CloudShield CS-2000 with CPOS 3.0.3 Security Target

Version: 1.0

Last Update: 2012-01-25

This document is provided AS IS with no express or implied warranties. Use the information in this document at your own risk.

Table of Content

Document History

Version	Date	Changes	Author
0.1	2008-08-25	Initial Version	Stephan Müller, atsec
0.2	2008-09-15	Updates based on evaluator comments	Stephan Müller, atsec
0.3	2008-09-29	Updates based on validator comments	Stephan Müller, atsec
0.4	2008-10-20	Consistency updates based on FSP assessment	Stephan Müller, atsec
0.5	2010-08-02	Include ASM2 hardware A.UTIL: clarification FCS_CKM.2(3) updated Update list of administrative interfaces TLS supports I&A by verifying client certificates	Stephan Müller, atsec
0.6	2011-02-04	Considerations of validator comments, update of assumptions to cover PD-0157	Stephan Müller, atsec
0.7	2011-02-11	Fixing minor issues from validator	Stephan Müller, atsec
0.8	2011-04-13	Fixing minor issues from validator	Stephan Müller, atsec
0.9	2011-06-27	Fixing minor issues from validator	Stephan Müller, atsec
0.10	2011-08-03	Fixing minor issues from validator	Stephan Müller, atsec
0.11	2011-11-02	Fixing minor issues from validator	Stephan Müller, atsec
0.12	2011-11-02	Fixing minor issues from validator	Stephan Müller, atsec
1.0	2012-01-25	Finalizing ST	Stephan Müller, atsec

1 ST Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is CloudShield CS-2000 with CPOS (CloudShield Packet Operating System) provided by CloudShield Technologies, Inc. The remaining portion of the Security Target will refer to the TOE as such or simply as CloudShield. CloudShield is an appliance which acts as a high performance packet processing application server. Administrator defined rule sets (also called policies and applications) provide the ability to inspect, modify and generate packets leveraging advanced features such as full packet string search, regular expression based pattern matching, stateful flow tracking, and statistical analysis. Based upon the administrator defined policies, the decisions and actions carried out upon network traffic allow the implementation of different ways to control or alter traffic flow, including the stateful filtering of packets. In addition, simple network tasks such as switching, routing, filtering, QoS (Quality of Service) and traffic management can be performed in policies. (Note: QoS refers to the mechanisms in the network software that make the actual determination of which packets have priority.)

1.1 *ST Structure*

The structure of this document is as defined by [CC] Part 1 Annex A:

- Section 1 is the TOE Overview Description.
- Section 2 provides the conformance claims.
- Section 3 provides the Security Problem Definition
- Section 4 provides the security objectives.
- Section 5 provides the extended components definition.
- Section 6 provides the security requirements.
- Section 7 provides the TOE summary specifications.

1.2 *Terminology*

This section contains definitions of technical terms that are used with a meaning specific to this document. Terms defined in the [CC] are not reiterated here, unless stated otherwise.

Administrative User A user who has the authorization to access the ASM part of the TOE and is allowed to read all data maintained by the TOE and to alter TSF data of the TOE.

Authentication data The password for each administrative user of the product. Authentication mechanisms using other authentication data are not supported in the evaluated configuration.

Data Arbitrary bit sequences in the TOE memory, including the package management facilities of the DPPM and the operational memory of the ASM, or on storage media, including the hardware database of the DPPM or on harddisk of the ASM.

Information Any data held within a CloudShield instance, including data in transit between systems.

Product Software components that comprise the CloudShield appliance, including the hardware and the CPOS operating system.

Target of Evaluation (TOE) The CloudShield hardware specified in this Security Target and the CPOS operating system providing the software for the ASM and the DPPM parts of the TOE.

Unauthenticated External IT Entities: IT Entities sending IP packets across the network which are intercepted and inspected by the DPPM part of the TOE.

User A user who has the authorization to access the ASM part of the TOE and is allowed to read data maintained by the TOE.

User Security Attributes As defined by functional requirement FIA_ATD.1, every user is associated with a number of security attributes which allow the TOE to enforce its security functions on this user.

