.
FMT_MSA.1	Management of the roles that can access user and role attributes.
FMT_SMR.1	Management of the group of users that are part of a role.
FCM_ASM_EXT.1	Management of snapshot jobs, such that they may be created, configured and scheduled to run periodically.
FCM_ASM_EXT.2	Management of audit jobs, such that they can be created, configured and scheduled to run periodically.
FCM_ASM_EXT.3	Management of patch analysis jobs, such that they can be created, configured and scheduled to run periodically.
FCM_ASM_EXT.4	Management of compliance jobs, such that they can be created, configured and scheduled to run periodically.
FCM_ASM_EXT.5	Management of file deploy jobs, such that they can be created, configured and scheduled to run periodically, or on ad hoc basis (e.g., in response to a compliance rule violation, the user may remediate the situation by running a File Deploy job).
FCM_ASM_EXT.6	Management of software deploy jobs, such that they can be created, configured and scheduled to run periodically, or on ad hoc basis (e.g., in response to a compliance rule violation, the user may remediate the situation by

	running a software deploy job).
FCM_ASM_EXT.7	Management of NSH Script jobs, such that they can be created, configured and scheduled to run periodically, or on ad hoc basis (e.g., in response to a compliance rule violation, the user may remediate the situation by running an NSH Script job).
FCM_ASM_EXT.8	Management of batch jobs, such that they can be created, configured and scheduled to run periodically, or on ad hoc basis (e.g., in response to a compliance rule violation, the user may remediate the situation by running a batch job that in turn contains a series of jobs).
FCM_SMR_EXT.1	Management of access to snapshot jobs (and therefore job results) using a Role-based access control mechanism.
FCM_SMR_EXT.2	Management of access to audit jobs (and therefore job results) using a Role-based access control mechanism.
FCM_SMR_EXT.3	Management of access to patch analysis jobs (and therefore job results) using a Role-based access control mechanism.

FMT_SMR.1	**Security roles**
Hierarchical to:	No other components.
Dependencies:	FIA_UID.1 Timing of identification
FMT_SMR.1.1	The TSF shall maintain the roles: [RBACAdmins, BLAdmins, additional roles as configured by a user in the RBACAdmins role].
FMT_SMR.1.2	The TSF shall be able to associate users with roles.

6.1.6 Protection of the TSF (FPT)

FPT_ITT.1	**Basic internal TSF data transfer protection**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_ITT.1.1	The TSF shall protect TSF data from *disclosure* **and** *modification* when it is transmitted between separate parts of the TOE.

6.1.7 Trusted Path/Channels (FTP)

FTP_TRP.1	**Trusted path**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
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 *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, [reports management]]*.

6.1.8 Compliance management (FCM)

FCM_ASM_EXT.1	**Snapshot reporting**
Hierarchical to:	No other components
Dependencies:	FPT_STM.1 Reliable time stamps
FCM_ASM_EXT.1.1	The TSF shall be able to generate a snapshot report based on server objects.

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

Application Note: This requirement implements the concept of a snapshot job described in Section 7.9.1. The 'Object' is the server being assessed.

FCM_ASM_EXT.2	**Audit job**
Hierarchical to:	No other components
Dependencies:	FPT_STM.1 Reliable time stamps
FCM_ASM_EXT.2.1	The TSF shall be able to compare snapshots to determine whether servers match a standard configuration.
FCM_ASM_EXT.2.2	The TSF shall be able to generate an '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].
FCM_ASM_EXT.2.3	The TSF shall record within each entry of the 'Audit 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].

Application Note: This requirement implements the functionality described for an audit job as defined in Section 7.9.2, and an audit report as defined in Section 7.9.3.

FCM_ASM_EXT.3	**Patch analysis job**
Hierarchical to:	No other components
Dependencies:	FPT_STM.1 Reliable time stamps
FCM_ASM_EXT.3.1	The TSF shall be able to check and record the configuration of patches on specific servers.

Application Note: This requirement implements the patch analysis job as defined in Section 7.9.4.

FCM_ASM_EXT.4	**Compliance job**
Hierarchical to:	No other components
Dependencies:	No dependencies.
FCM_ASM_EXT.4.1	The TSF shall be able to determine whether configuration elements of a server adhere to, or violate administrative user-defined rules.

Application Note: This requirement is used to capture the functionality of Compliance Job described in Section 7.9.5.

FCM_ASM_EXT.5	**Deployment of files**
Hierarchical to:	No other components
Dependencies:	No dependencies.

FCM_ASM_EXT.5.1	The TSF shall be able to deploy (or push) multiple files and directories to one or more managed servers.

Application Note: This requirement is used to capture the File Deploy functionality described in Section 7.9.6.

FCM_ASM_EXP.6	**Deployment of content**
Hierarchical to:	No other components
Dependencies:	No dependencies.
FCM_ASM_EXT.6.1	The TSF shall be able to execute deployed (or pushed) content to one or more managed servers unattended.

Application Note: This requirement is used to capture the Deploy Jobs functionality described in Section 7.9.7.

FCM_ASM_EXT.7	**Network shell script**
Hierarchical to:	No other components
Dependencies:	No dependencies.
FCM_ASM_EXT.7.1	The TSF shall be able to allow for the deployment and execution of previously saved network shell scripts.

Application Note: This requirement is used to capture the Network Shell Jobs functionality described in Section 7.9.9.

FCM_ASM_EXT.8	**Batch Jobs**
Hierarchical to:	No other components
Dependencies:	No dependencies.
FCM_ASM_EXT.8.1	The TSF shall be able to concatenate a series of deploy jobs, file deploy jobs, and network shell script jobs.

Application Note: This requirement is used to capture the functionality of Batch Jobs described in Section 7.9.10.

FCM_SMR_EXT.1	**Snapshot review**
Hierarchical to:	No other components
Dependencies:	FCM_ASM_EXT.1 Snapshot reporting
FCM_SMR_EXT.1.1	The TSF shall provide [users in the RBACAdmins, BLAdmins roles] with the capability to read information collected in the snapshot reports.
FCM_SMR_EXT.1.2	The TSF shall provide the snapshot report in a manner suitable for the user to interpret the information.

Application Note: This requirement the snapshot job as defined in Section 7.9.1. Note that the term ‘users in the RBACAdmins, BLAdmins roles’ also includes users in other roles that may be created with the required permissions to perform this task.

FCM_SMR_EXT.2	**Audit job review**

Hierarchical to:	No other components
Dependencies:	FCM_ASM_EXT.2 Audit job
FCM_SMR_EXT.2.1	The TSF shall provide [users in the RBACAdmins, BLAdmins roles] with the capability to read all information collected from the results of the Audit job.
FCM_SMR_EXT.2.2	The TSF shall provide the results of the Audit job in a manner suitable for the user to interpret the information.

