National Information Assurance Partnership

Common Criteria Evaluation and Validation Scheme
Validation Report

BMC Software PATROL®
Version 3.4.11

Report Number: CCEVS-VR-02-0024
Dated: 30 September 2002
Version: 1.01

National Institute of Standards and Technology
Information Technology Laboratory
100 Bureau Drive
Gaithersburg, MD 20899

National Security Agency
Information Assurance Directorate
9800 Savage Road STE 6740
Fort George G. Meade, MD 20755-6000

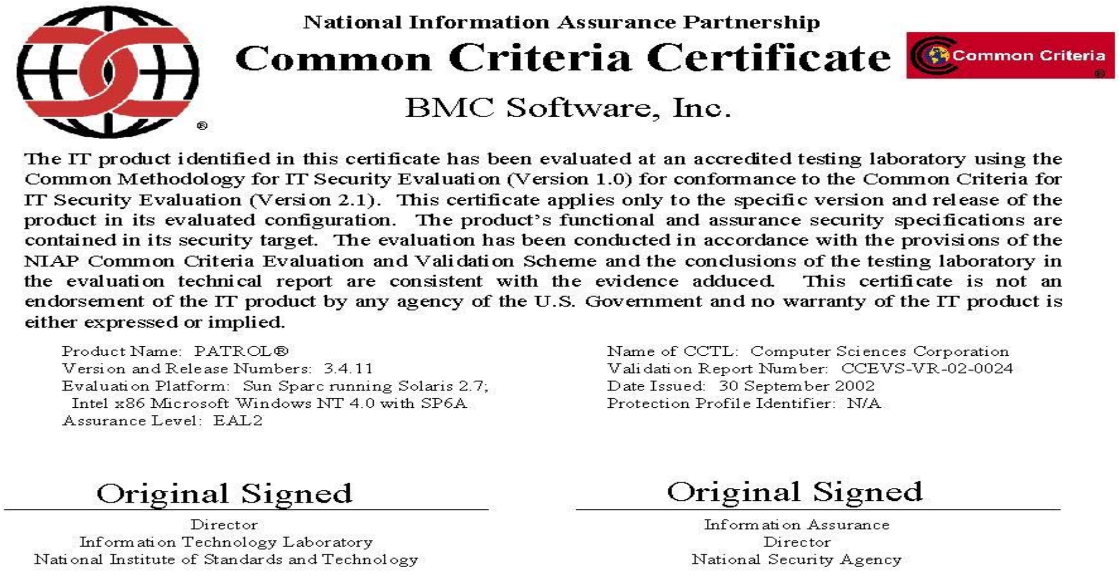
ACKNOWLEDGEMENTS

Validation Team

David A. Wheeler
William R. Simpson
Institute for Defense Analyses
Alexandria, Virginia

Common Criteria Testing Laboratory

Computer Sciences Corporation
Annapolis Junction, Maryland



National Information Assurance Partnership

Common Criteria Certificate

BMC Software, Inc.

The IT product identified in this certificate has been evaluated at an accredited testing laboratory using the Common Methodology for IT Security Evaluation (Version 1.0) for conformance to the Common Criteria for IT Security Evaluation (Version 2.1). This certificate applies only to the specific version and release of the product in its evaluated configuration. The product's functional and assurance security specifications are contained in its security target. The evaluation has been conducted in accordance with the provisions of the NIAP Common Criteria Evaluation and Validation Scheme and the conclusions of the testing laboratory in the evaluation technical report are consistent with the evidence adduced. This certificate is not an endorsement of the IT product by any agency of the U.S. Government and no warranty of the IT product is either expressed or implied.

Product Name: PATROL®
Version and Release Numbers: 3.4.11
Evaluation Platform: Sun Sparc running Solaris 2.7;
Intel x86 Microsoft Windows NT 4.0 with SP6A
Assurance Level: EAL2

Name of CCTL: Computer Sciences Corporation
Validation Report Number: CCEVS-VR-02-0024
Date Issued: 30 September 2002
Protection Profile Identifier: N/A

Original Signed
Director
Information Technology Laboratory
National Institute of Standards and Technology

Original Signed
Information Assurance
Director
National Security Agency

1. EXECUTIVE SUMMARY

An evaluation of the BMC Software, PATROL®, Version 3.4.11, was begun 5 March 2001 and completed 17 September 2002. PATROL® is a distributed systems application and event management tool. It provides an environment for monitoring the status of resources (such as operating systems and database servers) in a distributed environment. Users may use the PATROL® Console (client) to connect to multiple PATROL® Agents (servers). An Agent runs on each machine being monitored, running scripts called "Knowledge Modules" that collect information from that machine's resources (e.g., operating system or database server status); the Agent can then report summary information back to user Consoles.