1.3 ST Reference and TOE Reference

TOE Identification: CloudShield CS-2000 with CPOS 3.0.3

ST Version: 1.0

Authors: Stephan Müller

Publication Date: 2012-01-25

Keywords: CloudShield, CS-2000, CPOS, network appliance, packet inspection, high performance packet processing application server

1.4 TOE Overview

CloudShield is a multi-function solution appliance without any pre-configured network capability set programmed into the system. CloudShield allows network operators (administrators) to define policies (in the form of rule sets) that instruct the TOE to analyze, make decisions, and take action on packet data received from the network. Possible actions that the TOE can be configured to execute include inspection of any packet data, capture of portions or all packet data, modification of packet data, insertion of new packets, drop or discard of packets, and algorithm processing. The heuristics of these actions are defined by applications (rule sets) written in a high-level data plane programming language, called RAVE, designed to make the development of packet processing policies and applications easier. RAVE is the actual name of the programming language and is not an acronym. Administrators use the PacketWorks IDE (Integrated Development Environment) to develop RAVE applications (rule sets). The development of RAVE applications are performed on a separate personal computer executing the PacketWorks IDE and subsequently securely uploaded to the TOE through the management interfaces provided by the TOE. The PacketWorks IDE is not considered to be part of the TOE.

Based upon the RAVE application, the decisions and actions carried out upon network traffic allows the implementation of different capabilities to control or alter traffic flow, including the stateful filtering of packets.

The physical computer blade that carries out RAVE applications on packet data is called the Deep Packet Processing Module (DPPM). The DPPM is physically inserted into a CS-2000 chassis enclosure. It includes its own processors, memory, and physical interfaces and implements a RAVE execution engine to carry out the logic defined by a loaded RAVE application. There can be either one or at maximum two instances of the DPPM blade in the evaluated CS-2000 configuration and each DPPM can be configured to independently execute a different RAVE application.

The physical computer blade that provides management access to the system is called the Application Server Module (ASM). The ASM is physically plugged into the same chassis enclosure as the DPPM blades. The ASM includes its own processor, memory, disk storage, and physical interfaces. There is a single instance of the ASM in the evaluated CS-2000 configuration used to manage DPPM blades within the same CS-2000 chassis. The ASM provides a serial port for console access and its own network interfaces used to access the following CS-2000 management applications:

- The Web Management Interface (WMI) is a graphical administrative interface used to manage TOE functions (for example, read audit trail data from the audit records, import rule sets that permit or deny information flows, and modify user attribute values). Administrators access the WMI over a TLS protected HTTP channel using a web browser application in the operational environment. Web browsers Internet Explorer 8 and Firefox 3.6 are supported for access to the Web Management Interface (WMI).
- The Command Line Interface (CLI) is a text-based administrative interface used to manage TOE functions (for example, read audit trail data from the audit records, import rule sets that permit or deny information flows, and modify user attribute values). Administrators access the CLI over an SSH interface using a terminal application in the operational environment. The Telnet as well as the serial console access of the CLI are disabled in the evaluated configuration.
- The SNMP (Simple Network Management Protocol - RFC2571) interface provides read-only access to appliance health, system statistics, and general state information. Users access the SNMP interface using an industry-standard server management application in the operational environment.
- The GODYN dynamic data update API is a text-based command interface to manage TOE functions (specifically, modify attribute values of rule sets that permit or deny information flows) and retrieve TOE statistics. In addition, the JSON (JavaScript Object Notation) dynamic update API is a message-based interface to manage the same information as with the GODYN API. Administrators access the GODYN / JSON interface through the Command Line Interface. Therefore, all restrictions applicable to the CLI apply to the GODYN / JSON interface as well. The subsequent discussions about the CLI implicitly also

cover the GODYN / JSON interface unless specifically noted. Please note that JSON can also be accessed directly (i.e. without using the CLI) via an SSL-protected network channel.