Application Note: This requirement implements the audit job report described in Section 7.9.3. Note that the term 'users in the RBACAdmins, BLAdmins roles' also includes users in other roles that may be created with the required permissions to perform this task.

FCM_SMR_EXT.3	**Patch analysis job review**
Hierarchical to:	No other components
Dependencies:	FCM_ASM_EXT.3 Patch analysis job
FCM_SMR_EXT.3.1	The TSF shall provide [users in the RBACAdmins, BLAdmins roles] with the capability to read all information collected from the patch analysis jobs.
FCM_SMR_EXT.3.2	The TSF shall provide the patch analysis records in a manner suitable for the user to interpret the information.

Application Note: This requirement implements the patch analysis job functionality described in Section 7.9.4. Note that the term 'users in the RBACAdmins, BLAdmins roles' also includes users in other roles that may be created with the required permissions to perform this task.

FCM_CRP_EXT.1	**Compliance Reports**
Hierarchical to:	No other components
Dependencies:	No dependencies.
FCM_CRP_EXT.1.1	The TSF shall provide [users in the RBACAdmins, BLAdmins roles] with the capability to create compliance reports.
FCM_CRP_EXT.1.2	The TSF shall provide the report in a manner suitable for the user to interpret the information.

Application Note: This requirement is used to capture the functionality of the BDSSA Reports Server, described in Section 7.9.11. Note that the terms 'users in the RBACAdmins, BLAdmins roles' also includes users in other roles that may be created with the required permissions to perform this task.

6.2 Security Assurance Requirements (SARs)

The TOE satisfies the SARs delineated in Table 20.

Table 14 – TOE security assurance requirements

Class	Assurance Component
Development (ADV)	**ADV_ARC.1** Security architecture description
	ADV_FSP.2 Security-enforcing functional specification
	ADV_TDS.1 Basic design
Guidance Documents (AGD)	**AGD_OPE.1** Operational user guidance
	AGD_PRE.1 Preparative procedures
Life-Cycle Support (ALC)	**ALC_CMC.2** Use of a CM system
	ALC_CMS.2 Parts of the TOE CM coverage
	ALC_DEL.1 Delivery procedures
	ALC_FLR.2 Flaw reporting procedures
Security Target Evaluation (ASE)	**ASE_CCL.1** Conformance claims
	ASE_ECD.1 Extended components definition
	ASE_INT.1 ST introduction
	ASE_OBJ.2 Security objectives
	ASE_REQ.2 Derived security requirements
	ASE_SPD.1 Security problem definition
	ASE_TSS.1 TOE summary specification
Tests (ATE)	**ATE_COV.1** Evidence of coverage
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent testing – sample
Vulnerability Assessment (AVA)	**AVA_VAN.2** Vulnerability analysis

6.3 Security Requirements Rationale

6.3.1 Security Functional Requirements Rationale

The following two tables provide the security requirement to security objective mapping and a rationale to justify the mapping.

Table 15 – Objective to requirement correspondence

	O.ACCESS	O.AUDIT	O.AUTH	O.COMPLIANCE	O.EAVESDROPPING	O.MANAGE	O.MONITOR	O.IMPERSONATE
FAU_GEN.1 (1) Audit data generation (Appserver/RSCD logs)		X						
FAU_GEN.2 User identity association (Appserver/RSCD logs)		X						
FAU_GEN.1 (2) Audit data generation (Audit Trail)		X						
FAU_SAR.1 Audit review (Audit Trail)		X						
FAU_SAR.2 Restricted audit review (Audit Trail)		X						
FCS_CKM.1 (1) Cryptographic key generation					X			
FCS_CKM.1 (2) Cryptographic key generation					X			
FCS_CKM.4 Cryptographic key destruction					X			
FCS_COP.1 (1) Cryptographic operation					X			
FCS_COP.1 (2) Cryptographic operation					X			
FCS_COP.1 (3) Cryptographic operation					X			
FCS_COP.1 (4) Cryptographic operation					X			
FDP_ACC.1 Subset access control	X							
FDP_ACF.1 Security attribute based access control	X							
FIA_AFL.1 Authentication failure handling			X					
FIA_ATD.1 User attribute definition	X		X					
FIA_SOS.1 Verification of secrets			X					
FIA_UAU.2 User authentication before any action			X					
FIA_UAU.7 Protected authentication feedback			X					
FIA_UID.2 User identification before any action			X					
FMT_MSA.1 Management of security attributes						X		
FMT_MSA.3 Static attribute initialization						X		
FMT_MTD.1 Management of TSF data						X		
FMT_SMF.1 Specification of management functions						X		
FMT_SMR.1 Security roles	X					X		
FPT_ITT.1 Basic internal TSF data transfer protection					X			
FTP_TRP.1 Trusted path					X			X
FCM_ASM_EXT.1 Snapshot reporting							X	
FCM_ASM_EXT.2 Audit job							X	
FCM_ASM_EXT.3 Patch analysis job							X	
FCM_ASM_EXT.4 Compliance job				X			X	
FCM_ASM_EXT.5 Deployment of files				X			X	

	O.ACCESS	O.AUDIT	O.AUTH	O.COMPLIANCE	O.EAVESDROPPING	O.MANAGE	O.MONITOR	O.IMPERSONATE
FCM_ASM_EXT.6 Deployment of content				X			X	
FCM_ASM_EXT.7 Network shell script				X			X	
FCM_ASM_EXT.8 Batch jobs				X			X	
FCM_SMR_EXT.1 Snapshot review							X	
FCM_SMR_EXT.2 Audit job review							X	
FCM_SMR_EXT.3 Patch analysis job review							X	
FCM_CRP_EXT.1 Compliance Reports							X	

