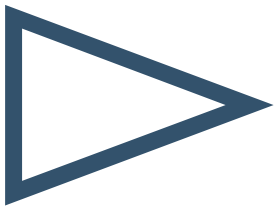


BMC® ProactiveNet Performance Management 9.5

Security Target

Version 0.4

18 July 2014

bmcsoftware
BUSINESS RUNS ON I.T.™

Document Revision History

Date	Revision	Author	Changes made
16 January 2014	0.1	TM	Initial Draft.
9 April 2014	0.2	TM	Addressed evaluator's comments
2 May 2014	0.3	TM	Addressed evaluator's comments
18 July 2014	0.4	TM	Updated document list

TABLE OF CONTENTS

1 SECURITY TARGET INTRODUCTION ____ 7

1.1 ST Reference ... 7
1.2 TOE Reference ... 7
1.3 Document References ... 7
1.4 Document Conventions ... 7
1.5 Document Terminology ... 8
1.5.1 CC Terminology ... 8
1.5.2 Abbreviations ... 9
1.6 TOE Overview ... 9
1.6.1 Usage and Major Security Features of the TOE ... 9
1.6.2 TOE Type ... 11
1.6.3 Required non-TOE Hardware and Software ... 11
1.7 TOE Description ... 12
1.7.1 Evaluated Components ... 12
1.7.2 Physical Scope and Boundary ... 12
1.7.3 BMC ProactiveNet Components ... 13
1.7.4 Logical scope and boundary ... 15
1.7.5 Functionalities and Components Excluded from the Evaluated TOE ... 16

2 CONFORMANCE CLAIMS ____ 17

2.1 Common Criteria Conformance Claim ... 17
2.2 Protection Profile Claim ... 17
2.3 Assurance Package Claim ... 17

3 SECURITY PROBLEM DEFINITION ____ 18

3.1 Threats ... 18
3.2 Organizational Security Policies ... 18
3.3 Assumptions ... 18

4 SECURITY OBJECTIVES ____ 19

4.1 Security Objectives for the TOE ... 19
4.2 Security Objectives for the Environment ... 19
4.3 Security Objectives Rationale ... 20

5 EXTENDED COMPONENTS DEFINITION ____ 23

5.1 Class FPM Performance Management ... 23
5.1.1 Performance management (FPM_COL_EXT.1) ... 23
5.2 Rationale for the Extended TOE Security Functional Components ... 23
5.3 Extended TOE Security Assurance Components ... 23

6 SECURITY REQUIREMENTS ____ 24

6.1 Security Functional Requirements ... 24
6.1.1 Security Audit (FAU) ... 24
6.1.2 User Data Protection (FDP) ... 24
6.1.3 Identification and Authentication (FIA) ... 25
6.1.4 Security Management (FMT) ... 26
6.1.5 Performance Management (FPM) ... 27
6.2 Security Assurance Requirements ... 28
6.3 Security Requirements Rationale ... 28
6.3.1 Security Functional Requirements Rationale ... 28
6.3.2 Rationale for SFR Dependencies ... 30
6.3.3 Security Assurance Requirements Rationale ... 30

7 TOE SUMMARY SPECIFICATION ____ 31

7.1 Mapping of the TSFs to SFRs ... 31
7.2 Audit ... 31
7.2.1 PATROL Agent ... 31
7.2.2 BPPM Server ... 32
7.3 Access Control ... 33
7.4 Identification and Authentication ... 33
7.5 Security Management ... 33
7.6 Performance Management ... 34

1 SECURITY TARGET INTRODUCTION

This section presents Security Target (ST) identification information and an overview of the ST for *BMC ProactiveNet Performance Management 9.5* (hereinafter referred to as *BMC ProactiveNet*).

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 IT Security Requirements sections, respectively).
- The IT security functions provided by the TOE that meet the set of requirements in the TOE Summary Specification section.

1.1 ST Reference

ST Title:	BMC® ProactiveNet Performance Management 9.5 Security Target
ST Version:	Version 0.4
ST Date:	18 July 2014

1.2 TOE Reference

TOE Identification:	BMC® ProactiveNet Performance Management 9.5 build 241196772 with PATROL Agent 140117-124726.GA (Bundled with Patrol base repository 9.5_base-20140117-122744-GA) and Knowledge Modules 140117-170323.GA (Bundled with Patrol base repository 9.5_base-20140117-122744-GA)
TOE Developer	BMC Software, Inc.
TOE Type	Network Resource Management and Optimization

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, September 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

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]*.

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).

In addition, the following general conventions are also used in this document:

5) Plain *italicized text* is used to introduce the names of TOE components and specific concepts.

6) ***Bold italicized text*** is used for emphasis.

7) `Text in Courier Font` is used to identify file name and directory paths.

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 behavior enforced by the TSF and expressible as a set of SFRs.
Security objective	A statement of an intent to counter identified threats and/or satisfy identified organization security policies and/or assumptions.
Security requirement	A requirement, stated in a standardized 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 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
BPPM	BMC ProactiveNet Performance Management
CC	Common Criteria
CI	Configuration Item
CLI	Command Line Interface
CMA	Central Monitoring Administration
CMDB	Configuration Management Database
EAL	Evaluation Assurance Level
HTTPS	Hyper Text Transfer Protocol Secure
IT	Information Technology
KM	Knowledge Module
KPI	Key Performance Indicators
LDAP	Lightweight Directory Access Protocol
PCM	PATROL Configuration Manager
PP	Protection Profile
REST API	Representational State Transfer Application Programming Interface
RLS	Read Level Security
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SFP	Security Function Policy
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functionality
WLS	Write Level Security

1.6 TOE Overview

1.6.1 Usage and Major Security Features of the TOE

The TOE is the BMC ProactiveNet Performance Management 9.5 which performs real-time predictive root cause analysis to sift through events and abnormalities collected from the application and infrastructure components that support business services, identifying a prioritized set of the most likely problem causes. This information provides continuous visibility into problems as they develop; allowing the diagnosis of intermittent performance issues without requiring users to reproduce problems. BPPM allows users to create a baseline of the system through system monitoring. The baseline is the expected normal operating range for a metric or attribute of a monitor. Abnormalities are generated when the data values from a monitor fall outside of the normal baseline range for a statistically significant number of points within the sample window specified in the threshold. When a threshold is

exceeded, BPPM registers this as an event. Event rules define the set of actions that can be performed when an event occurs. Event management involves setting thresholds and creating, modifying, deleting and querying event rules.

BMC ProactiveNet Performance Management consolidates data and events spanning multiple vendors, platforms, and sources. It supports agent-less and agent-based monitoring of infrastructure, applications, real and synthetic end-user transactions, SNMP networks, configuration changes, business metrics, and custom information. In addition, it collects data and events from non-BMC monitoring and event management tools. Finally, it monitors on-premise and public cloud resources (e.g., storage, UCS) and services, to include Software as a Service (SaaS), Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) resources. Through analysis, BMC ProactiveNet automatically discovers and learns the behavioral and performance trends for each of the monitored applications and services, identifying normal and abnormal behavior. Events are only generated when significant abnormal behavior is detected. This creates a manageable number of intelligent events, allowing the most critical business issues to be prioritized.