Computer Sciences Corporation (CSC) performed the evaluation in the United States. The evaluation was carried out in accordance with requirements drawn from the Common Criteria CCv2.1, Part 3 for EAL2 [CC_PART3] and Common Evaluation Methodology for Information Technology Security – Part 2: Evaluation Methodology [CEM_PART2] extended, with guidance and interpretations provided by the U.S. evaluation scheme. The assurance activities in this assurance level offer confidence that the BMC Software, PATROL® Version 3.4.11 (with documentation and software deliverables as defined in sections 6 and 8, respectively) contains requirements that are:

- Justifiably included to counter stated threats and meet realistic security objectives,
- Internally consistent and coherent,
- Technically sound, and
- Free from vulnerabilities associated with obvious and publicly known threats.

Computer Sciences Corporation, a Common Criteria Testing Laboratory [CCTL], is accredited by the National Voluntary Laboratory Accreditation Program (NVLAP) and approved by the NIAP Common Criteria Evaluation and Validation Scheme [CCEVS]to perform Common Criteria evaluations. The CCTL has presented CEM work units and rationale that are consistent with the CC, the CEM and CCEVS publication number 4 Guidance to CCEVS Approved Common Criteria Testing Laboratories [CCEVS_PUB 4]. The CCTL team concluded that the requirements of the EAL 2 have been met. Therefore, a **pass** verdict has been issued, by the CCTL, for the BMC Software, PATROL® Version 3.4.11.

The information contained in this Validation Report is not an endorsement of PATROL® by any agency of the U.S. Government and no warranty of the product is either expressed or implied.

Evaluation Highlights

Dates of Evaluation:	5 March 2001 – 30 September 2002
Evaluated Product:	PATROL®, Version 3.4.11
Security Target:	BMC Software PATROL Version 3.4.11 Security Target Version 1.0, September 13, 2002
Developer:	BMC Software Inc., 2101 Citywest Boulevard, Houston, TX 77042
CCTL:	Computer Sciences Corporation
Evaluation Class:	EAL2
PPs Claimed:	None.
Validation Team:	David A. Wheeler, Institute for Defense Analyses William R. Simpson, Institute for Defense Analyses (backup)
Version of CC:	Common Criteria version 2.1, August 1999
Version of CEM:	Common Evaluation Methodology 1.0, August 1999
Effective Date for Interpretations:	All interpretations as of 5 March 2001

2. PRODUCT IDENTIFICATION

The Target of Evaluation (TOE) is BMC Software, PATROL ®, Version 3.4.11. It consists of the following components:

- PATROL® Console for UNIX,
- PATROL® Event Manager (UNIX),
- PATROL® Agent for (UNIX),
- PATROL® Console for Microsoft Windows 2000,
- PATROL® Agent for Microsoft Windows 2000,
- PATROL® KM for UNIX V8.3, and
- PATROL® KM for NT V.3.5

The product must run on top of hardware and an operating system. The TOE has been evaluated on the following platforms (one Unix, one Windows):

PATROL® Console/Agent for UNIX™:

- SUN SPARC-based platform running Solaris 2.7

PATROL® for Microsoft Windows 2000 Server:

- Intel x86-based platform capable of running Microsoft Windows NT 4.0, SP 6a

The product also requires certificates; default certificates are provided with the product, but they are not part of the evaluated configuration (since they should not be used in a production environment). Thus, certificates must be generated using some process outside the scope of the evaluation (Microsoft IIS was used to generate the certificates in the evaluation).

Note that the evaluated configuration is a special configuration that, after purchase, must be installed and configured by the vendor at the customer's premises; this evaluation does not apply to the "standard" product that can be purchased and directly installed by customers. This special configuration includes in it an additional package specifically to improve security, namely Security Pack for PATROL® for Unix 8.3.04 (for Unix) or the Security Pack for PATROL® Microsoft Windows 2000 Servers 2.1.01 (for Windows).

Also note that informally this product is sometimes referred to as "PATROL® Classic," to differentiate it from other products such as PATROL® Perform/Predict and PATROL® Enterprise Manager.

3. SECURITY POLICY

The TOE with support from its IT environment provides the following security functions:

- Auditing,
- User Data Protection,
- Identification and Authentication,
- Security Management, and
- Protection of Security Functions.

The ST defines this in greater detail.