Table 16 – Security functional requirements rationale for the TOE

Objective	SFR	Rationale
O.ACCESS	**FDP_ACC.1, FDP_ACF.1**	FDP_ACC.1 and FDP_ACF.1 meet the O.ACCESS security objective by controlling user access to specified resources based on the permissions associated with the user's current role, and the object level access controls found in the ACLs for the object being accessed. This enforces the BSA Role Based Access Control Policy.
	FIA_ATD.1	FIA_ATD.1 meets the O.ACCESS security objective by ensuring that the role information is maintained by the TOE for each user.
	FMT_SMR.1	FMT_SMR.1 supports O.ACCESS by providing the roles required to meet the BSA Role Based Access Control Policy.
O.AUDIT	**FAU_GEN.1(1)**	FAU_GEN.1(1) meets the O.AUDIT security objective by defining the security relevant events that will be recorded by the TOE along with the details of the event that will be recorded. The specific audit events ensure that users are not given permissions that would allow misuse of the TOE or its security features, or would compromise the integrity of the TOE and violate the security objectives.
	FAU_GEN.2	FAU_GEN.2 meets the O.AUDIT security objective by requiring that the TOE associate each auditable event with the identity of the user that caused the event, ensuring that misuse would be detected and attributable to the user responsible.
	FAU_GEN.1(2)	FAU_GEN.1(2) meets the O.AUDIT security objective by recording all attempts to access objects. The access events recorded ensure detection of permissions mistakenly granted that could allow misuse of the TOE or its security features, or would compromise the integrity of the TOE and violate the security objectives.
	FAU_SAR.1	FAU_SAR.1 meets the O.AUDIT security objective by allowing authorized TOE users the ability to view the audit trail records.
	FAU_SAR.2	FAU_SAR.2 meets the O.AUDIT security objective by ensuring that only the authorized users are able to view the audit trail records.
O.MONITOR	**FCM_ASM_EXT.1**	FCM_ASM_EXT.1 meets the O.MONITOR security objective by defining the server object data that will be recorded by the TOE for each snapshot job. This data may then be used to determine if the server in the operational environment is configured in accordance with the policies established for that server.
	FCM_ASM_EXT.2	FCM_ASM_EXT.2 meets the O.MONITOR security objective by defining the details to be listed in the 'Audit Report'. This report identifies whether or not servers are configured in accordance with the policies established for those servers.

Objective	SFR	Rationale
	FCM_ASM_EXT.3	FCM_ASM_EXT.3 meets the O.MONITOR security objective by specifying the data (hotfixes and patches) that will be recorded by the TOE in support of each patch analysis job. This is essential for ensuring that servers in the operational environment are patched to the appropriate level in accordance with the established policy.
	FCM_ASM_EXT.4	FCM_ASM_EXT.4 meets the O.MONITOR security objective by providing the capability to run a compliance job. This function provides the analysis that determines if the configuration of servers meets the established security policy requirements for those servers.
	FCM_ASM_EXT.5	FCM_ASM_EXT.5 meets the O.MONITOR security objective by providing the capability to push the files required for a deploy job to the server.
	FCM_ASM_EXT.6	FCM_ASM_EXT.6 meets the O.MONITOR security objective by providing the capability to force the pushed content to run, thereby bringing an uncompliant server into compliance with the established security policy.
	FCM_ASM_EXT.7	FCM_ASM_EXT.7 meets the O.MONITOR security objective by providing the capability to deploy and execute network shell scripts. These scripts may then be run to bring a server into compliance with the established security policy.
	FCM_ASM_EXT.8	FCM_ASM_EXT.8 meets the O.MONITOR security objective by providing the capability to combine deploy, file deploy and network shell script jobs into batch jobs. These batch jobs may then be run to bring a server into compliance with the established security policy.
	FCM_SMR_EXT.1	FCM_SMR_EXT.1 meets the O.MONITOR security objective by providing the capability to read information from the snapshot jobs. This data may then be used to determine if the server in the operational environment is configured in accordance with the policies established for that server.
	FCM_SMR_EXT.2	FCM_SMR_EXT.2 meets the O.MONITOR security objective by providing the capability to read information from the results of an Audit job. This report identifies whether or not servers are configured in accordance with the policies established for those servers, based on information captured in the 'Audit Report'.
	FCM_SMR_EXT.3	FCM_SMR_EXT.3 meets the O.MONITOR security objective by providing the capability to read information from the patch analysis job. This information is required to ensure that servers in the operational environment are patched to the appropriate level in accordance with the established security policy.
	FCM_CRP_EXT.1	FCM_CRP_EXT.1 meets the O.MONITOR security objective by providing a means of creating reports to present the results of data collected and analyzed from monitored servers.
O.AUTH	**FIA_ATD.1**	FIA_ATD.1 meets the O.AUTH security objective by specifying the security attributes that are maintained in support of user identification and authentication.
	FIA_UAU.2	FIA_UAU.2 meets the O.AUTH security objective by ensuring that the user is authenticated prior to being allowed access to the TOE and resources protected by the TOE.
	FIA_UAU.7	FIA_UAU.7 meets the O.AUTH security objective by ensuring that only limited feedback information is provided to the user during the authentication. By mitigating the risk of passwords being discovered, this helps to ensure that a user's identification and authentication information uniquely identifies that user.
	FIA_UID.2	FIA_UID.2 meets the O.AUTH security objective by ensuring that a user is identified before any access to the TOE and resources protected by the TOE is allowed.
	FIA_AFL.1	FIA_AFL.1 meets the O.AUTH security objective by mitigating the risk that unauthorized users will attempt a brute force attack on the unique identification and authentication information of a user.
	FIA_SOS.1	FIA_SOS.1 meets the O.AUTH security objective by ensuring that the strength of the authentication mechanism is sufficient to mitigate the risk that unauthorized users will be able to impersonate an authorized user.
O.MANAGE	**FMT_MSA.1**	FMT_MSA.1 meets the O.MANAGE security objective by restricting the management of role permissions and users assigned to roles (user account information) to users in the RBACAdmins role.