BMC ProactiveNet Performance Management automatically discovers and learns the behavioral and performance trends for each of the monitored application and service components; identifying normal and abnormal behavior. Unlike threshold-based monitoring systems, the analytics engine only generates events when there is significantly abnormal behavior. Based on these learned trends, customers eliminate their reliance on reactive thresholds, capture critical events missed by static thresholds, and realize a significant reduction in the number of false events generated by reactive, threshold-based approaches. With fewer, more intelligent events administrators may more easily pinpoint and prioritize the most critical business issues.

The solution applies predictive correlation and filtering techniques that leverage real-time service relationships in the BMC Atrium CMDB and configuration changes collected from BMC BladeLogic (or any other change source), along with additional detailed diagnostic data. This information provides continuous visibility into problems as they develop; allowing administrators to diagnose intermittent performance issues without requiring reproduction of the problem. As a result, problem resolution time is greatly reduced.

BMC ProactiveNet Performance Management supports a number of features that enable users to adapt their implementation to meet the needs of their own networks and architectures. They include:

- Use of external authentication such as Active Directory or BMC Atrium Single Sign On
- BMC ProactiveNet Performance Management supports the use of either an embedded Sybase database, or an external Oracle database. The external Oracle database is used in the evaluated configuration.
- Support for the BMC Atrium CMDB for Configuration Item based access control
- The BMC ProactiveNet Performance Management server includes an Apache server which may be configured to provide HTTPS protected communications between distributed parts of the product

The PATROL Agent is the core piece of the BMC ProactiveNet architecture that monitors and manages host computers. The PATROL Agent performs the following tasks:

- Runs commands to collect system or application information; the information is collected according to applications and parameters defined in Knowledge Modules
- Stores information locally for retrieval by the BMC ProactiveNet Integration Service
- Loads specified Knowledge Modules (KMs) at start-up

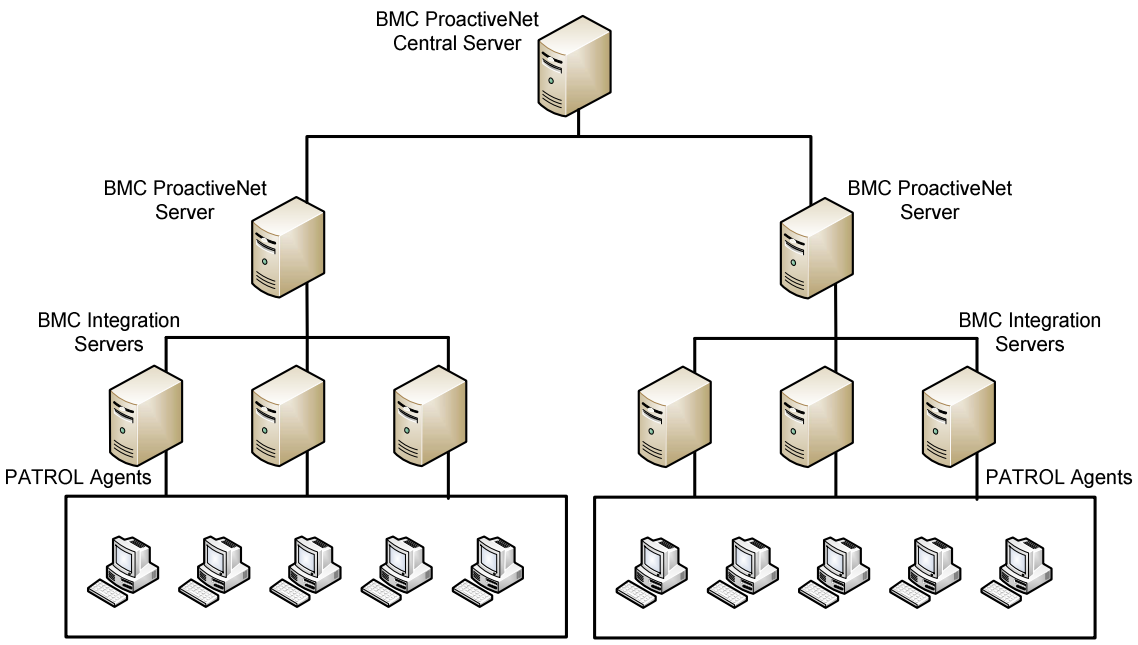
The PATROL Knowledge Module is a set of files from which a PATROL Agent receives information about all of the resources, such as databases and file systems running on a monitored computer. PATROL KMs provide information to the PATROL Agent about:

- The identity of objects
- Parameters
- Actions to take when an object changes a state
- How to monitor the application

Communications between components is protected using HTTPS protocols.

Figure 1 shows a typical implementation configuration for BMC ProactiveNet Performance Management.

Figure 1– BMC ProactiveNet



1.6.2 TOE Type

BMC ProactiveNet Performance Management is an IT management application.

1.6.3 Required non-TOE Hardware and Software

The hardware requirements for any given environment depend on the size and amount of activity expected. In most cases, BMC recommends that an analysis of the organization's needs be performed to determine the hardware requirements for the installation.

The following tables identify the requirements for all components of the BMC ProactiveNet,

Table 1 – Operating System and Hardware Requirements for TOE Components

TOE Component	Operating System	Hardware
BMC ProactiveNet Server	Red Hat Enterprise Linux 6.2 (64-bit)	Intel Xeon CPU E5645 @ 2.40 GHz, 2 CPUs and a total of 2 Cores, minimum 16 GB memory
BMC ProactiveNet Integration Service	Red Hat Enterprise Linux 6.2 (64-bit)	Intel Core i7, 2 CPUs and a total of 2 Cores, 3.067 GHz frequency, 2400 MHz bus-speed, and 16 threads or equivalent, minimum 8 GB memory
BMC PATROL Agent	Windows 7 (64-bit)	General Purpose Computer Hardware
	Red Hat Enterprise Linux 6.2 (64-bit)	General Purpose Computer Hardware

Table 2 –Requirements for Non-TOE Components

Non-TOE Component	Requirement
Web browser	Microsoft Internet Explorer 8, Adobe Flash V10 or later, display resolution 1280x1024

Non-TOE Component	Requirement
External Oracle Database	Oracle Database 11g r2 (11.2.0.2) or higher Standard Edition, 64-bit production. In an Oracle RAC environment, Oracle Database Enterprise Edition is recommended.

1.7 TOE Description

This section provides context for the TOE evaluation by identifying the product type and describing the evaluated configuration.

1.7.1 Evaluated Components

Table 3 identifies the BMC ProactiveNet components included in the evaluated configuration. The "abbreviated name" is used in this Security Target for discussion purposes.