4. ASSUMPTIONS AND CLARIFICATION OF SCOPE

4.1 Usage Assumptions

The following usage assumptions were made for the TOE:

- The evaluated configuration is a special configuration that, after purchase, is installed and configured by the vendor at the customer's premises; this evaluation does not apply to the "standard" product that can be purchased and directly installed by customers.
- Only two Knowledge Modules (KMs) were included in the evaluation: PATROL® KM for UNIX V8.3 and PATROL® KM for NT V.3.5 (to monitor Unix and Windows respectively). Many other KMs are available but they are not part of the TOE; the impact of adding other KMs was not evaluated and thus will take the TOE out of its evaluated configuration.

4.2 Environmental Assumptions

Since the operating system and certificate authority are not included in the evaluation, they must be separately secured.

4.3 Clarification of Scope

Note the following:

- This is a limited security functionality product evaluated at EAL2. As with any evaluation, this evaluation only shows that the evaluated configuration meets the security claims made, with a given level of assurance.
- As with all EAL2 evaluations, this evaluation did not search for, nor seriously attempt to counter, vulnerabilities that were not "obvious" (as this term is defined in the CC and CEM), or so-called "vulnerabilities" to objectives not claimed in the ST. For example, the evaluation does not claim a resistance of the TOE to denial-of-service attacks.
- The ST asserts that one controlled capability is "launching a Console in developer mode"; this requirement is interpreted as gaining the privileges implied by developer mode (e.g., changing Agent KMs). Although some effort was made to make it difficult for naïve users to make a Console display as if it is in developer mode, simply causing a display to claim it is in developer mode is not the same as actually being in developer mode (after all, someone could write another program that created the same display). The critical issue here is *privilege*, and in the evaluated configuration the Agents perform I&A (using passwords sent over an encrypted channel, to determine if the user may perform such actions.

Specific threats to IT security that should be countered by the BMC Software, PATROL® version 3.4.11, can be grouped into threats against the system as a whole, threats against the Consoles (clients), and threats against the Agents (servers).

First, here are the threats the TOE, as a whole must counter (as identified in the ST):

Name	Description
T.REPLAY	A hostile/unauthorized user would use replay to obfuscate unauthorized activity.
T.TRAFFIC_SPOOF	A hostile/unauthorized user would attempt to spoof Agent communications in order to hide or perform unauthorized activity, or provide false data.
T.TROJAN	A hostile/unauthorized user will attempt to use the PATROL□ Scripting Language "files create" as a mechanism to get file access.
T.UNAUTH_ACCESS_DATA	A hostile/unauthorized user would attempt to read TOE data/configuration files in order to: • Ascertain TOE, or managed application, secrets. • Modify TOE behavior.
T.UNDETECTED_ACTIONS	Authorized and unauthorized users would use the fact that identification and recording of their actions was not taking place in order to circumvent the TSP.

Here are the threats the Console (client) must counter:

Name	Description
TC.UNAUTH_DEPLOY	A hostile/unauthorized user would attempt to deploy an unauthorized KM(s) on a remote platform to change/modify/attack the "system"/system management.
TC.UNAUTH_CHANGES	A hostile/unauthorized user would attempt to make unauthorized changes to the Agent and KM configuration to change/modify/attack the "system"/system management.
TC.UNAUTH_COMMANDS	A hostile/unauthorized user would attempt to execute unauthorized system commands on the target system to change/modify/attack the "system"/system management.

Here are the threats the Agents (servers) must counter (the subscript “R” is short for “Remote,” but the agent may be in the same physical facility or even on the same machine as the Console):

Name	Description
TR.ELEVATE_ACCESS	A hostile/unauthorized user may attempt to bypass the security of the TOE through attempting to use the PATROL□ Agent to elevate access to remote machines.
TR.APPLICATION_SECRETS	A hostile/unauthorized user will attempt to access Agent configuration/data files in order to obtain secrets (e.g., passwords) to monitored applications in order to gain unauthorized access to those applications.
TR.KM_TAMPER	A hostile/unauthorized user will attempt to modify Agent and/or KM behavior by making unauthorized changes to KM script files to modify TOE behavior, or gain unauthorized access.