- The MySQL administrative interface provides read-only access to system statistics and general state information of the TOE at a similar level as the SNMP interface. Users access the SNMP interface using an industry-standard server management application in the operational environment. The interface to the MySQL database is disabled and disallowed in the evaluated configuration.

The TOE is deployable in various network topologies. When connected in-line, the DPPM executes a rule set that the TOE uses to actively mediate traffic between separate networks connected to different DPPM ports. The TOE receives packets on one port, processes them according to the RAVE application logic, and then if applicable, sends the same packets, modified packets or new packets out another port to return to the network. In an in-line configuration, the location of the TOE in the network ensures that network traffic cannot pass between networks without passing through the TOE processing logic.

When connected in a tap configuration, the TOE does not mediate traffic. Instead it receives a duplicate of the original network traffic (via a tap or span port) to perform passive analysis, statistics gathering, monitoring, and logging.

The TOE operates transparently to the IP infrastructure because the DPPM traffic forwarding interfaces do not own an IP address while providing a physically separate ASM implementing the functionality of system management, control, and monitoring applications. The ASM communicates with remote entities via network interfaces which are independent from the DPPM. This provides hardened protection since packet processing execution can only be controlled from the ASM, making it impossible to reach the network control logic, or assume control of the TOE, via the DPPM interfaces. Since a DPPM does not provide a MAC (Media Access Control) or IP address to the network, it is “invisible” to other network entities, regardless of whether it is deployed in an in-line or tap configuration.

How an enterprise wishes to use the TOE will determine how it should be connected to the network. The use of the TOE for traffic management and control, network monitoring and reporting, network security, and/or security policy enforcement, will dictate whether the TOE should be located on the boundary between an organizations’ internal network and external networks or whether it should be placed within the enterprises’ networks. Guidance documents are provided to help users install the TOE in the correct location for its intended usage and CloudShield will assist customers if requested. For stateful filtering and traffic flow functions, the TOE should be placed within a topology so that it can see both sides of all network conversations of interest. A TOE enforcing active functions such as replication and filtering can be placed between routers, between a router and switch or between two switches. A TOE enforcing passive applications such as statistics gathering and logging can be placed between the network routers or simply attached (tapped) to a network segment. Placements can be made on the boundary between an organization’s internal network and external networks or between two networks both belonging to an organization. Graphical representations of these placements are depicted in the Technical Training documentation.

1.4.1 TOE Type

The TOE is a programmable generic network device that controls, alters traffic flow or traffic contents based on freely programmable rulesets.

CloudShield can therefore establish the functionality of a Secure Messaging, Guard, Network Management, Cross Domain Solutions, or Content Router.

1.4.2 Intended Method of Use

The evaluation is intended for the operation of the TOE operating in an in-line setup or in a tap setup.

The administrative interface is intended to be connected to a network with other well-behaved network systems and operating systems administered by the same management domain. All systems connected to that network must be configured according to a defined common security policy.

1.4.3 Provided Security Functionality

The TOE provides the following security functions:

- Security audit
- Cryptographic support
- Information flow control

- Identification and authentication
- Security Management
- Protection of the TSF

The architectural properties of the TOE of domain separation and reference mediation support the main security functions and ensure that these mechanisms are always invoked and cannot be bypassed.

1.5 TOE Description

1.5.1 CloudShield Structure

The CS-2000 2RU chassis supports dual, hot-swappable AC or DC power supply modules and a redundant fan tray assembly accessible from the rear of the chassis. The ASM and one or two DPPM modules are physically inserted into the front of the chassis. All of these components are included in the evaluated CS-2000 configuration.

TOE components and their relationship with each other are depicted in the figure below. Blades communicate with each other using an internal Gigabit Ethernet network interface provided by the chassis that is not otherwise externally visible.