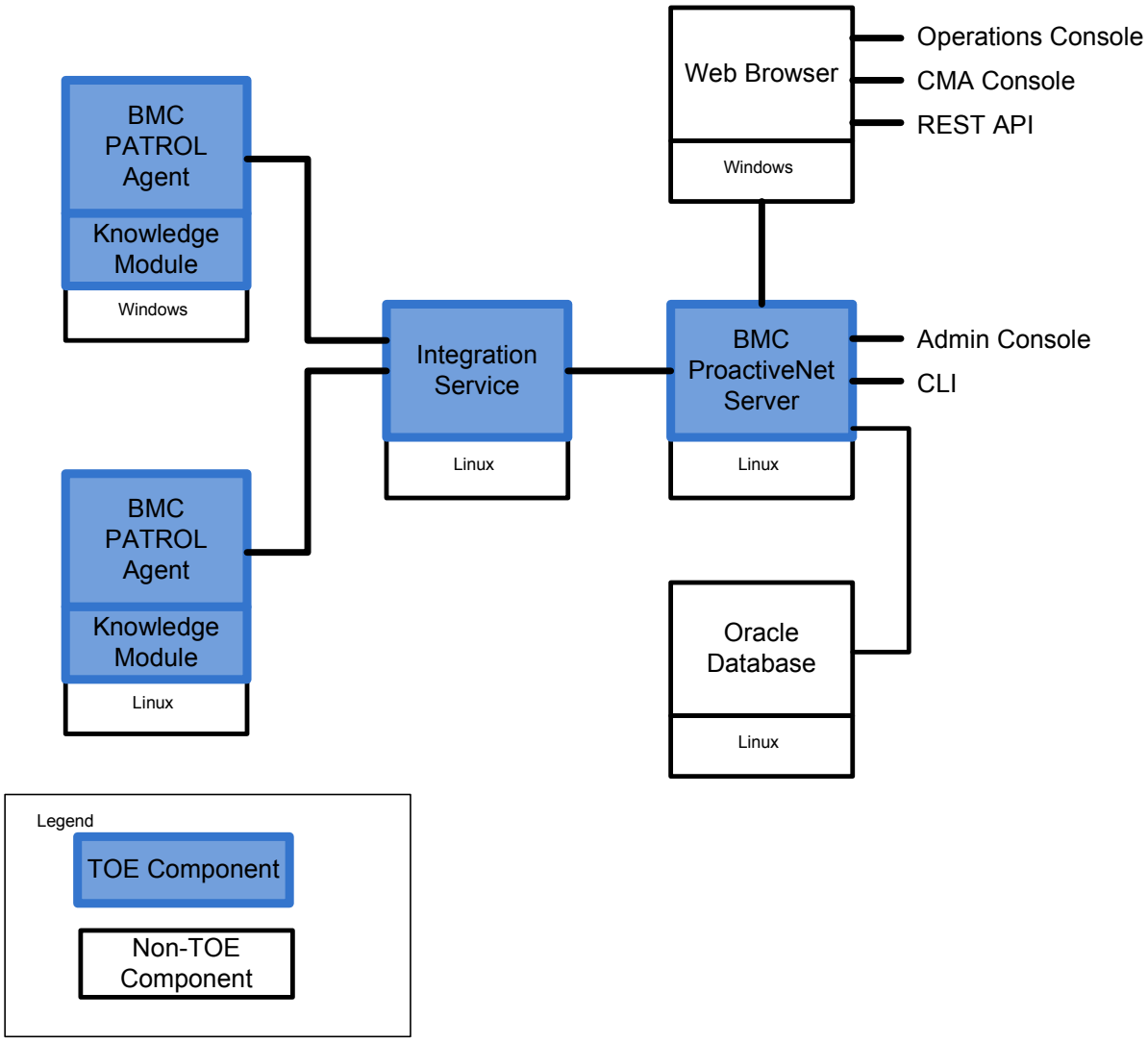
Table 3 – BMC ProactiveNet component names

BMC ProactiveNet component name	Abbreviated name
BMC ProactiveNet Server	*ProactiveNet Server, BPPM Server*
BMC ProactiveNet Integration Service	*Integration service*
BMC PATROL Agents	*PATROL agent(s), Agent(s)*

1.7.2 Physical Scope and Boundary

Figure 2 identifies the physical scope and the physical boundary of the TOE, as well as showing how all the components of the TOE tie together with IT systems in the enterprise.

Figure 2– BMC ProactiveNet TOE Boundary



1.7.3 BMC ProactiveNet Components

1.7.3.1 BMC ProactiveNet Server

The BMC ProactiveNet Server configures and controls data and event collection, stores the data, and presents that data in the form of graphs, reports, views, and events. It also provides the following functionality:

- Performs collection and analysis functions
- Performs authentication and access control
- Runs the BMC ProactiveNet Operations Console
- Runs the BMC ProactiveNet Central Monitoring Administration Console, which allows users to configure PATROL agents and knowledge modules based on monitoring profiles

- Provides the BMC Patrol Configuration Manager Console to manage PATROL Agents and Knowledge Modules which provides rule management features, such as basic change control, rule organization, and rule deployment methods
- Provides the BMC ProactiveNet Administration Console which enables users to modify and manage the BMC ProactiveNet Server and the Integration Service network management areas by adding or deleting users, groups, Integration Services, monitored devices, applications, and services, or changing event notifications and thresholds
- Controls the actions of the PATROL Agents
- Includes an Apache HTTPD server to support HTTPS communications to other components
- Includes an embedded database to store data
- Includes a Command Line Interface (CLI) for setup and diagnostic activities

1.7.3.2 Administrative Interfaces

The TOE is administered via a number of interfaces accessed directly, or via a web browser. They are:

a. The Operations Console: This is a web Interface accessible from https://server.domain.name, operated through a supported browser. This console is used to:

   - administer events
   - generate reports
   - determine the probable cause for an event

   This interface is described in the BMC ProactiveNet User Guide;

b. Central Monitoring Administration (CMA): Also known as the CMA Console, this is a web Interface accessible from https://server.domain.name/admin. It is used to configure PATROL agents, Integration Servers, thresholds, and users. On a multi-server deployment, the CMA web application is hosted on the BPPM server, and is described in BMC CMA Administering-v57-20131105_1528.

c. BMC ProactiveNet Administration Console (or Admin Console): This console may be used to:

   - configure connections
   - manage devices
   - manage users, groups and roles

   This console is described in the BMC ProactiveNet Administrator Guide. In the evaluated configuration, this console is installed on the BPPM Server and accessed directly.

d. The BMC ProactiveNet Command Line Interface (CLI) is a collection of BMC ProactiveNet and Event Management commands that may only be run from the BMC ProactiveNet server as the root (on Linux) or the Administrator (on Windows). The CLI provides an alternative to the Administrative Console for setup operations, and provides diagnostics capabilities. This interface is described in the BMC ProactiveNet Command Line Interface Reference Guide. Use of SSH or Remote Desktop to access this CLI remotely is prohibited in the evaluated configuration.

e. REST API: This is a programmatic interface hosted by the server. It may be used in place of the CLI and is documented in BMC ProactiveNet Performance Management Web Services Reference Guide.

1.7.3.3 BMC ProactiveNet Integration Service

The BMC ProactiveNet Integration Services enable the BMC ProactiveNet Server to remotely gather statistical data from all supported operating systems. The BMC ProactiveNet Integration Service acts as a concentrator, collecting data from BMC PATROL Agents and providing the data to the BMC ProactiveNet Server.