Objective	SFR	Rationale
	FMT_MSA.3	FMT_MSA.3 meets the O.MANAGE security objective by providing restrictive default values to the security attributes (security information) used to enforce the BSA Role Based Access Control Policy. FMT_MSA.3 further supports this objective by allowing the authorized administrators to override the default values set for security attributes when, for example, creating user accounts.
	FMT_MTD.1	FMT_MTD.1 meets the O.MANAGE security objective by providing authorized administrators with the ability to affect the object level permissions used to support the BSA Role Based Access Control Policy (TOE resources).
	FMT_SMF.1	FMT_SMF.1 meets the O.MANAGE security objective by providing the management functions required to manage the TOE security functionality reflected in the SFRs claimed in this ST.
	FMT_SMR.1	FMT_SMR.1 meets the O.MANAGE security objective by providing the security roles required by the Security management SFRs, and ensuring that users may be associated with these roles.
O.EAVESDROPPING	**FCS_CKM.1(1)**	FCS_CKM.1(1) meets the O.EAVESDROPPING security objective by providing RSA key generation in support of the encryption services provided by the TOE.
	FCS_CKM.1(2)	FCS_CKM.1(2) meets the O.EAVESDROPPING security objective by providing AES key generation in support of the encryption services provided by the TOE.
	FCS_CKM.4	FCS_CKM.4 meets the O.EAVESDROPPING security objective by providing destruction of encryption keys.
	FCS_COP.1 (1)	FCS_COP.1 (1) meets the O.EAVESDROPPING security objective by providing AES encryption and decryption in support of the encryption services provided by the TOE.
	FCS_COP.1 (2)	FCS_COP.1 (2) meets the O.EAVESDROPPING security objective by providing cryptographic signature services with RSA in support of the encryption services provided by the TOE.
	FCS_COP.1 (3)	FCS_COP.1 (3) meets the O.EAVESDROPPING security objective by providing cryptographic hashing services with SHA-1 in support of the encryption services provided by the TOE.
	FCS_COP.1 (4)	FCS_COP.1 (4) meets the O.EAVESDROPPING security objective by providing keyed-hash message authentication with HMAC-SHA-1 in support of the encryption services provided by the TOE.
	FTP_TRP.1	FPT_TRP.1 meets the O.EAVESDROPPING security objective by ensuring that communications between the TOE and remote users are encrypted to provide protection from disclosure.
	FPT_ITT.1	FPT_ITT.1 meets the O.EAVESDROPPING security objective by ensuring that communications between components of the TOE are encrypted to protect TSF data from disclosure and modification.
O.COMPLIANCE	**FCM_ASM_EXT.4**	FCM_ASM_EXT.4 meets the O.COMPLIANCE security objective by ensuring that the TOE has the ability to detect any deviations between a server's configuration and the configuration required by the established security policy for files, registry and patch elements.
	FCM_ASM_EXT.5	FCM_ASM_EXT.5 meets the O.COMPLIANCE security objective by ensuring that the TOE has the ability to deploy (or push) multiple files and directories to one or more managed servers, in support of compliance with local security policies for files.
	FCM_ASM_EXT.6	FCM_ASM_EXT.6 meets the O.COMPLIANCE security objective by ensuring that the TOE has the ability to execute deployed (or pushed) content to one or more managed servers unattended. This ensures maintenance of compliance with local security policies.
	FCM_ASM_EXT.7	FCM_ASM_EXT.7 meets the O.COMPLIANCE security objective by ensuring that the TOE has the ability to deploy and execute previously saved network shell scripts. This functionality may be used to maintain compliance with local security policies for files, registry and patches.

Objective	SFR	Rationale
	FCM_ASM_EXT.8	FCM_ASM_EXT.8 meets the O.COMPLIANCE security objective by ensuring that the TOE has the ability to concatenate a series of deploy jobs, file deploy jobs, and network shell jobs. This functionality may be used to maintain compliance with local security policies for files, registry and patches.
O.IMPERSONATE	**FTP_TRP.1**	FPT_TRP.1 meets the O.IMPERSONATE security objective by providing assured identification of end points prior to allowing remote user access to the TSF.

6.3.2 Rationale for SFR Dependencies

Table 23 is a cross-reference of the functional components, their related dependencies, and whether the dependency was satisfied.

Table 17 – SFR dependency status

SFR	Dependencies	Fulfilled by SFRs in this ST
FAU_GEN.1(1)	FPT_STM.1	See the Note provided below
FAU_GEN.2	FAU_GEN.1	FAU_GEN.1(1)
	FIA_UID.1	FIA_UID.2
FAU_GEN.1(2)	FPT_STM.1	See the Note provided below
FAU_SAR.1	FAU_GEN.1	FAU_GEN.1(2)
FAU_SAR.2	FAU_SAR.1	FAU_SAR.1
FCS_CKM.1(1)	[FCS_CKM.2 or FCS_COP.1]	FCS_COP.1 (1), FCS_COP.1 (2), FCS_COP.1 (3) and FCS_COP.1 (4)
	FCS_CKM.4	FCS_CKM.4
FCS_CKM.1(2)	[FCS_CKM.2 or FCS_COP.1]	FCS_COP.1 (1), FCS_COP.1 (2), FCS_COP.1 (3) and FCS_COP.1 (4)
	FCS_CKM.4	FCS_CKM.4
FCS_CKM.4	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1(1) and FCS_CKM.1(2)
FCS_COP.1 (1)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1(1) and FCS_CKM.1(2)
	FCS_CKM.4	FCS_CKM.4
FCS_COP.1 (2)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1(1) and FCS_CKM.1(2)
	FCS_CKM.4	FCS_CKM.4
FCS_COP.1 (3)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1(1) and FCS_CKM.1(2)
	FCS_CKM.4	FCS_CKM.4
FCS_COP.1 (4)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1(1) and FCS_CKM.1(2)
	FCS_CKM.4	FCS_CKM.4
FDP_ACC.1	FDP_ACF.1	FDP_ACF.1
FDP_ACF.1	FDP_ACC.1	FDP_ACC.1
	FMT_MSA.3	FMT_MSA.3
FIA_AFL.1	FIA_UAU.1	FIA_UAU.2
FIA_ATD.1	None	Not applicable
FIA_SOS.1	None	Not applicable
FIA_UAU.2	FIA_UID.1	FIA_UID.2
FIA_UAU.7	FIA_UAU.1	FIA_UAU.2
FIA_UID.2	None	Not applicable
FMT_MSA.1	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1
	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1

SFR	Dependencies	Fulfilled by SFRs in this ST
FMT_MSA.3	FMT_MSA.1	FMT_MSA.1
	FMT_SMR.1	FMT_SMR.1
FMT_MTD.1	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_SMF.1	None	Not applicable
FMT_SMR.1	FIA_UID.1	FIA_UID.2
FPT_ITT.1	None	Not applicable
FTP_TRP.1	None	Not applicable
FCM_ASM_EXT.1	FPT_STM.1	See note below
FCM_ASM_EXT.2	FPT_STM.1	See note below
FCM_ASM_EXT.3	FPT_STM.1	See note below
FCM_SMR_EXT.1	FCM_ASM_EXT.1	FCM_ASM_EXT.1
FCM_SMR_EXT.2	FCM_ASM_EXT.2	FCM_ASM_EXT.2
FCM_SMR_EXT.3	FCM_ASM_EXT.3	FCM_ASM_EXT.3
FCM_ASM_EXT.4	None	Not applicable
FCM_ASM_EXT.5	None	Not applicable
FCM_ASM_EXT.6	None	Not applicable
FCM_ASM_EXT.7	None	Not applicable
FCM_ASM_EXT.8	None	Not applicable
FCM_CRP_EXT.1	None	Not applicable

Note: In accordance with OE.TIMESTAMP, the IT environment will provide an accurate time source that will be available to the TOE for use in determining the timestamp for the audit trail.