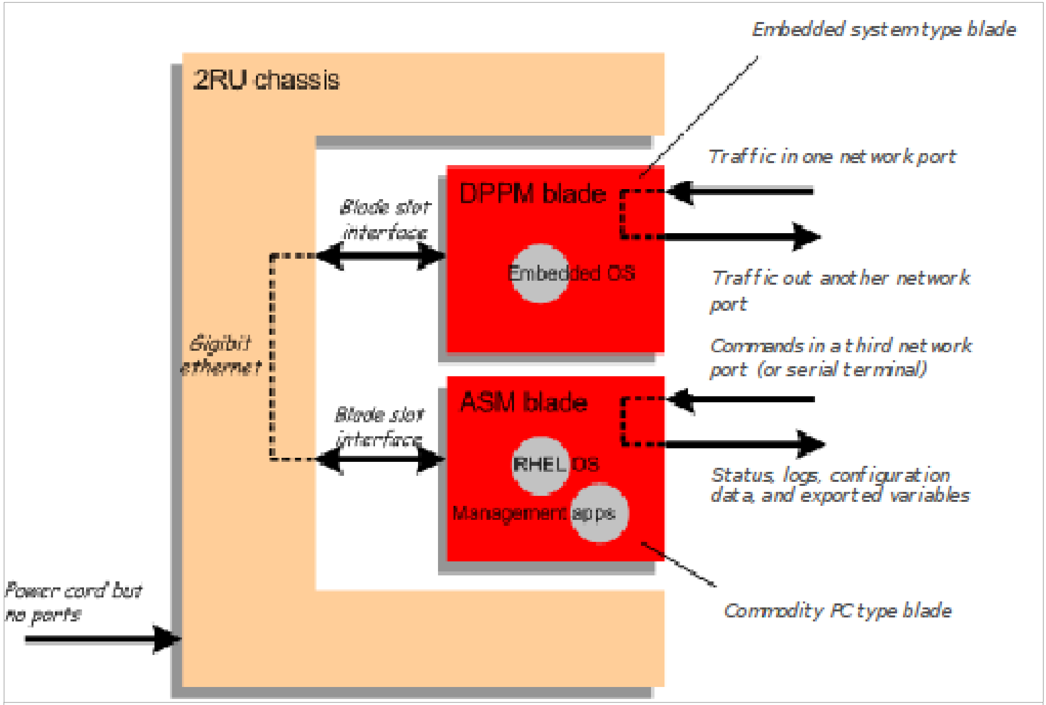
Embedded system type blade
2RU chassis
Traffic in one network port
DPPM blade
Blade slot interface
Embedded OS
Traffic out another network port
Commands in a third network port (or serial terminal)
Gigabit ethernet
ASM blade
Blade slot interface
RHEL OS
Management apps
Status, logs, configuration data, and exported variables
Power cord but no ports
Commodity PC type blade

Illustration 1 CS-2000 Structure of DPPM and ASM

1.5.1.1 DPPM

Each DPPM consists of a network processing complex, a silicon database subsystem which provides a transient storage area to process streams of packets, a regular expression pattern matching sub-system, an ARM processor running an embedded Linux, three external physical connectivity options; Ethernet, OC-48 Packet over SONET, SDH (Synchronous Digital Hierarchy), or 10G Ethernet. In addition, each DPPM has a dedicated Gigabit Ethernet port for packet capture and logging.

The DPPM Control OS is an embedded Linux implementation that has been modified by CloudShield to support the DPPM blade hardware. The embedded version of Linux differs from non-embedded Linux in several major ways.

- Embedded Linux in general is intended to be used on dedicated devices (i.e. single-purpose product types) such as the DPPM blade.
- Embedded Linux is not intended for use as a multi-user operating system. For example it does not support memory protection (all processes can access the data of all other processes).

- Embedded Linux is very similar to a real-time operating system (RTOS) in that it is built for speed. For example, the Embedded Linux kernel is different than the standard versions in that it is pre-emptible (the kernel can be interrupted mid-task, so that other applications can continue to run when another application is working in the background).

Network data sent by unauthenticated external IT entities enters and exits the TOE through the physical DPPM network interfaces. Incoming traffic passes into a framer that recognizes and packages frames into packets – the logical unit of data in networks.