1.7.3.4 BMC PATROL Agents

The BMC PATROL Agent is a core component of the BMC ProactiveNet architecture that monitors and manages the distributed environment. A BMC PATROL Agent performs the following tasks:

- Run commands to collect information either locally or remotely; the information is collected according to the applications and parameters defined in Knowledge Modules (KMs)
- Act as a service provider for event and data management for BMC ProactiveNet Server
- Interpret data by using defined rules
- Load specified KMs at start-up, runs commands, and updates configuration information
- Store information locally

Knowledge Modules are packaged monitors that acquire performance and availability data for a defined operating system or application environment. They:

- Run under the control of the PATROL agents
- May monitor the local managed system or remote systems without agents
- Pass discovered instance and monitor data to the PATROL agent

Only the system KM is enabled in the evaluated configuration

1.7.3.5 Guidance Documentation

The TOE includes the following guidance:

- BMC PATROL Agent Reference Manual
- BMC PATROL for Microsoft Windows Servers Getting Started Guide
- BMC PATROL for UNIX and Linux Getting Started Guide
- BMC ProactiveNet 9.5
-

1.7.4 Logical scope and boundary

The TOE provides the following security functions:

- Audit
- User Data Protection
- Identification and Authentication
- Security Management
- Performance Management

1.7.4.1 Audit

BMC ProactiveNet provides the capability to audit actions at the BMC PATROL Agent and the BMC ProactiveNet Server. The TOE relies on the operational environment to store and protect the audit data, and provide a means to view the data. It also relies on the TOE environment to provide an appropriate time stamp for use in the audit records.

1.7.4.2 User Data Protection

Access to data in BMC ProactiveNet administrative features is controlled through Configuration Item based access controls. Only users belonging to a group with read and/or write permissions for a Configuration Item are able to perform actions on that Configuration Item.

1.7.4.3 Identification and Authentication

BMC ProactiveNet identifies users by their user names. By default, users who access BMC ProactiveNet through a web browser or Java console are prompted for a user name and password by BMC ProactiveNet and must be identified and authenticated before they can access the system. After identification and authentication, the user name is then used as part of every ProactiveNet Server request, since no action can be taken unless a valid user name is associated with it.

The CLI allows access to users logged into the underlying operating system. In the evaluated configuration, the CLI may only be accessed locally on the BMC ProactiveNet Server, and only the root (Linux) or local administrator (Windows) have access to this machine.

1.7.4.4 Security Management

The BMC ProactiveNet includes a number of interfaces allowing local and remote management of the security functions. These functions support the administrator's ability to:

- configure the system by managing devices
- manage users and groups
- configure event attributes to support performance management functions

1.7.4.5 Performance Management

The TOE natively collects and leverages key performance indicators (KPIs) to assess the relevance and impact of abnormalities, alerts, events and trends from the applications and infrastructure components that support enterprise IT services, identifying a prioritized set of the most likely problem causes.

1.7.5 Functionalities and Components Excluded from the Evaluated TOE

The following components and features are excluded from this evaluation:

- External Authentication Provider. For external authentication, BMC ProactiveNet can be configured to use Lightweight Directory Access Protocol (LDAP) or to integrate with the BMC Atrium Single Sign On component.
- BMC Configuration Management Database (CMDB)
- All PATROL legacy interfaces (i.e. PATROL Configuration Manager, PATROL Central Operator, PATROL Console Server, PATROL Operator Console, PATROL Developer Console, PATROL Event Manager, PATROL Integration Products, PATROL Agent Query and PATROL Command Line Interface)

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

Interpretations of the Common Methodology for Information Technology Security Evaluation, Version 3.1, Revision 4, September 2012 [CEM] have been 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 (EAL) 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.TAMPER	A hostile/unauthorized user may be able to modify TOE behavior or gain TOE access by tampering with the TOE or the TOE operational environment.
T.UNAUTH	A hostile/unauthorized user may be able to read TOE data/configuration files in order to ascertain TOE, or managed application, secrets, or modify TOE behavior.
T.UNDETECT	Authorized or unauthorized users may be able to access TOE data or modify TOE behavior without a record of those actions in order to circumvent TOE security functionality.
T.PERFORM	Network problems caused by attackers or environmental conditions may go undetected, causing network performance to degrade.

3.2 Organizational Security Policies

There are no organizational security policies defined for this ST.

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.ACCESS	The operating systems upon which the TOE software runs are under the same administrative management as the TOE. The operating systems upon which the TOE software runs is configured to restrict modification to TOE executables and configuration files to only Authorized TOE Administrators.
A.MANAGE	There will be one or more competent individuals assigned to manage the TOE. Those assigned to manage the TOE are appropriately trained and follow all administrator guidance.
A.OPERATE	The computer platforms and operating systems upon which the TOE software runs operates correctly.
A.PEER	Any other systems that communicate with the TOE are under the same management control and will operate under the same security policy constraints.
A.PHYSICAL	The processing resources of the TOE will be located within facilities providing controlled access to prevent unauthorized physical access. Only authorized users will have physical access to the server platforms and are expected to follow all security policies.
A.AUTHORIZED	Only authorized TOE users and administrators will have accounts on the operating system platforms on which the TOE software executes.
A.TIME	The operational environment will provide reliable system time.

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.ADMIN	The TOE must provide functions to enable administrators to effectively manage and maintain the TOE and its security functions, ensuring that only they can access administrative functionality and TOE data.
O.AUDIT	The TOE must provide an audit capability to report security relevant events so that the responsible subjects can be held accountable for their actions.
O.ACCESS	The TOE must be able to control access to configuration items.
O. IDENTIFICATION	The TOE must identify users to verify that permission for access to TOE components, or data is authorized.
O.ROLES	The TOE must accommodate separate roles for Authorized Administrators to limit their access to the TOE security mechanisms.
O.PERFORMANCE	The TOE must be capable of collecting and analyzing data to identify the causes of system and network performance and availability issues.

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.PLATFORM	The TOE environment must provide appropriate and reliable hardware and software to support the TOE.
OE.PROTECT	The TOE environment must protect itself and the TOE from external interference or tampering.
OE.SERVER_ACCESS	Access to the TOE server hardware and operating system must be limited to authorized administrators.
OE.INSTALL	Those responsible for the TOE must ensure that the TOE is delivered, installed, managed, and operated in a manner that maintains IT security objectives.
OE.TIME	The TOE operational environment must provide correct system time to facilitate reliable timestamps.