6.3.3 Security Assurance Requirements Rationale

EAL2+ was chosen to provide a low to moderate level of assurance that is consistent with good commercial practices. As such, minimal additional tasks are placed upon the vendor assuming the vendor follows reasonable practices and can provide support to the evaluation for design and testing efforts. The chosen assurance level is appropriate with the threats defined for the environment. The TOE is expected to be in a non-hostile location and protected by other products designed to address threats that correspond with the intended environment. ALC_FLR.2 was added to provide assurance that the vendor will respond to reports of new threats and reported defects in the TOE.

7 TOE SUMMARY SPECIFICATION

7.1 Mapping of the TSFs to SFRs

The specified TSFs work together to satisfy the TOE SFRs. Table 18 provides a mapping of SFRs to the TSFs to show that each SFR is captured within a security function.

Table 18 – Mapping of TSFs to SFRs

TSF	SFR
Security Audit	**FAU_GEN.1 (1)** Audit data generation (Appserver/RSCD logs)
	FAU_GEN.2 User identity association (Appserver/RSCD logs)
	FAU_GEN.1 (1) Audit data generation (Audit Trail)
	FAU_SAR.1 Audit review (Audit Trail)
	FAU_SAR.2 Restricted audit review (Audit Trail)
Cryptographic Support	**FCS_CKM.1 (1)** Cryptographic key generation
	FCS_CKM.1 (2) Cryptographic key generation
	FCS_CKM.4 Cryptographic key destruction
	FCS_COP.1 (1) Cryptographic operation
	FCS_COP.1 (2) Cryptographic operation
	FCS_COP.1 (3) Cryptographic operation
	FCS_COP.1 (4) Cryptographic operation
Access Control	**FDP_ACC.1** Subset access control
	FDP_ACF.1 Security attribute based access control
Identification and Authentication	**FIA_AFL.1** Authentication failure handling
	FIA_ATD.1 User attribute definition
	FIA_SOS.1 Verification of Secret
	FIA_UAU.2 User authentication before any action
	FIA_UAU.7 Protected authentication feedback
	FIA_UID.2 User identification before any action
Security Management	**FMT_MSA.1** Management of security attributes
	FMT_MSA.3 Static attribute initialization
	FMT_MTD.1 Management of TSF data
	FMT_SMF.1 Specification of management functions
	FMT_SMR.1 Security roles
Protection of the TSF	**FPT_ITT.1** Basic internal TSF data transfer protection
Trusted Path	**FTP_TRP.1** Trusted Path
Compliance Management	**FCM_ASM_EXT.1** Snapshot reporting
	FCM_ASM_EXT.2 Audit job
	FCM_ASM_EXT.3 Patch analysis job
	FCM_SMR_EXT.1 Snapshot review
	FCM_SMR_EXT.2 Audit job review
	FCM_SMR_EXT.3 Patch analysis job review
	FCM_ASM_EXT.4 Compliance job
	FCM_ASM_EXT.5 Deployment of files
	FCM_ASM_EXT.6 Deployment of content

TSF	SFR
	FCM_ASM_EXT.7 Network shell script
	FCM_ASM_EXT.8 Batch jobs
	FCM_CRP_EXT.1 Compliance reports

7.2 Security Audit

BSA provides multiple methods for recording activities that take place within the TOE. Claims are made with respect to the Appserver/RSCD logs and the Audit Trail.

7.2.1 Appserver/RSCD Logs

The claims made in FAU_GEN.1(1) and FAU_GEN.2 reference the functionality provided by the Appserver and RSCD logs. BSA uses log4j to capture these log messages. Log4j is an open source logging framework used to control logging output from Java applications.

System activity is recorded on the Application Server, Console, BLCLI and on RSCD Agents. Much of the activity related to the auditing of security events may be found in the BSA Application Server log. By default, the appserver.log logs at the info level, but may be changed using the log4crc.txt file. This file also contains the instructions for rolling over log files. During initial configuration, the files may be set to rollover when they reach a particular size (rollsize) or after a predetermined interval (rolltimeinsec). The rollmaxfiles option is used to specify the maximum number of files kept before they are overwritten. The layout option must include layout=dated to ensure that a timestamp precedes each log entry.

Log entries contain the timestamp, the type of event, including success or failure, and the role and username of the user in that role, where applicable.

The following table provides a list of logs that contain security relevant entries.

Table 19 – Log files and locations

Log File Name	Description and Default Location
Application Server	
appserver.log	Application Server log installDirectory/br/appserver.log
AppServerLauncher.log	AppServer Launcher log installDirectory/br/ AppServerLauncher.log
post_install.log	Application Server configuration log installDirectory/br/post_install.log
Console.log Application	Server Console log installDirectory/br/Console.log
RSCD Agent (Windows)	
rscd.log*	Windows RSCD Agent log rscdInstallDirectory\rscd.log
rscdsvc.log*	RSCD Agent service rscdInstallDirectory\rscdsvc.log
*.log	Deploy Job log rscdInstallDirectory\Transactions\log*.log
RSCD Agent (UNIX)	
rscd.log*	UNIX RSCD Agent log rscdInstallDirectory/log/rscd.log
*.log	Deploy Job log rscdInstallDirectory/Transaction/log/*.log
BLCLI (Windows)	
blcli.log	BLCLI Windows log C:\Documents and Settings\Administrator\Application Data\BladeLogic\blcli.log

Log File Name	Description and Default Location
BMC Server Automation Console	
ui.cf	Installation dependent

The log4j audit logs provide the functionality for the TOE to meet the requirements of FAU_GEN.1 (1) and FAU_GEN.2.

7.2.2 Audit Trail

The claims made in FAU_GEN.1 (2), FAU_SAR.1 and FAU_SAR.2 reference the Audit Trail functionality. The Audit Trail view provides a record of who has sought authorization for specific actions on the current object. Each audit trail entry records the following information:

- Date when a user requests authorization to access the object
- Role trying to access the object
- User who has assumed a role in the current session
- Authorization requested
- Status (success or failure)

Audit trail records only capture information about users seeking authorization. The type of event is always object access. The audit trail does not capture changes to an object itself.

Audit trail information is stored in the Core Database and is only accessible through the TSF to authorized users. The audit trail may be viewed from the Audit Trail tab group in the console GUI.

The Audit Trail provides the functionality for the TOE to meet the requirements of FAU_GEN.1 (2), FAU_SAR.1 and FAU_SAR.2.

7.3 Cryptographic Support

The TOE integrates two CMVP certified cryptographic modules in support of the cryptographic functions.

The following tables identify the cryptographic modules which have been validated to FIPS 140-2 by the Cryptographic Module Validation Program (CMVP), and the algorithms validated by the Cryptographic Algorithm Validation Program (CAVP) that are used within the TOE.