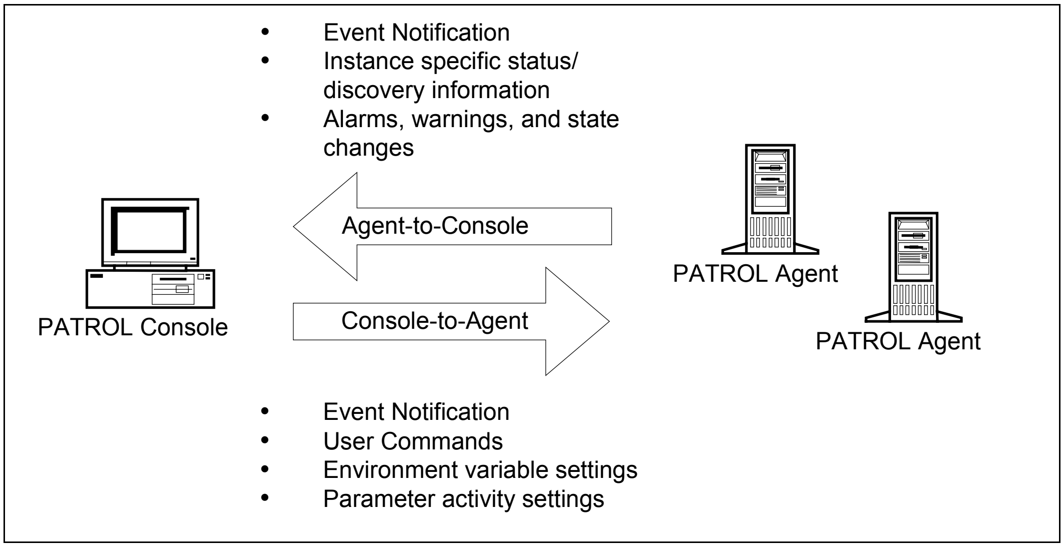
5. ARCHITECTURAL INFORMATION

PATROL® is a systems application and event management tool. It provides an environment by which the status of every vital resource in the distributed environment being managed can be monitored. Users may use the PATROL® Console (client) to connect to multiple PATROL® Agents (servers). An Agent runs on each machine being monitored, running scripts called "Knowledge Modules" that collect information from that machine's resources (e.g., operating system or database server status); the Agent can then report summary information back to user Consoles. PATROL® is a suite of products consisting of:

- PATROL® Console,
- PATROL® Agents,
- PATROL® Event Manager (PEM), and
- PATROL® Knowledge Modules (KMs), which are executed by the Agents.

In the context of PATROL®, applications are any resource used by, or running on, a computer.

The following diagram shows, at a high level, the information flows between the PATROL® Console and PATROL® Agent components:


• Event Notification
• Instance specific status/ discovery information
• Alarms, warnings, and state changes
Agent-to-Console
PATROL Console
Console-to-Agent
PATROL Agent
PATROL Agent
• Event Notification
• User Commands
• Environment variable settings
• Parameter activity settings

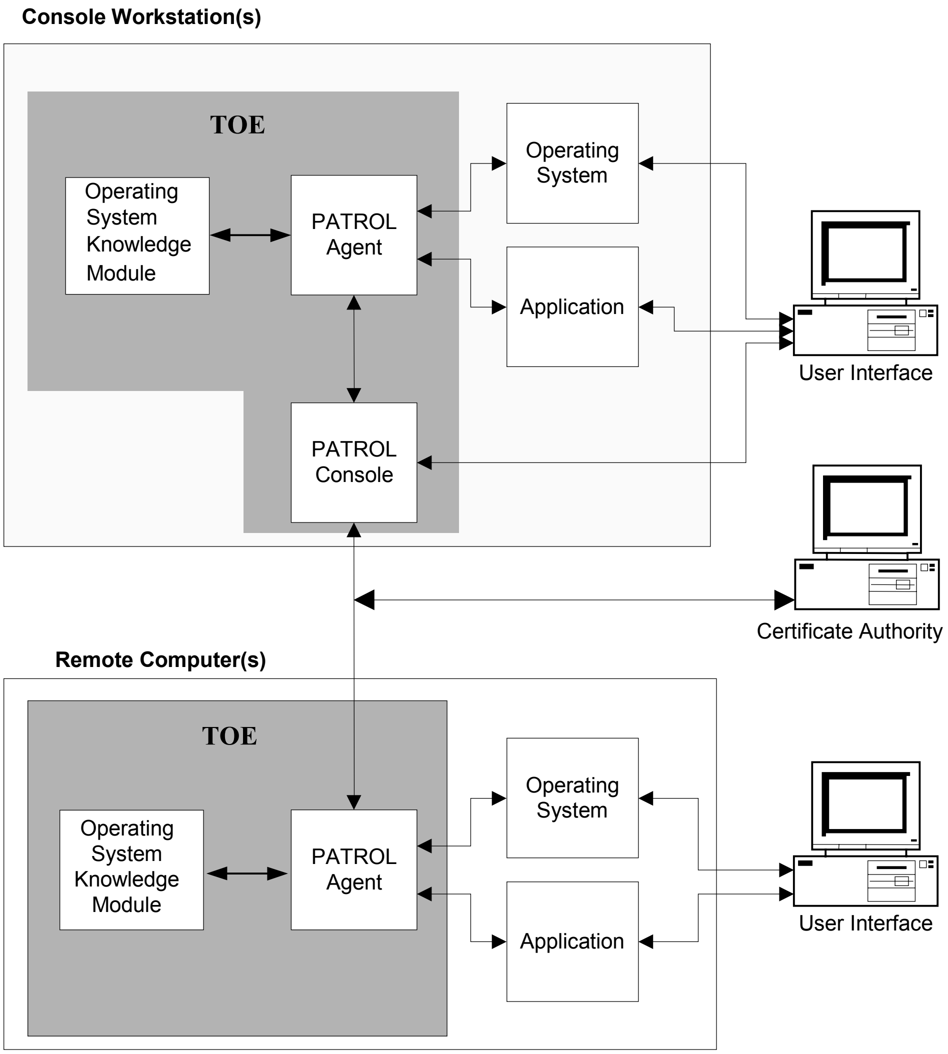
Interactions between the Console and Agent