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 correspondence

	Threats			
	T.TAMPER	T.UNAUTH	T.UNDETECT	T.PERFORM
O.ADMIN	X	X		
O.AUDIT	X		X	
O.ACCESS	X	X		
O.IDENTIFICATION	X	X		
O.ROLES	X	X		
O. PERFORMANCE				X
OE.PLATFORM	X			
OE.PROTECT	X	X		
OE.SERVER_ACCESS	X	X		
OE.INSTALL	X			
OE.TIME				

Table 9 – Security objective to assumptions correspondence

	Assumptions						
	A.ACCESS	A.MANAGE	A.OPERATE	A.PEER	A.PHYSICAL	A.AUTHORIZED	A.TIME
O.ADMIN							
O.AUDIT							
O.ACCESS							
O.IDENTIFICATION							
O.ROLES							
O. PERFORMANCE							
OE.PLATFORM			X		X		
OE.PROTECT	X				X		
OE.SERVER_ACCESS	X	X		X	X	X	
OE.INSTALL		X					
OE.TIME							X

Table 10 – Security objectives rationale for the TOE

Objective	Threat or OSP	Rationale
O.ADMIN	**T.TAMPER**	O.ADMIN counters this threat by providing administrative functionality only to authorized administrators.
	T.UNAUTH	O.ADMIN counters this threat by ensuring that only authorized administrators may access TOE data.
O.AUDIT	**T.TAMPER**	O.AUDIT counters this threat by ensuring that actions causing modification to TOE behavior are recorded.
	T.UNDETECT	O.AUDIT counters this threat by ensuring that changes to TOE data and behavior are recorded, holding those who perform those activities responsible for their actions.
O.ACCESS	**T.TAMPER**	O.ACCESS counters this threat by ensuring that access to TOE configuration items is restricted to authorized users.
	T.UNAUTH	O.ACCESS counters this threat by ensuring that access to data is controlled, and limited to authorized personnel.
O. IDENTIFICATION	**T.TAMPER**	O.IDENTIFICATION counters this threat by ensuring that users are identified before having access to the TOE applications that facilitate the modification of TOE behavior.
	T.UNAUTH	O.IDENTIFICATION counters this threat by ensuring that only authorized users, based on identification, are permitted access to the TOE.
O.ROLES	**T.TAMPER**	O.ROLES counters this threat by ensuring that TOE administrators and users have access to only those functions required to perform their duties.
	T.UNAUTH	O.ROLES counters this threat by ensuring that TOE administrators and users have access to only that TOE data required to perform their duties.
O.PERFORMANCE	**T.PERFORM**	O.PERFORMANCE counters this threat by providing the capabilities to collect and analyze data to identify the causes of system and network performance and availability issues.

Table 11– Environment security objectives rationale for the TOE

Objective	Threat, OSP, Assumption	Rationale
OE.PLATFORM	**T.TAMPER**	OE.PLATFORM counters this threat by ensuring that appropriately secure hardware and software support the TOE.
	A.OPERATE	OE.PLATFORM upholds this assumption by ensuring that the hardware and software supporting the TOE operate correctly
	A.PHYSICAL	OE.PLATFORM upholds this assumption by ensuring the hardware and software support the policies required to ensure secure operation.
OE.PROTECT	**T.TAMPER**	OE.PROTECT counters this threat by ensuring that the TOE is protected from external interference or tampering.
	T.UNAUTH	OE.PROTECT counters this threat by ensuring that the TOE and the TOE environment are protected from unauthorized users.
	A.ACCESS	OE.PROTECT upholds this assumption by ensuring that the TOE environment is protected from external interference.
	A.PHYSICAL	OE.PROTECT upholds this assumption by ensuring that the TOE and its environment are protected from external interference and tampering.
OE.SERVER_ACCESS	**T.TAMPER**	OE.SERVER_ACCESS counters this threat by ensuring that TOE access is limited to authorized administrators.

Objective	Threat, OSP, Assumption	Rationale
	T.UNAUTH	OE.SERVER_ACCESS counters this threat by ensuring that TOE access is limited to authorized administrators, mitigating the risk of access by hostile/unauthorized users.
	A.ACCESS	OE.SERVER_ACCESS upholds this assumption by ensuring that TOE access is limited to authorized administrators, mitigating the risk of access by other users.
	A.MANAGE	OE.SERVER_ACCESS upholds this assumption by ensuring that TOE access is limited to authorized administrators, allowing only competent authorized users to manage the TOE.
	A.PEER	OE.SERVER_ACCESS upholds this assumption by ensuring that TOE is accessible from systems under the same management, mitigating the risk of access by hostile/unauthorized users.
	A.PHYSICAL	OE.SERVER_ACCESS upholds this assumption by ensuring only authorized administrators have physical access to the TOE server resources.
	A.AUTHORIZED	OE.SERVER_ACCESS upholds this assumption by ensuring that only authorized administrators have access to the operating systems on which the TOE is installed.
OE.INSTALL	**A.MANAGE**	OE.INSTALL upholds this assumption by ensuring the TOE is installed and managed correctly by competent users who follow all guidance.
	T.TAMPER	OE.INSTALL upholds this assumption by reducing the risk of tampering through proper installation, management and operation.
OE.TIME	**A.TIME**	OE.TIME upholds this assumption by ensuring that the operational environment provides reliable time to the TOE.

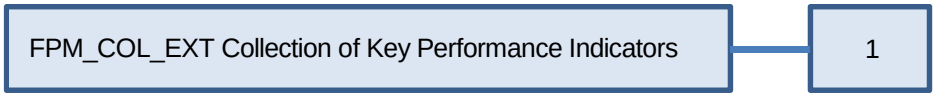
5 EXTENDED COMPONENTS DEFINITION

This section specifies the extended Security Functional Requirement (SFR) used in this ST.

5.1 Class FPM Performance Management

The Performance Management class addresses the collection and analysis of data used to identify and prioritize problems within a network. The class is modeled on the FAU Security Audit class. One family, Collection of Key Performance Indicators, is defined for this class. The FPM_COL_EXT Collection of Key Performance Indicators family is modeled after FAU_GEN Security audit data generation, and FPM_COL_EXT.1 was modeled after FAU_GEN.1. Component leveling is shown in Figure 3 below.

Figure 3– Component Leveling



5.1.1 Performance management (FPM_COL_EXT.1)

Family Behavior: This family defines a requirement for ensuring that the TOE natively collects and leverages key performance indicators (KPIs) to assess the relevance and impact of abnormalities, alerts, events and trends from the applications and infrastructure components that support enterprise IT services, identifying a prioritized set of the most likely problem causes.

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

- Managing event rules.

Audit Activity: The following actions should be auditable if FAU_GEN Security Audit data generation is included in the ST:

- Basic: Administrator changes to the event rules.

FPM_COL_EXT.1	**Collection of Key Performance Indicators**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPM_COL_EXT.1 .1	The TSF shall be able to collect and leverage Key Performance Indicators (KPIs) to assess the relevance and impact of events to identify likely problem causes.

5.2 Rationale for the Extended TOE Security Functional Components

FPM_COL_EXT.1 was created to capture the basic functionality provide by 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

Table 12 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 12 – TOE security functional requirements

Class	Functional component	A	S	R	I
Security Audit (FAU)	**FAU_GEN.1** Audit data generation	X	X		
User Data Protection (FDP)	**FDP_ACC.1** Subset access control	X		X	
	FDP_ACF.1 Security attribute based access control	X			
Identification and Authentication (FIA)	**FIA_ATD.1** User attribute definition	X			
	FIA_UAU.1 Timing of authentication	X			
	FIA_UID.1 Timing of identification	X			
Security Management (FMT)	**FMT_MSA.1** Management of security attributes	X	X		
	FMT_MSA.3 Static attribute initialization	X	X		
	FMT_SMF.1 Specification of Management Functions	X			
	FMT_SMR.1 Security roles	X			
Performance Management (FPM)	**FPM_COL_EXT.1** Collection of key performance indicators				

6.1.1 Security Audit (FAU)

FAU_GEN.1	**Audit data generation**
Hierarchical to:	No other components.
Dependencies:	FPT_STM.1 Reliable time stamps
FAU_GEN.1.1	The TSF shall be able to generate an audit record of the following auditable events: a) Start-up and shutdown of the audit functions; b) All auditable events for the *not specified* level of audit; and c) [none].
FAU_GEN.1.2	The TSF shall record within each audit record at least the following information: a) Date and time of the event, type of event, subject identity (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, [none].