Table 20 – TOE Cryptographic Modules

BSA Component	Cryptographic Module Name	Overall Security Level	Certificate Number
RSCD Agent	OpenSSL FIPS Object Module	1	1051
Application Server Console BLCLI Network Shell	RSA BSAFE® Crypto-J JCE Provider Module (Software Version: 4.0)	1	1048

Table 21 – TOE Algorithm Implementations

Algorithm	OpenSSL FIPS Object Module	RSA BSAFE® Crypto-J JCE Provider Module
AES	695	669
RSA	323	311
SHS[1]	723	702
HMAC	373	353

Session keys are protected and destroyed within the resources provided by the crypto libraries in the FIPS 140-2 validated cryptographic modules. For example, sessions keys are cleared using the SensitiveData.clear zeroization function from the RSA Crypto-J library.

The cryptographic support function of the BSA enforces the FCS_CKM.1 (1), FCS_CKM.1(2), FCS_CKM.4, FCS_COP.1 (1), FCS_COP.1 (2), FCS_COP.1 (3), and FCS_COP.1 (4), requirements through the implementation of CMVP certified cryptographic modules.

7.4 User Data Protection

The TOE uses a system of role-based 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.

7.4.1 Role Based Access Control

The TOE supports authorization via ACLs and a Role Based Access Control (RBAC) model. 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 those roles to users. 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 RSCD 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 include ACLs that define user access to an RSCD Agent.

BSA controls access to objects and remote servers based on the roles defined in the system. There are two roles (RBACAdmins and BLAdmins) defined out of the box and customers can define more as per their needs. Capability of a role is defined by the authorizations granted to a role at role level. Users can be assigned one or more roles. They must pick one during logon from the list of assigned roles to perform a given task. Roles can be configured to define Agent ACLs. Agent ACLs control what type of access the role has on managed servers (read write vs read only), and what local user the role will be mapped to when connecting to the remote server.

7.4.2 Object Based Permissions

BSA allows users to assign permissions directly to objects. In BSA, object level permissions are granted to specific roles, not users. To accomplish this, an ACL is defined for every system object created in BSA. 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 BSA users to delegate authority for managing different objects within BSA to different roles. The object level permissions are stored in the Core Database.

Every object in BSA also has an object based ACL, which defines what roles have access to that specific object and what type of access. Effective permissions of a role on a given object is an intersection of authorizations defined at the role level and object level for that role. For example, there could be a role called WindowsAdmins with DeployJob.* permissions defined at the role level. This makes the role capable of creating, executing or performing any operation of deploy jobs. If there are two deploy jobs that exist in the product and only one deploy job has its object ACL defining permission for WindowsAdmins role, this role will only be granted access to that one deploy job and not to the other. Notice, this is true even when the WindowsAdmins role has full DeployJob permissions defined at the role level. So, the runtime permissions (effective permissions) for a role on a given object is defined by what authorizations are granted to the role at the role level and what access is granted to that role at the object level.

[1] SHS is the Secure Hash Standard which details the Secure Hash Algorithm, in this case, SHA-1.

7.4.3 System level authorizations

Some authorizations are defined only at the role level to determine whether users with that role can perform certain actions or not. This includes permissions to perform administration tasks within the system. There are no corresponding object level permissions for such actions as they are not performed on an object but on the system or infrastructure.

7.4.4 Access Controls at the Server Level

There is a final layer of access checks that are performed at the remote server level. These are controlled using various configuration files. These configuration files on windows targets are created in c:\windows\rsc folder, and on Linux, these are created in the /etc/rsc folder.

1. exports file: This file on a given agent is generally used to define what remote hosts have access to this agent and what type of access do they have. Best practices suggest that this file should allow read write access only from the appserver hosts. This can be achieved by having a line like following for every host that has an application server installed on it
   <Hostname or IP address of appserver host> rw

2. users.local file: This file on a given agent is generally used to define what local user should be used to impersonate[2] a connection to the agent. Best practices suggest to use this file only to define impersonation for system:system and BLAdmins (a built in role). The syntax would be as shown below:
   system:system rw, map=<local username, could be administrator or root>
   BLAdmins:* rw, map=<local username, could be administrator or root>

3. users file: Although the syntax for users file is the same, best practices suggest not to change content of this file manually as this is the file that is used to manage access to the target centrally from appserver. As per the configuration of roles' Agent ACLs tabs and ACL on a given server (object ACL for that server), agent ACLs can be pushed to a remote server using ACL push job and that generates and pushes the content of this file.

4. secure file: This file sets the communication parameters that define how client and server machines communicate. RSCD Agents, Application Servers, and client installations each have their own secure file. A client's secure file specifies how the client communicates with RSCD Agents. A server's secure file specifies how an RSCD Agent communicates with clients. An Application Server secure file specifies how the Application Server communicates with RSCD Agents and how the file server (typically created on the same host as the Application Server) communicates with clients. The secure file also determines whether a Network Shell client communicates with servers using a Network Shell proxy server.

7.4.5 Effective Permissions

The combination of role-based and object-based authorizations determines a user's effective permissions – 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.

To determine if a user in a specified role has access to an RSCD Agent, the following logic is applied:

- First, the client (Console, BLCLI or NSH) 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.

This functionality satisfies the requirements FDP_ACC.1 and FDP_ACF.1.

[2] 'Impersonate' is used here in the context of describing the mechanism for obtaining credentials and should not be confused with the action of a user pretending to be another user.

7.5 Identification and Authentication

BSA employs a two-stage procedure for authenticating client application users to their respective middle-tier servers. First, client users (i.e. users of the Console, BLCLI or NSH) authenticate to the TOE, and acquire a session credential. This is done through the console, or using the blcred. The NSH does not have its own authentication mechanism. Then, having acquired a credential, the client application establishes a TLS session with a middle tier service – either the Application Service or Network Shell Proxy Service. Once the TLS session is established, the client presents its session credential to the service, which validates the credential and uses it to establish the identity of the client user.

BSA users authenticate via username and password, using Secure Remote Password (SRP) (see RFC 2945). No actions are permitted on any of the user interfaces until the user has been identified and authenticated. SRP is a protocol used for integrating secure password authentication into networked applications by using a password and key exchange for authentication. The SRP protocol resists dictionary attacks by network intruders and protects past sessions and passwords against future compromise.

To access the BDSSA Reports Server, users present their usernames and passwords to the BDSSA Reports Server across a web interface. The client/webserver communications establishes an HTTPS session, guarding against an eavesdropper acquiring the user's credentials.

When entering passwords via the Console, BLCLI, or web interface, the passwords are obscured.