The TOE configuration essentially consists of two types of subsystems:

- The PATROL® Console Workstation (running the Console and PEM), and
- One or more PATROL® Agents that execute KMs on remote computer platform(s) functioning as either a workstation or server.

The Console workstation executes the PATROL® Console and PATROL® Agent. For UNIX™ workstations the PATROL® Event Manager is an additional sub-component that is instantiated on the PATROL® Console Workstation; under Windows 2000™, the PATROL® Event Manager is integral to the PATROL® Console applications. The remote server(s)/workstation(s) execute the PATROL® Agent(s). PATROL® Operator Console and the PATROL® Developer Console are the graphical workspaces from which commands are issued to manage the distributed environment monitored by PATROL®. The PATROL® Console displays all of the monitored computers and applications. The PATROL® Console can work in two console modes: Operator Console and Developer Console.

The following figure illustrates this, as well as showing their relationship to other environmental component (in particular the operating system, applications, certificate authority, and network):



With the PATROL® Operator Console the following tasks can be performed:

- Define which applications PATROL should monitor ,
- Monitor and manage computers and applications through the PATROL®,
- Agent and PATROL® Knowledge Modules,
- Monitor the PATROL® Agent's use of resources,
- Run predefined or user-defined commands and tasks against monitored machines,
- Run state change action commands on the PATROL® Console machine when a state change occurs on a monitored computer,
- Log on to any managed computer (only for Unix and OpenVMS.),
- Start and stop PATROL® Agents remotely,
- View parameter data, and
- Retrieve historical data stored by the PATROL Agent.

The PATROL® Developer Console in the evaluated configuration is used only in the installation and initial start-up. The PATROL® Developer Console is responsible for the following restricted Security Management activities:

- Committing PATROL® KM changes to a PATROL® Agent (Changes to the KM result in an unevaluated configuration.);
- Issuing operating system commands at the PATROL® system output window (Outside scope of evaluated configuration.);
- Modifying the PATROL® Agent's parameter attributes (Outside scope of evaluated configuration.); and
- Launching a PATROL® Console in developer mode. (Outside scope of evaluated configuration.)

PATROL Agent is the core piece of the PATROL® 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 PATROL® consoles.
- Loads specified Knowledge Modules (KMs) at start-up runs menu commands, and updates InfoBoxes in the PATROL® Console.
- Acts as a service provider for event management.

The PATROL® Event Manager (running on a PATROL® Console) is the
component by which the following tasks can be performed:

- View events,
- Manage events and use events to control the managed environment,
- Trigger events,
- Generate event statistics,
- Acknowledge events, and
- Close events.

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. The evaluation only focused on the UNIX and Windows NT OS KMs. No other KMs were tested during the course of this evaluation. (Changes to the KM will result in an unevaluated configuration.) PATROL® KMs provide information to the PATROL® Agent about:

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

Physically, each TOE platform consists of a processor architecture appropriate for the Operating System on which the TOE component runs. The TOE does not include any physical network components between the adapters of a connection between platforms. The ST assumes that any network connections, equipment (e.g., routers), and cables are appropriately protected in the TOE security environment.

A proprietary PATROL communications protocol is used between the console and agent. In the evaluated configuration, this PATROL protocol is used to encapsulate the SSL protocol. The SSL protocol is used to provide encryption between the console and agent. SSL keys are generated outside of the TOE, stored as certificates, and are distributed using means outside the TOE (typically keys are distributed manually during installation time through means such as a floppy disk). Authentication of hosts is done through confirmation of both the certificates and the IP address. Authentication of users is done through passwords sent through this encrypted link.