6.1.2 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 [Administrative Access Control SFP] on [Subjects: BPPM Administrators and Users Objects: BPPM Configuration Items Operations: event management operations.] **using the Operations Console, CMA Console, Administration Console, and REST API.**

Application Note: Event management includes setting thresholds for the detection of abnormal behaviour and creating, modifying, deleting and querying event rules, which define the set of actions that may be performed when an event occurs.

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 [Administrative Access Control SFP] to objects based on the following: [Subjects: BPPM Administrators and Users Subject Attributes: Group membership Objects: BPPM Configuration Items Object Attributes: Read Level Security and Write Level Security settings].
FDP_ACF.1.2	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [a. access to a Configuration Item is allowed if the Administrator or User belongs to a Group associated with that Configuration Item; and b. the User Group name must be present in the Read Level Security and/or Write Level Security setting associated with the Configuration Item.]
FDP_ACF.1.3	The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [Users in the Administrator role have access to all Configuration Items].
FDP_ACF.1.4	The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [none].

6.1.3 Identification and Authentication (FIA)

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: [a) User Name, b) Password, c) User Group].

FIA_UAU.1	**Timing of authentication**
Hierarchical to:	No other components
Dependencies:	FIA_UID.1 Timing of identification.
FIA_UAU.1.1	The TSF shall allow [CLI Access to the root or Administrator user at the physical console] on behalf of the user to be performed before the user is authenticated.

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

FIA_UID.1.1	The TSF shall allow [CLI Access to the root or Administrator user at the physical console] on behalf of the user to be performed before the user is identified.

6.1.4 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 [Administrator Access Control SFP] to restrict the ability to *query, modify, delete,* the security attributes [event attributes, users and groups] to [authorized Administrators].

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 [Administrator Access Control SFP] to provide *permissive* default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2	The TSF shall allow the [any Administrator with access] to specify alternative initial values to override the default values when an object or information is created.

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: [device management, event management, user and group management].

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 [BPPM Administrator, BPPM Model Administrator, BPPM Monitoring Administrator, BPPM Operator, BPPM Supervisor, BPPM Viewer, and BPPM WS Full Access].
FMT_SMR.1.2	The TSF shall be able to associate users with roles.

6.1.5 Performance Management (FPM)

FPM_COL_EXT.1	Collection of key performance indicators
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPM_COL_EXT.1 .1	The TSF shall be able to collect and leverage Key Performance Indicators (KPIs) to assess the relevance and impact of events to identify likely problem causes.

6.2 Security Assurance Requirements

The TOE satisfies the SARs listed in Table 13.

Table 13 – 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
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 14 – Objective to requirement correspondence

	O.ADMIN	O.AUDIT	O.ACCESS	O.IDENTIFICATION	O.ROLES	O.PERFORMANCE
FAU_GEN.1 Audit data generation		X				
FDP_ACC.1 Subset access control			X		X	
FDP_ACF.1 Security attribute based access control			X		X	
FIA_ATD.1 User attribute definition				X		
FIA_UAU.1 Timing of authentication			X	X		
FIA_UID.1 Timing of identification			X	X		
FMT_MSA.1 Management of security attributes	X					
FMT_MSA.3 Static attribute initialization	X					
FMT_SMF.1 Specification of Management Functions	X					
FMT_SMR.1 Security roles					X	
FPM_COL_EXT.1 Collection of key performance indicators						X

Table 15 – Security functional requirements rationale for the TOE

Objective	SFRs	Rationale
O.ADMIN	**FMT_MSA.1** **FMT_MSA.3** **FMT_SMF.1**	FMT_MSA.1 ensures that the TOE is able to manage security attributes required to support the Administrative Access Control SFP, and FMT_MSA.3 specifies that permissive defaults will be used where possible. FMT_SMF.1 ensures that the appropriate management functions are in place to enable administrators to manage security functions.
O.AUDIT	**FAU_GEN.1**	FAU_GEN.1 specifies that audit data is generated, ensuring that subjects may be held accountable for their actions.
O.ACCESS	**FDP_ACC.1** **FDP_ACF.1** **FIA_UAU.1** **FIA_UID.1**	FDP_ACC.1 and FDP_ACF.1 describe the Administrative Access Control SFP that determines how access control to configuration items is performed. FIA_UAU.1 and FIA_UID.1 ensure that access control is applied to identified and authenticated users.
O.IDENTIFICATION	**FIA_ATD.1** **FIA_UAU.1** **FIA_UID.1**	FIA_UAU.1 and FIA_UID.1 ensure that users are identified and authenticated. FIA_ATD.1 ensures that users may be mapped to the User Groups that determine their access permissions.
O.ROLES	**FDP_ACC.1** **FDP_ACF.1** **FMT_SMR.1**	FDP_ACC.1 and FDP_ACF.1 describe that Administrative Access Control SFP that limits access to TOE security mechanisms based on roles. FMT_SMR.1 ensures that separate roles are available to limit access to security functionality for different users.
O.PERFORMANCE	**FPM_COL_EXT.1**	FPM_COL_EXT.1 ensures that the TOE has the capability to collect and analyze data to identify the cause of system and network performance and availability issues.

6.3.2 Rationale for SFR Dependencies

Table 16 lists the functional components, their related dependencies, and whether the dependency was satisfied, or provides justification for why the dependency is not satisfied.

Table 16 – SFR dependency status

SFR	Dependencies	Fulfilled by SFRs in this ST
FAU_GEN.1	FPT_STM.1	Timestamps are provided by the environment in accordance with the OE.TIME objective.
FDP_ACC.1	FDP_ACF.1	FDP_ACF.1
FDP_ACF.1	FDP_ACC.1 FMT_MSA.3	FDP_ACC.1 FMT_MSA.3
FIA_ATD.1	None	Not applicable
FIA_UAU.1	FIA_UID.1	FIA_UID.1
FIA_UID.1	None	Not applicable
FMT_MSA.1	[FDP_ACC.1 or FDP_IFC.1] FMT_SMR.1 FMT_SMF.1	FDP_ACC.1 FMT_SMR.1 FMT_SMF.1
FMT_MSA.3	FMT_MSA.1 FMT_SMR.1 FMT_SMF.1	FMT_MSA.1 FMT_SMR.1 FMT_SMF.1
FMT_SMF.1	None	Not applicable
FMT_SMR.1	FIA_UID.1	FIA_UID.1
FPM_COL_EXP.1	None	Not applicable

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 installed on or 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 17 provides a mapping of SFRs to the TSFs to show that each SFR is captured within a security function.