This satisfies the requirements of FIA_UID.2 , FIA_UAU.2 and FIA_UAU.7.

7.5.1 User Attribute Definition

When a user is created, information is added for the following categories:

- General Information: This includes name, an optional description and information on how the user is authenticated. (The information on how the user is authenticated is described as 'authentication data' in the SFR.) It also has information on whether or not the user account is disabled, and if the account has been automatically disabled because the user did not login for an administrator defined period of time.
- Role Selection: A user may be assigned more than one role. If more than one role is associated with a user, the user is prompted to select the active role during login.

This information is maintained by the TOE. This satisfies the requirements described in FIA_ATD.1.

7.5.2 Password Policy

Users in the RBACAdmins role may configure the password policies through the Console. The following settings are available and must be set to 'true' in the evaluated configuration:

- Password minimum length – by default or by entering "0" there is no minimum length for passwords. It should be noted that the password minimum length must be set to no shorter than 8 characters in the evaluated configuration.
- Maximum password age – by setting a maximum password age, users will be required to change their passwords at specified intervals. Entering "0" causes the passwords to never expire.
- Password complexity – password complexity rules are set to true or false. If set to true, the password must meet the following requirements:
    - Passwords cannot contain a user's account name or part of the user's account name.
    - The system applies the following rules when checking password rules:
        - Passwords must contain characters from three of the following categories:
            - Uppercase letters
            - Lowercase letters
            - Digits 0 through 9

- Non-alphanumeric characters: ~!@#$%^&*_--+=`|(){}[]:;'"<>,.?/
- Any Unicode characters that are not characterized as upper or lowercase letters.

This meets the requirements of FIA_SOS.1.

7.5.3 Authentication Failure Handling

The TOE is able to detect and lock users out after an administrator-specified number of unsuccessful login attempts. The threshold may be set by a user in the RBACAdmins role using the Console. The administrator sets a threshold, which determines how many failed log on attempts cause a user to be locked out, and a duration, which determines for how long the user is locked out. An administrator in the RBACAdmins role may unlock a user before the duration has passed. If the duration is set to '0', an administrator in the RBACAdmins role will always be required to unlock the locked out user.

This functionality meets the requirements of FIA_AFL.1.

7.6 Security Management

7.6.1 Security Management Interfaces

Security management functions are provided through the BSA Console and the BLCLI. Reporting functionality may be accessed through the web interface to the BDSSA Reports Server and the NSH may be used to perform security management activities with the RSCD Agents.

7.6.1.1 BMC Server Automation Console

The BSA Console contains global menus, toolbars, perspectives, views, objects, and content editors, referred to collectively as the console. These are used to perform the tasks required to provision and manage the data center efficiently.

7.6.1.2 BMC BladeLogic Command Line Interface (BLCLI)

The BLCLI is a Java-based CLI that allows administrators to perform many of the actions available in the BSA Console by entering commands or running scripts. The CLI provides the following capabilities:

- Entry of commands to initiate actions, such as showing all running jobs or deploying patches to groups of servers. Many advanced users prefer a CLI over a point-and-click GUI.
- Automation of complex multi-step tasks.

7.6.1.3 BMC BladeLogic Decision Support for Server Automation (BDSSA) Reports Server

The BDSSA client is a web browser that connects to the Reports Server. After a user on the reports client is authenticated, the Reports Server obtains data for reports from the reporting Data Warehouse.

7.6.1.4 Network Shell

Using the Network Shell (NSH) commands, administrators can manage the network of managed servers as one large host. NSH commands may be used to access files on local and remote hosts directly from the command line. NSH commands may also be used to write new scripts, or modify existing scripts and distribute them.

7.6.2 Management of Security Attributes

The BSA enforces security management through the BSA Role Based Access Control Policy (as described in Section 7.4) by restricting the ability to affect security attributes to only authorized users. This functionality enforces the requirements set out in FMT_MSA.1, allowing only users associated with

the RBACAdmins role the ability to change role permissions and the users assigned to roles. When a new role is created, it is not provided with any permissions until specifically added by an administrator in the RBACAdmins role. Likewise, a new user has no permissions until the authorized administrator assigns that user a role. This functionality satisfies the requirements set out in FMT_MSA.3. The ability to change object level permissions on other objects is available to users in the RBACAdmins and BLAdmins roles, and other roles that may be created with similar permissions. This satisfies the requirements of FMT_MTD.1.

7.6.3 Specification of Management Functions

The BSA Console provides the tools to manage BSA. The BSA Console allows the system administrator to create, modify, delete, read, and rename objects used to perform tasks. The BSA Console also allows the system administrator to manage applications, software packages, compliance, and remote administration.

The BMC BladeLogic Command Line Interface (BLCLI) is a Java-based CLI that allows administrators to perform many of the actions available in the BSA Console by entering commands or running scripts. The CLI provides the following capabilities:

- Entering commands to initiate actions, such as showing all running jobs or deploying patches to groups of servers.
- Automating complex multi-step tasks. For example, a script running CLI commands can add servers or populate the Depot with patches and other software---actions that would otherwise require numerous user interactions with the BSA Console.

The Console or the BLCLI may be used to exercise the system management functions listed in Table 13. This satisfies the requirements FMT_SMF.1.

7.6.4 Security Roles

The BSA Console provides several predefined users. The RBACAdmin user is a user in the RBACAdmins role and has full permission to manage roles and users in the Role-based Access Control (RBAC) Manager workspace in the BSA Console, where you can assign permissions for all users. The BLAdmin user is a user in the BLAdmins role and has Read access for all system objects within the BSA Console. RBACAdmins and BLAdmins roles may not be deleted.

Additional roles may be created and assigned the required permissions for that role. Generally, new roles should be created with the minimum permissions required by the users to do their jobs.

When defining authorizations for a role, the included authorizations apply throughout BSA whenever that role performs a particular type of action. For example, if a role is granted the DeployJob.Read authorization, that role is always capable of reading Deploy Jobs — assuming the role is also granted permission to read an individual Deploy Job object. When defining a role, an access control list (ACL) template may be specified. This acts as an object permissions template. When a user in a particular role creates an object, any permission defined in the object permissions template is automatically applied to the object being created. For example, if the ACL template grants a role DeployJob.* (that is, full authority to do anything with Deploy Jobs), that role is granted DeployJob.* whenever the role creates a Deploy Job object.

This satisfies the requirement for FMT_SMR.1.

7.7 Protection of the TSF

The TOE uses TLS for internal communications between the client tier applications (i.e., Console and BLCLI) 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. Encryption addresses the requirement for protecting data from disclosure, and message integrity addresses the requirement for protecting the data from modification when it is transmitted between separate parts of the TOE.