Each packet is forward to the Packet Switch Field-Programmable Gate Arrays (FPGA) abbreviated as PSW for pre-processing and distribution. The PSW performs a checksum validation for L3/L4, a complete IP header decode, and initializes a Packet Information Block (PIB) or metadata control structure, to accompany the incoming packet to the network processing complex where the RAVE rule set executes. In the DPPM-800, the PSW FPGA also incorporates a Traffic Control Subsystem (TCS) that provides adaptable selective traffic filtering and load-balancing to distribute high-speed 10G traffic streams over multiple DPPM-800 modules clustered together. The TCS analyzes traffic at layers 2-4 to direct packets to specific destinations based on the results of the analysis. Each destination is a port or DPPM network processing complex. Packets then pass from the PSW FPGA to the network processing complex (note: in the DPPM-800, the TCS may filter/drop packets before sending to the network processing complex).

The network processing complex receives and transmits packets to and from the PSW using board-level components called Packet Receivers and Packet Transmitters, respectively. Packets are placed into an input buffer and the network processing complex executes the rule set to examine or modify each packet and make logic decisions regarding the handling of each packet. Messages that span multiple packets are reassembled by the network processing complex.

A silicon database subsystem provides persistent storage for the network processing complex applications running on the DPPM. It is a firmware implementation of a relational database and supports the definition of database tables for state tracking and the storage of global data, arrays, and matrices.

A regular expression pattern matching subsystem performs unstructured packet processing of packets as requested from the network processing complex, the results of which are then returned to the network processing complex.

If the RAVE application logic transmits a packet back to the network, the packet is sent to the PSW for checksum re-calculation and then out the selected DPPM physical interface.

1.5.1.2 ASM

The ASM includes its own processor, memory, disk storage and system database, physical interfaces and executes a version of RHEL operating system.

The TOE employs a multi-layered approach to security. A three-tiered architecture allows only the secure management plane to control packet processing through tightly controlled communications channels. Since packet processing execution can only be controlled from the ASM, it is not possible to assume control of, nor even compromise, the CS-2000 from the DPPM. The TOE enforces the following protection mechanisms: Identification and authentication mechanism is enforced whereby a user must enter a username and password to access the ASM. After a user successfully logs in, the CloudShield's Mandatory Access Control (MAC) System enforces roles which control further access to the TOE's management functions. In this sense, the term MAC does not refer to the traditional sense of controlling access to user data, but rather it restricts access to TSF data by the use of administrative roles.

1.5.1.3 Network traffic rules – RAVE Applications

Network traffic processing rule sets are created using the CloudShield PacketWorks IDE application on a commodity PC in the operational environment. The rules are implemented using the RAVE programming language. The RAVE programming language is the interface for administrators to configure the TOE (the PacketC environment is outside of the scope of this evaluation).

Multiple rule sets may be combined together to form a single Application Deployment Package (ADP). An ADP incorporates a virtual patch panel concept to connect multiple rule sets through "virtual wires" that map between the start and stop nodes of each individual rule set. This allows programmers to combine the policy logic of different discrete network traffic features (e.g. Distributed Denial of Service (DDoS) protection, anti-virus, etc) effectively into one comprehensive rule set that can be deployed on a DPPM to support multi-mission policy enforcement.

The progression through an ADP represents the application of multiple discrete rule sets, or policies, according to the "virtual wire" connectivity. When one intermediate rule set completes execution, the "virtual wire" hands off processing to the next rule set. Processing terminates with the last rule set defined in the ADP.

From the IDE, an ADP is uploaded to the TOE using the WMI or CLI where it is saved into the ASM system database. Using the WMI or CLI, a user chooses an ADP from the selection stored in the database and commits (i.e. loads) it onto the ASM which in turn forwards it to the intended DPPM to configure the packet processing capabilities of the DPPM. If more than one DPPM is present in the TOE, each may receive an independent ADP rule set. Once loaded into the DPPM, all new incoming packets are subjected to the new rule set logic.

This evaluation confirms that the RAVE instruction set and patch panel work correctly, but does not and cannot confirm that any particular RAVE program is suitable for the tasked claimed by its author. This evaluation establishes confidence that the RAVE program will work as written.