Table 17 – Mapping of TSFs to SFRs

TSF	SFR
Audit	**FAU_GEN.1** Audit data generation
Access Control	**FDP_ACC.2** Complete access control
	FDP_ACF.1 Security attribute based access control
Identification and Authentication	**FIA_ATD.1** User attribute definition
	FIA_UAU.1 Timing of authentication
	FIA_UID.1 Timing of identification
Security Management	**FMT_MSA.1** Management of security attributes
	FMT_MSA.3 Static attribute initialization
	FMT_SMF.1 Specification of Management Functions
	FMT_SMR.1 Security roles
Performance Management	**FPM_COL_EXT.1** Collection of key performance indicators

7.2 Audit

The PATROL Agent and BMC ProactiveNet Server components perform auditing of security relevant activities. The TOE relies on the operational environment to store and protect the audit data, and provide a means to view the data. It also relies on the TOE environment to provide an appropriate time stamp for use in the audit records.

7.2.1 PATROL Agent

The auditing feature is controlled by the configuration variable `/AgentSetup/auditLog`. The standard PATROL installation process does not create this variable. The PATROL System Administrator must create and set this variable to enable audit logging.

Table 18 – Audit Log Entries for PATROL Agent

Functional Component	Auditable Event	Audit Record Contents
PATROL Agent	Spawned Commands	The entry in the log file records explicitly created external processes.
PATROL Agent	Commands executed	The audit log records each command (i.e. script) that is executed as a result of a Menu Command or an InfoBox Command. The entry in the log file records the console-ID of the peer and the local account name used for the connection.
PATROL Agent	Connect/Disconnect	The entry in the log file records each connection/disconnection.
PATROL Agent	Commit Actions	The entry in the log file records each file that is transferred during a commit.
PATROL Agent	Configuration Actions	The entry in the log file records each explicit pconfig, wpconfig, or xpconfig action that affects the state of the PATROL Agent.

The PATROL Agent uses a number of audit log keys and values to determine the behavior of audit logging. The Audit Log configuration variable, /AgentSetup/auditLog, consists of a new line separated list of KEY=VALUE pairs, with the following possible values:

Key	Description
Active	This key determines whether the audit logging feature is turned on or off, and where the information is being logged. The recognized values include: 0 - turns off audit logging and is the default setting 1 - logs information to a file (This is the evaluated configuration setting) 2 - log information is sent to the Applications log 3 - logs information to both a file and Windows Event Log
Delimiter	This key determines the delimiter character that separates the fields in the log file. The default character is the pipe-symbol '|'
FileAging	This key determines the interval at which a new log file is created. -Daily N - create a new log file every day at approximately the hour N, where N ranges from midnight 12 a.m. represented as 0 to 11 p.m. represented as 23; the default is Daily 0 -Entries N - create a new log file after logging N entries, where N is the number of entries; for example, N >= 100 -Size N - create a new log file when the file reaches a designated size, where N is the file size in KB; for example, N >= 32
FileCount	This key determines how many old log files are retained. The default value is 5. Each time a new logfile is created, the previous files are renamed.
FileName	This key determines the pathname and filenaming convention for the audit log file. The name can contain the following macros %H- refers to the current Agent-host %P- refer to the port-number being used If path is not a fully qualified pathname, the PATROL Agent treats it as being relative to the PATROL_HOME/log directory. All subdirectories in the pathname must already exist. PATROL Agent creates the log file but not the directories leading up to the file. If the file cannot be opened, the Agent writes an error message to the Agent's log file. The default path and file name is NT—%PATROL_HOME%\log\PatrolAgent-%H-%P.audit UNIX—$PATROL_HOME/log/PatrolAgent-%H-%P.audit

7.2.2 BPPM Server

The BMC ProactiveNet Server maintains audit logs of security related events.

Table 19 – Audit Log Entries for BPPM Server

Auditable Event	Audit Record Contents
Remote Actions	Actions performed by administrators on remote hosts, including the Integration Service.
Administrative changes	The AdminAudit.log logs administrative and configuration changes. This includes user actions such as Add, Edit and Delete user events.
Administrative changes made via the CLI	All actions are recorded and are readable by issuing the 'pw log list' command.

By default, audit logs are located at: <bppm>/pw/pronto/logs/<logfolder>. The main file for administrative logs is AdminAudit.log.

The audit log properties may be modified in the propertiesfile. This file may be used to set the size of the audit log file. The default is 5000 bytes, and while there is no predefined maximum size, BMC recommends that each log file not exceed 5 MB. Users may also set the number of audit log files that are cycled through during a rotation. After the specified number is reached, the cycle repeats itself, overwriting in sequence the log files of the previous cycle. The default is 1.

For example, if
auditLogFilename=AuditLog%g.log
auditLogFilecount=10
auditLogLimit=5000
then the initial audit log is assigned the name AuditLog1.file. When its file size reaches 5000 bytes, a new audit log is generated with the same name but

incremented by one: AuditLog2.file. As each log reaches the maximum size, a new audit log is created and incremented by one. When the maximum log file count (10 in this example) is reached, then the process repeats itself because only one cycle of logs is maintained. The first audit log of the new cycle starts at 1 (AuditLog1), overwriting the existing file. As new logs are generated in the new cycle, they overwrite the existing ones in sequence.

Whenever a BPPM Administrator manages the TOE through the CLI, these actions are recorded. The log records are accessible by issuing the 'pw log list' in the CLI.

The BMC ProactiveNet Server includes a logs directory (<bppm>/pw/pronto/logs and <bppm>/pw/pronto/logs/debug) where the logs for various functions are held. When debugging is configured, the logs/debug directory collects additional logs.

7.3 Access Control

Access to data in BMC ProactiveNet administrative features is controlled through Role-Based access controls. Users and User Groups are assigned one or more roles. Each configuration item has an ACL (Access Control List) detailing read and write permissions. Only users belonging to a group with read and/or write permissions for a Configuration Item are able to perform actions on that Configuration Item. For each role, individual administrative and operational features may be enabled or disabled, resulting in more fine grained access control to read and modify data. This is also called 'feature level access control'.

Within BMC ProactiveNet, a CI is any accessible resource. Access control is driven by the Read Level Security (RLS) and Write Level Security (WLS) settings on CIs, and the roles and groups associated with that CI. A CI is accessible to a user group only when the user group name is present in the BMC ProactiveNet defined RLS or WLS fields of the CI. If the user group name is present in the WLS field of the CI, then the user group has write access. If the user group name is present in the RLS field of the CI, then the user group has read access. If the user group name is not present in either of these fields of the CI, then the CI is not accessible to the user group. Such a CI is accessible only to admin users or users who have access to all CIs. Roles determine the permission users have to perform certain actions on accessible CIs. Roles are assigned to user groups, and user groups can have multiple roles. One role can be assigned to multiple user groups, giving identical permissions to different user groups.

Users may be added from the BMC ProactiveNet Administration console and associated to a user group. A user must be associated at least one user group. When users are assigned to a user group, they have access to the CIs defined in that user group. If the user group has access to a CI, the user has access to all devices associated with that CI.

Users may be added and associated with a user group from the BMC ProactiveNet Administration console. Users can be assigned to more than one user group, but must be associated with at least one user group in order to create the user. The available user groups are listed in the User Groups pane, which also allows for editing and deleting users. The User folder maintains user accounts, and allows an administrator to identify who has access to the BMC ProactiveNet system.