The TOE uses TLS for securing communications between clients and the Application Server, and between the Application Server and RSCD Agents.

For TLS, the TOE employs the TLS_RSA_WITH_AES_128_CBC_SHA cipher suite; that is:

- RSA key negotiation

- 128-bit AES block encryption algorithm
- CBC (Cipher Block Chaining) block cipher mode
- SHA1 HMAC construction for integrity protection

This satisfies the requirements for FPT_ITT.1.

7.8 Trusted Path

The TOE provides for the secure transfer 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 Java RunTime Environment (JRE) 1.6 is required. The user is presented with a certificate verifying the server's identity and the user must authenticate to BSA in order to have access, providing assured identification of the end points of this communication. This path is used to both authenticate the user and allow the user to perform management activities using the Reports Server. This satisfies the requirements for FTP_TRP.1.

7.9 Compliance Management and Reporting

BSA provides functionality to automate the management of enterprise-class data centers. Using BSA, 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.

BSA 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). Note that the TOE will determine compliance with the implemented rules based on a regulatory policy, but this does not verify compliance to the regulatory policy as a whole.
- 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, remediation 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).

The compliance functions described below enforce the requirements FCM_ASM_EXT.1 through FCM_ASM_EXT.8, FCM_SMR_EXT.1 through FCM_SMR_EXT.3 and FCM_CPR_EXT.1.

7.9.1 Snapshot Job

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 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 are 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 job 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 (CVS) format, which can be imported into databases or spreadsheets

7.9.2 Audit Jobs

An Audit Job involves constructing and comparing two snapshot files on the Application Server, and recording only the differences. The comparison files are given an .snp suffix. For Audit Jobs, master .snp files are always constructed as part of the job. For live audits, a snapshot of the master target is performed and the results are captured in .snp files. For a snapshot-based audit, the master .snp file is generated from data in the database, and may be used to demonstrate compliance to a specific security policy. The master .snp files are copied to the file server, if necessary, and shared among any Application Servers that run work items for the Audit Job.

After the master .snp files have been constructed, each target of the audit job is processed by first constructing a target .snp file and then comparing the master and target .snp files. Differences between the two .snp files are recorded in the database. For each difference detected, the asset from the target .snp file is persisted in the database, regardless of how many earlier audits might have detected the same difference. After each audit target is processed, the target .snp files for that target are discarded. Upon completion of the Audit Job, master .snp files are automatically deleted from the /tmp folder on the file server.

An Audit Job compares servers or snapshots to determine whether their configurations match a standard configuration. Audit jobs may be used to 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 may also be used to identify 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.

7.9.3 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, and provide the following details:

- 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
- location of configuration file

7.9.4 Patch Analysis and Patch Management

A Patch Analysis Job can be used to manage the configuration of patches on servers by comparing them to the desired 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).

Patch Management determines the 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) 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.

7.9.5 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, Compliance Jobs are used to ensure consistency with compliance policies.

7.9.6 File Deploy Job

File Deploy Jobs allow the user to deploy or push multiple files and directories to one 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 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.

7.9.7 Deploy Job

Deploy Job is the deployment of software packages or a BLPackage to one 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.

7.9.8 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.

7.9.9 Network Shell Script 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 servers.

7.9.10 Batch Jobs

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.

7.9.11 Reporting

BDSSA reports are generated from data that is derived from the BSA database. Data is collected, in part, using several BSA jobs. BDSSA uses ETL to transfer data from the BSA Core Database to the reports Data Warehouse. ETL may be run from the command line or as a BSA Network Shell Script Job. BDSSA provides several scripts to execute the ETL process.

BDSSA provides several built-in reports but allows for custom reports, as well. The following built in reports are available to users.

- Audit reports – Audit reports show how accurately servers are configured in relation to standard configurations in a data center. These reports show various levels of summarized and detailed information about the results of an audit on the servers. Information in Audit reports is derived from the Audit Job runs in BSA. When an Audit Job is defined, a master configuration in the form of a snapshot or a live server is identified.
- Command Usage reports – The Command Usage reports allow administrators to monitor the execution of various commands on servers and when those commands have been executed. These reports provide information derived from logs recorded by Remote System Call Daemon (RSCD) Agents. The Command Usage reports can also help troubleshoot issues with particular servers.
- Compliance reports – Compliance reports show how accurately servers are configured in relation to the compliance rules specified in component templates. These reports provide overview and detailed information about the compliance data. This information is derived from the Compliance Job runs in BSA servers. Reports contain information such as server compliance, policy compliance, number of compliant and non-compliant servers, and compliance trends.
- Inventory and Change Tracking reports – The Inventory and Change Tracking reports provide details about the hardware and software objects and the associated changes in the server environment. Virtual Inventory reports provide details about the virtual machines in the environment. Information in the Inventory and Change Tracking reports is mainly derived from the Discovery and Snapshot job runs in BSA. Some information comes from extended objects. The Change Tracking reports show information only if there are changes captured in a Snapshot Job run.
- Job Activity reports – Job Activity reports provide data and statistical measurements related to all jobs that run in BSA. These reports provide both overview and detailed information about jobs. The Job Activity reports provides information on how many jobs of a given type are running, the platforms on which they are running, their success rate, and other detailed information such as job scheduling and job approval details.
- Patch reports – Patch reports provide overview and detailed information about which vendor recommended patches are missing on servers. Patch reports show how well servers in the data center conform to vendor recommendations and can be used to track patch policy compliance over time.

- Provisioning reports – The Provisioning reports provide overview and detailed information about the provisioning activities in the data center. The information includes the system package details and the device details that are used for provisioning.
- RBAC reports – RBAC reports provide information about role level authorizations, object level authorizations, assigned roles for users, and audit trails for changes to system objects. The information is derived from the role-based access control (RBAC) settings in BSA.
- Server Activity reports – The Server Activity report provides information about the BSA activities that happened on a selected server over a selected time interval. Activities include audit job, compliance, patching, and snapshot/change tracking.
- BSA Utilization Dashboard – BSA Utilization Dashboard provides a monthly graphical and tabular view of the following information:
  - Servers managed by OS types over a selected time interval
  - Number of activities by job types (audit, compliance, patch, provisioning, and snapshot) that happened on a particular number of servers distributed per OS
  - Number of policies and users managed over a selected time interval
- KPI Dashboard reports – Key Performance Indicator (KPI) dashboards provide data related to productivity, compliance, and maturity in your server environment. Data for one month at a time is collected automatically during ETL runs. Data is collected from a variety of BSA jobs.

This satisfies the requirements for FCM_CRP_EXT.1.


bmcsoftware
BUSINESS RUNS ON I.T.™