The RAVE application can specify a particular rule that forwards the first 64 bytes of the processed IP packet to the syslog trail maintained by the ASM to support a logging of the ongoing communication. In addition, RAVE applications can specify a different rule which allow the IP packets to be stored in the MySQL database of the ASM. The MySQL database is a storage backend for statistical data and various configuration options.

1.5.1.4 TOE guidance

Additional information and guidance documentation is provided in the following documents:

- CloudShield CS-2000 Series Documentation Release 3.0.3
- CloudShield Installation, and Hardware Reference, And Ordering Guide: CS2000 Series Release 3.0.3
- CloudShield CS-2000 Quick Start Guide Release 3.0.3
- CloudShield System Software Release Notes, Release 3.0.3 CloudShield CS-2000 Command Line Interface Reference Guide Release 3.0.3
- CloudShield CS-2000 Web management Interface User Guide Series Release 3.0.3
- CloudShield Application Integration User Guide 3.0.3

  Secure Setup For Common Criteria Guide, Release 3.0.3

With the exception of the Secure Setup For Common Criteria Guide and Software Release Notes, the user guidance is available on CD shipped along with the TOE.

The Secure Setup For Common Criteria Guide can be obtained from the CloudShield CloudShield support website at https://www.cloudshield.com/support/. Please verify the certificate of the web server to ensure the authenticity and integrity of the guide.

The CloudShield System Software Release Notes is available as paper copy shipped along with the TOE.

1.5.2 Definition of the TOE Boundaries

1.5.2.1 Logical boundary

The logical boundaries of the TOE are described in the following sections covering the security functions.

In addition to the security functionality provided by the TOE, it relies on the correct operation of the IDE and the RAVE code compiler as well as the client software accessing the administrative interfaces provided by the TOE, like a web browser accessing the WMI.

To support the security functionality, the following components must be present in the operational environment:

- NTP server

In addition, the following optional components may be present in the operational environment to support the TOE:

- Syslog server

1.5.2.1.1 Audit

The ASM part of the TOE collects audit data and generates system audit log records for all configuration and security-relevant user actions. This provides the ability to investigate unauthorized system security and configuration activities after they occur so that proper remedial action can be taken. Configuration changes and security-related events and failures are recorded in a security audit log.

Operations invoked via the WMI, CLI, and GODYN / JSON type administrative interfaces provided by the ASM generate audit records. All audit log records are maintained in the ASM system database.

The system restricts the ability to manage the security audit logs by a privilege assigned to an administrator role. Only authorized administrators who have been identified and authenticated have access to the audit functions. Using the WMI and CLI management interfaces, authorized security administrators have the ability to:

- View all information related to a security audit log. The system ensures that no user without the proper authorization is able to view the security audit logs. Any unauthorized attempt results in a security audit log. The logs may be sorted in ascending/descending order or by time, type, source IP address, or user name to facilitate searches.
- Generate a security audit log file to upload for off-system archival and analysis
- Delete audit log entries and audit log files. The audit logs are protected from unauthorized deletion. Only authorized administrators who have been identified and authenticated have the ability to delete audit log records and files.

When the audit log fills up, the TOE allows the specification of one of the following behaviors:

- Stopping of traffic until a portion of the audit log is deleted
- Wrapping of the audit logs and overwriting the oldest entries
- Wrapping of the audit logs and overwriting the oldest entries and sending an alarm every five minutes

Please note that the syslog functionality provided by the TOE (including the functionality to send syslog data to remote log hosts) is not considered to be the auditing functionality and therefore not covered by the security claim.

The DPPM inherently does not generate audit logs.

1.5.2.1.2 Cryptographic support

The ASM part of the TOE includes its own instance of a cryptographic library to support remote trusted IT products to initiate SSL connections with the TOE for the purposes of uploading rule sets and remote administration of the TOE implementing a trusted channel to a remote trusted entity. The WMI, CLI, and GODYN / JSON administrative interfaces provide secure system management through the use of SSH and TLS-protected HTTP for protection to access the ASM. Encryption is not utilized on the DPPM interfaces nor supported for encrypting or decrypting network content flowing through the DPPM as the DPPM is the network analyzing part of the TOE that is never the endpoint of a TLS communication.