6. DOCUMENTATION

The documentation provided with the product is as follows:

[BPC_ARM]	PATROL® Agent Reference Manual, Version 3.4
[BPC_CGS]	PATROL® Console for Unix Getting Started, Version 3.4
[BPC_CUG]	PATROL® Console for Unix User Guide, Version 3.4
[BPC_KMU]	PATROL® Knowledge Module™ for UNIX User Guide, Version 8.3
[BPC_STB]	PATROL® Security Technical Bulletin, May 13, 2002
[BPC_URN]	PATROL® for Unix Release Notes, Version 8.3.04
[BPC_WRN]	PATROL® for Microsoft Windows 2000 Servers Release Notes, Version 2.1.01
[BPC_WUG1]	PATROL® for Windows 2000 User Guide, Volume 1
[BPC_WUG2]	PATROL® for Windows 2000 User Guide, Volume 2
[BPC_WUG3]	PATROL® for Windows 2000 User Guide, Volume 3

Note that PATROL® Packaging and Delivery Procedures, September 27, 2001 [BPC_PDG] are not delivered to the customer, but instead are applied by vendor personnel to deliver and install the system. For a longer list of the major pieces of evidence examined during the evaluation, see section 14 of this report.

7. IT PRODUCT TESTING

7.1 Examination of Vendor Tests

The vendor provided test plans, procedures, test results and a test coverage document. The evaluator examined the test coverage analysis and found that the vendor provided a correspondence between the tests provided for evaluation and the functional specification (once an issue involving the reporting software was explained).

The evaluator found many tests in the test documentation that were not addressed by the test coverage analysis nor did these tests verify TOE functionality; since these tests did not correspond to the TOE, the evaluator did not consider these tests evidence. Nevertheless, once the irrelevant tests were discounted, the evaluator determined that the vendor tested (at a high level) most security-relevant aspects of the product. The evaluator was able to devise a test sample based upon the vendor's test coverage analysis and administration/user guidance.

The evaluator determined that the developer's tests were sound in their approach. The test document provided the configuration of the test hardware and software, the objective for each of the tests, and test procedures. The information provided was adequate to be able to reproduce the tests. The evaluators determined that the developer's approach to testing the TSF was appropriate for this EAL2 evaluation. The developer tested aspects of the security kit and concentrated on the operation of the product with the security patch installed.

7.2 Evaluator Independent Tests

The evaluator used the evaluation evidence provide as the test basis for the independent testing effort. The evaluator analyzed [BPC_ST] and determined that the primary purpose of the TOE was to provide protection of the data between the console and the agent. To especially examine this issue, the evaluator devised four test areas around the SFRs allocated to the User Data Protection TSF to especially examine this issue, and then devised tests in these areas:

1. Secure Socket Layer Protocol,
2. Host-based Access Control,
3. User-based Access Control, and
4. Security Management.

The evaluator took advantage of the nature of the Audit TSF to verify aspects of FAU_GEN.1 during tests. The evaluator used a combination of manual and inspection techniques to execute these tests. In all cases, the evaluator's expected results matched the actual results.

The configuration used for testing consisted of an Ethernet network with five systems. These systems were the Console Workstation and Agent on each of Sun Solaris and Microsoft

Windows platforms, plus a packet sniffer used for testing. More specifically, the configuration was as follows:

1. PATROL® Console Workstation version 3.4.11 running on a Sun Workstation with Sun Solaris 2.7.
2. PATROL® Agent version 3.4.11 running on a Sun Workstation with Sun Solaris 2.7.
3. PATROL® Console Workstation 3.4.11 running on a generic PC with Microsoft Windows NT version 4.0.
4. PATROL® Agent version 3.4.11 running on a Sun Workstation with Sun Solaris 2.7 with Microsoft Windows NT version 4.0.
5. A network analyzer and packet sniffer (implemented using a generic PC running Red Hat Linux 7.2, Ethereal 0.8.18, and IP Sorcery 1.4.3). Note that this is not actually part of the TOE, but is instead simply a tool used to examine network traffic.

7.3 Strength of Function

Even if a TOE security function cannot be bypassed, deactivated, or corrupted, it may still be possible to defeat it because there is a vulnerability in the concept of its underlying security mechanisms. For those functions a qualification of their security behavior can be made using the results of a quantitative or statistical analysis of the security behavior of these mechanisms and the effort required to overcome them. The qualification is made in the form of a strength of TOE security function claim. The overall SOF claim for the TOE made in ST is expressed as an SOF rating, SOF-basic. The vendor provided calculations supporting their claim that the TOE met this rating. The evaluation team believed that the vendor calculations and assumptions were pessimistic, but since any change would only strengthen the claim that it met the rating SOF-basic, the evaluators determined that the claim of SOF-basic was justified.

7.4 Vulnerability Analysis

Vulnerability analysis is an assessment to determine whether vulnerabilities identified, during the evaluation of the construction and anticipated operation of the TOE or by other methods (e.g. by flaw hypotheses), could allow users to violate the TSP.