These access controls are applied any time a user accesses TOE resources through the Operations Console, the Central Monitoring Administration, the BMC ProactiveNet Administration Console, or the REST API. However, this is not enforced for the BMC ProactiveNet Command Line Interface, which provides setup and diagnostic capabilities.

7.4 Identification and Authentication

The BMC ProactiveNet server maintains the username, password and group information for individual users in an external Oracle database.

Users must log into the Operations Console, the Central Monitoring Administration, the BMC ProactiveNet Administration Console, the PATROL Configuration Manager and the REST API as one of these users. Users may log into the CLI and use the CLI or the Admin Console by logging into the server operating system. In the evaluated configuration, these interfaces may only be used locally, and are limited to the Linux root or Windows Administrator user.

7.5 Security Management

The TOE is administered via five interfaces accessed directly, or via a web browser. They are:

- The Operations Console: This is a web Interface accessible from https://server.domain.name, operated through a supported browser. This is the primary interface for managing the performance management functionality of the TOE. This interface is used by the BPPM Operator, BPPM Supervisor and BPPM Viewer role users to perform the functions required to configure events and event relationships, to view probable cause information and generate reports. Users of this interface may be restricted to administer only particular parts of the monitored network (i.e. certain PATROL Agents) based on Group membership. Users of this interface work with events and event relationships to help identify the probable cause of network issues. The functionality provided by this console is described in the BMC ProactiveNet User Guide.
- Central Monitoring Administration (CMA): Also known as the CMA Console, this is a web Interface accessible from https://server.domain.name/admin. It is used by the BPPM Administrator, BPPM Model Administrator, and BPPM Monitoring Administrator to configure PATROL agents and verify their connection status, and to manage the monitoring policies employed in the network, including setting thresholds for determining abnormalities. The CMA web application is hosted on the BPPM server. The functionality provided by this interface is described in BMC CMA Administering-v57-20131105_1528.
- BMC ProactiveNet Administration Console: This is a legacy administration console. Most of the functionality of this interface is also available on the CMA. There are some key administrative features that are only available from this interface, including the administration of users, user groups and roles. This console is used by the BPPM Administrator to perform some user management and some configuration tasks for the Integration Servers. This interface is described in the BMC ProactiveNet Administrator Guide. In the evaluated configuration, this console is installed on the BPPM Server and accessed directly.
- The BMC ProactiveNet CLI is a collection of BMC ProactiveNet and Event Management commands that may only be run from the BMC ProactiveNet server as the root (on Linux) or the Administrator (on Windows). The CLI provides an alternative to the Administrative Console for setup operations, and provides diagnostics capabilities. This interface is described in the BMC ProactiveNet Command Line Interface Reference Guide. Use of SSH or Remote Desktop to access this CLI remotely is not included in the evaluated configuration.
- REST API: This is a programmatic interface hosted by the server. It may be used in place of the CLI and is documented in BMC ProactiveNet Performance Management Web Services Reference Guide. BMC ProactiveNet REST-based web service APIs can be used to retrieve metadata, configuration data, and statistical data for specific BMC ProactiveNet- monitored resources. It may be used in place of the CLI for some functions. The REST API is documented in the BMC ProactiveNet Performance Management Web Services Reference Guide.

The TOE supports the following roles: BPPM Administrator, BPPM Model Administrator, BPPM Monitoring Administrator, BPPM Operator, BPPM Supervisor, BPPM Viewer, and BPPM WS Full Access. These are known as default user groups in the BPPM documentation, and each group is associated with one or more roles to determine the available permissions. Finally, users are put into user groups and are granted the permissions associated with the group.

7.6 Performance Management

The TOE consolidates data and events from the PATROL Agents. Using this information, the TOE learns the behavioral and performance trends for each of the monitored components and identifies normal and abnormal behavior. Metrics that are determined to be of particular importance are designated as key performance indicators. This information is provided to the analytics engine, which then generates events when there is significantly abnormal behavior. Based on these learned trends, critical events are identified. These events point to the root cause of network issues and may then be examined by the users to correct the problem causing the network issue.

Abnormalities are generated when the data values from a monitor fall outside of the normal baseline range for a statistically significant number of points within the sample window specified in the signature threshold. The abnormalities generated have a severity field, which records which baseline was exceeded for a significant number of points (helping determine the severity of the abnormality). Generally, abnormalities that exceed Weekly baseline are more serious in nature than ones that exceed Daily baseline, and abnormalities that exceed Daily baseline are more serious in nature than ones that exceed Hourly baseline. By default, if no signature threshold is set, abnormalities are still generated. A minimum of five data points are required. If the sample window is set too small (sample window is set in time duration), the algorithm will still automatically wait for five data points to come in. For example, if you set a sample window to 10 minutes on a specific monitor attribute but the polling rate of that monitor is 5 minutes, 25 minutes must pass before an abnormality is generated. For this reason, it is better to use lower polling rates for monitors. Abnormalities are closed when the number of data points exceeded in the last window sample size is not considered significant. For example, if six out of seven data points out of range are statistically significant, then as soon as the last exceeded points drop to five out of seven points, the abnormality will be closed. By default, even if no explicit global or instance signature threshold is set, BMC ProactiveNet will generate abnormalities for these baseline conditions.

Event collectors gather events for display in an event list. They provide operators with meaningful groups of events or abnormalities. Relationships may be established between events by defining rules and policies. In addition to the normal events generated after a problem arises, BMC ProactiveNet can also generate predictive events. Predictive events are early warning events that BMC ProactiveNet generates before an event condition occurs on an existing metric. Such events facilitate the identification of potential problems before the condition causes impacts to business services and before the end users notice the problem. The typical predictive window is three hours or less, which means most predictive events are generated up to three hours before an actual event condition would occur. Predictive events target persistent anomalies that might go undetected for most of the day (during the off-peak periods) but become serious issues at peak load periods. These kinds of problems are typically caused by configuration changes or user load shifts. For example, if a change is made to a load balancer that accidentally redirects more end-user transactions to one of the servers in the pool, that server might now have much more load than the other servers in the pool. However, this would not be obvious during low traffic periods. As the peak load period approaches, the imbalance would become more serious and the server load issue would become obvious, eventually causing an event on an end user transaction and performance or availability issues for all transactions routed through that server. In this scenario where BMC ProactiveNet events are used, an abnormality is generated as soon as the first change in behavior is detected, a predictive event would be generated two to three hours before any impact or before a real event condition. Predictive events are enabled through the absolute threshold settings. The predictive algorithm leverages the hourly baseline in conjunction with the threshold setting to determine when to create a predictive event.

By gathering data from different sources and applying filters to rule out unrelated events, BMC ProactiveNet can determine the most-likely causes for an event, such as an attribute that is outside the desired range. This process of gathering and filtering data to determine the cause for an event is called probable cause analysis. The probable cause analysis process analyzes data and displays the relevant events automatically. The accuracy of probable cause analysis is increased by providing relationships between devices and monitors. The more data that is provided about an event, the more accurate probable cause analysis will be. Probable cause analysis focuses on events that are able to impact other events in unexpected ways.

The resulting analysis may be viewed through the Operations Console. Reports may be generated showing various levels of detail.


bmcsoftware
BUSINESS RUNS ON I.T.™