The evaluation team examined the vendor's vulnerability analysis [BPC_VLA] and found that the vendor considered all of the evaluation deliverables: the ST, functional specification, high-level design, user guidance, administrator guidance, secure installation, generation, and start-up procedures, vulnerability analysis, and the strength of function claims analysis when developing [BPC_VLA]. The evaluation team found that the developer consulted the following web sites:

- http://www.cert.org,
- http://www.SecurityFocus.com,
- http://www.SecuriTeam.com,

- http://xforce.iss.net, and
- http://www.rootshell.com

[BPC_AVA] states that the threat agent is considered to possess a low attack potential such that their level of expertise is that of a layman, possessing no sophisticated tools, and only public knowledge of the TOE. This is consistent with the threat described in [BPC_ST].

The evaluation team devised a penetration test plan around three test areas: Network, Environment, and Advanced Attacks. Network attacks concentrated or attacking the agent from the network. Environment Attacks attempted to exploit the agent and console as an untrusted user from the keyboard of the host system. Advanced Attacks attempted to exploit the TOE using attack methods and techniques that are not part of the TOE environment (the evaluation team tried a UDP bounce attack as well as a "Shatter" attack based on an August 2002 paper by Chris Paget).

The evaluators did not identify a vulnerability that violated the TOE security policy given its environment and threats. Therefore, the evaluators determined that the product met the criteria of EAL2 for vulnerability analysis (in particular, that "the evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure obvious vulnerabilities have been addressed").

9. RESULTS OF THE EVALUATION

The validation team followed the procedures outlined in the Common Criteria Evaluation Scheme [CCEVS] publication number 3 for Technical Oversight and Validation Procedures. [CCEVS_PUB 3]. The validation team has observed that the evaluation and all of its activities were in accordance with the Common Criteria, the Common Evaluation Methodology [CEM], and the CCEVS. The validation team therefore concludes that the evaluation and its results of **pass** are complete and accurate.

9.1 Assurance Content

The evaluation provides for Assurance at the EAL 2 level without augmentation. Therefore, this includes the assurance components as shown in the table below:

EAL2 Assurance Requirements

Assurance Class	**Assurance Family**
ST Evaluation	ASE_DES.1
	ASE_ENV.1
	ASE_INT.1
	ASE_OBJ.1
	ASE_PPC.1
	ASE_REQ.1
	ASE_SRE.1
	ASE_TSS.1
Configuration Management	ACM_CAP.2
Delivery and Operation	ADO_DEL.1
	ADO_IGS.1
Development	ADV_FSP.1
	ADV_HLD.1
	ADV_RCR.1
Guidance Documents	AGD_ADM.1
	AGD_USR.1
Tests	ATE_COV.1
	ATE_FUN.1
	ATE_IND.2
Vulnerability Assessment	AVA_SOF.1
	AVA_VLA.1

10. VALIDATOR COMMENTS/RECOMMENDATIONS

As with any evaluation, this evaluation shows that the evaluated configuration meets the security claims made, with a certain level of assurance. It is worth noting that the evaluated configuration is a special configuration that, after purchase, is installed and configured by the vendor at the customer's premises; this evaluation does not apply to the "standard" product that can be purchased and directly installed by customers. The product has been evaluated at the assurance level of EAL 2 that it meets its functional claims.

Be sure to note the assumptions and clarifications of scope in section 4 of this report.

The validator observed that the evaluation and all of its activities were in accordance with the CC the CEM, and CCEVS practices; and that the CCTL presented appropriate CEM work units and rationale. The validation team therefore concludes that the evaluation, and its results of **pass,** are complete.

LIST OF ACRONYMNS AND GLOSSARY OF TERMS

The following acronyms are provided for reference:

CC	Common Criteria
CCEL	Common Criteria Evaluation Laboratory
CCEVS	Common Criteria Evaluation and Validation Scheme
CCTL	Common Criteria Testing Laboratory
CEM	Common Evaluation Methodology
CI	Configuration Items
CSC	Computer Sciences Corporation
DSA	Developer Security Analyst
EAL	Evaluation Assurance Level
EDR	Evaluation Discovery Report
ETR	Evaluation Technical Report
MRA	Mutual Recognition Arrangement
NIAP	National Information Assurance Program
NIST	National Institute of Standards & Technology
NSA	National Security Agency
OR	Observation Report
PP	Protection Profile
SAR	Security Assurance Requirement
SFR	Security Functional Requirements
SOF	Strength of Function
ST	Security Target
TCSEC	Trusted Computer Systems Evaluation Criteria
TOE	Target of Evaluation
TSC	TOE Scope of Control
TSF	TOE Security Functions
TSFI	TSF Interface

The following terms are provided for reference:

User Any entity (human user or external IT entity) outside the TOE that interacts with the TOE.

Human user Any person who interacts with the TOE.

Authorized User A user that, in accordance with the TOE Security Policy (TSP) may perform an action. (As identified by group membership.)

External IT entity Any IT product or system, untrusted or trusted, outside of the TOE that interacts with the TOE.

Role A predefined set of rules establishing the allowed interactions between a user and the TOE.

Identity A representation (e.g., a string) uniquely identifying an authorized user, which can be either the full or abbreviated name of that user or a pseudonym.

Authentication data Information used to verify the claimed identity of a user.

Collection Process A TOE process that collects pre-defined data for a pre-defined period of time, and results in data that is re-formatted into UDR format for use by the Manager, Predict, Analyze, and Visualizer components of the TOE.

DOCUMENTATION

The evidence used in this evaluation is based upon the product and the following documentation:

[BPC_ARM] PATROL® Agent Reference Manual, Version 3.4, Chapters 3, 5, 8, 9, and 10

[BPC_CGS] PATROL® Console for Unix Getting Started, Version 3.4

[BPC_CUG] PATROL® Console for Unix User Guide, Version 3.4

[BPC_FSP] BMC Software, PATROL® Classic Version 3.4.11, Functional Specification, Version 1.8

[BPC_HLD] BMC Software, PATROL® Classic Version 3.4.11, High-Level Design Document, Version 1.8

[BPC_KMU] PATROL® Knowledge Module™ for UNIX User Guide, Version 8.3

[BPC_SOF] BMC Software PATROL® Version 3.4.11 Strength of Function Analysis, Version 1.0, June 26, 2002

[BPC_ST] BMC Software, PATROL® Version 3.4.11, Security Target, Version 1.0

[BPC_STB] PATROL® Security Technical Bulletin, May 13, 2002

[BPC_TLS] Spyrus TLS Gold™ Toolkit Version 4.0 Programmer's Guide

[BPC_URN] PATROL® for Unix Release Notes, Version 8.3.04

[BPC_WRN] PATROL® for Microsoft Windows 2000 Servers Release Notes, Version 2.1.01

[BPC_WUG1] PATROL® for Windows 2000 User Guide, Volume 1

[BPC_WUG2] PATROL® for Windows 2000 User Guide, Volume 2

[BPC_WUG3] PATROL® for Windows 2000 User Guide, Volume 3

[BPC_PDG] PATROL® Packaging and Delivery Procedures, September 27, 2001

[BPC_EML] Email from Michael Crane on October 10,2001, that states the Security Pack for PATROL® for Unix 8.3.04 and the Security Pack for PATROL® Microsoft Windows 2000 Servers 2.1.01 is always delivered

[BPC_AU] PATROL® Classic Administrator/User Guidance Mapping

The evaluation and validation methodology was drawn from the following:

[CC_PART1] Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and general model, dated August 1999, version 2.1.

[CC_PART2] Common Criteria for Information Technology Security Evaluation – Part 2: Security functional requirements, dated August 1999, version 2.1.

[CC_PART2A] Common Criteria for Information Technology Security Evaluation – Part 2: Annexes, dated August 1999, version 2.1.

[CC_PART3] Common Criteria for Information Technology Security Evaluation – Part 3: Security assurance requirements, dated August 1999, version 2.1.

[CEM_PART1] Common Evaluation Methodology for Information Technology Security – Part 1: Introduction and general model, dated 1 November 1997, version 0.6.

[CEM_PART2] Common Evaluation Methodology for Information Technology Security – Part 2: Evaluation Methodology, dated August 1999, version 1.0.

[CCEVS_PUB 1] Common Criteria, Evaluation and Validation Scheme for Information Technology Security, Organization, Management and Concept of Operations, Scheme Publication #1, Version 2.0, May 1999.

[CCEVS_PUB 2] Common Criteria, Evaluation and Validation Scheme for Information Technology Security, Validation Body Standard Operating Procedures, Scheme Publication #2, Version 1.5, May 2000

[CCEVS_PUB 3] Common Criteria, Evaluation and Validation Scheme for Information Technology Security, Technical Oversight and Validation Procedures, Scheme Publication #3, Version 1.0, January 2002.

[CCEVS_PUB 4] Common Criteria, Evaluation and Validation Scheme for Information Technology Security, Guidance to CCEVS Approved Common Criteria Testing Laboratories, Scheme Publication #4, Version 1, March 20, 2001

[CCEVS_PUB 5] Common Criteria, Evaluation and Validation Scheme for Information Technology Security, Guidance to Sponsors of IT Security Evaluations, Scheme Publication #5, Version 1.0, 31 August 2000.