The cryptography used in this product has not been FIPS 140-2 certified. This Security Target claims compliance with the external standard for the cipher suites explained by the SFRs of FCS_COP.1 for the definition of the encryption algorithm. There are many ways of determining compliance with a standard. The vendor asserts the correctness of the cryptographic mechanisms.

1.5.2.1.3 Information flow control

The DPPM part of the TOE enforces information flow control based on defined RAVE applications. The evaluation ensures that the RAVE language constructs behave as documented. The creation of the applications is outside the scope of the TOE. The applications are assumed to be protected by the environment during creation and prior to being uploaded to the TOE by an authorized administrator via TLS-protected HTTP giving access to the WMI or through the SSH channel allowing access to the CLI.

RAVE language constructs allow the specification of rules to identify Ethernet frames and subsequently act on identified frames. RAVE allows the specification of actions including forwarding the frame, altering the frame, generating a new frame or dropping the frame.

The IDE distributes pre-defined RAVE subroutines that a developer can use to generate the intended RAVE application. The evaluation makes no claims to the correctness of these routines or their suitability for their claimed tasks.

1.5.2.1.4 Identification and authentication

The ASM part of the TOE maintains security attributes for each user account, and includes the ability to assign users to groups and to define access for users, providing administrative flexibility. The TOE requires users to provide unique identification and authentication data before any administrative access to the system is granted. The TOE has the ability to lock a user's account if the authentication attempts threshold has been exceeded. Furthermore, it provides a password quality enforcement mechanism.

The RADIUS support is not subject to evaluation.

All protocols listed in section 0 allowing the communication with the ASM are subject to identification and authentication.

1.5.2.1.5 Security management

The ASM part of the TOE provides the authorized administrators the ability to define policies that define the access rules on the traffic received by the TOE. There are several functions available to the authorized administrator, such as manage user accounts and modify the behavior of the information flow policies. Modification of the rule sets can only be done using the RAVE programming language – see section 7.3 for more details about the RAVE language.

The TOE supports administrative roles which are defined by the groups assigned to human users by an authorized administrator at the time a user account is created.

Individual users are not assigned access rights directly. Access to the TOE is controlled by defined groups and their privileges and by assigning users to one or more of the groups. Once groups are defined, individual users are placed into the group or groups with the appropriate access levels.

User and Group definitions are stored in the MySQL database. Generally, the MySQL database is used to store configuration information as well as statistical data.

Each group is granted one of three privilege levels (Read/Write, Read-only, or None) to one or more of the five management areas on the TOE:

- Hardware
- Network
- Software
- Security
- Configuration

A sixth management area, the Database, defines access to the TOE system database which is managed by the internal ASM software and can only be accessed remotely using SQL read-only queries through the administrative interfaces. The database is used to store the following configuration data:

- Administrative controls and safeguards are enforced for access to the TOE:
- Invalid login lockout thresholds and controls
- Password composition regulations
- Inactive session termination controls

1.5.2.1.6 Protection of the TSF

The architecture of the TOE provides protection mechanisms for its security functions as the TOE executes on stand-alone, protected hardware. One of the protection mechanisms is the enforcement that users must authenticate before any administrative operations can be performed on the system, whether those functions are related to the management of user accounts or the configuration of traffic flows (i.e. loading a RAVE application) or any other administrative activity. In addition, the TOE provides appropriate time stamps used for the auditing system as well as residual information protection which implies clearing any resource before re-allocating it again.

1.5.2.1.7 TOE Access

The TOE displays access banners before users perform identification and authentication. Interactive sessions of administrators can be configured to be locked when unattended.

1.5.2.2 Physical boundary

TOE and operational environment components are depicted in the figure below. The TOE includes the gray shaded areas in the illustration.

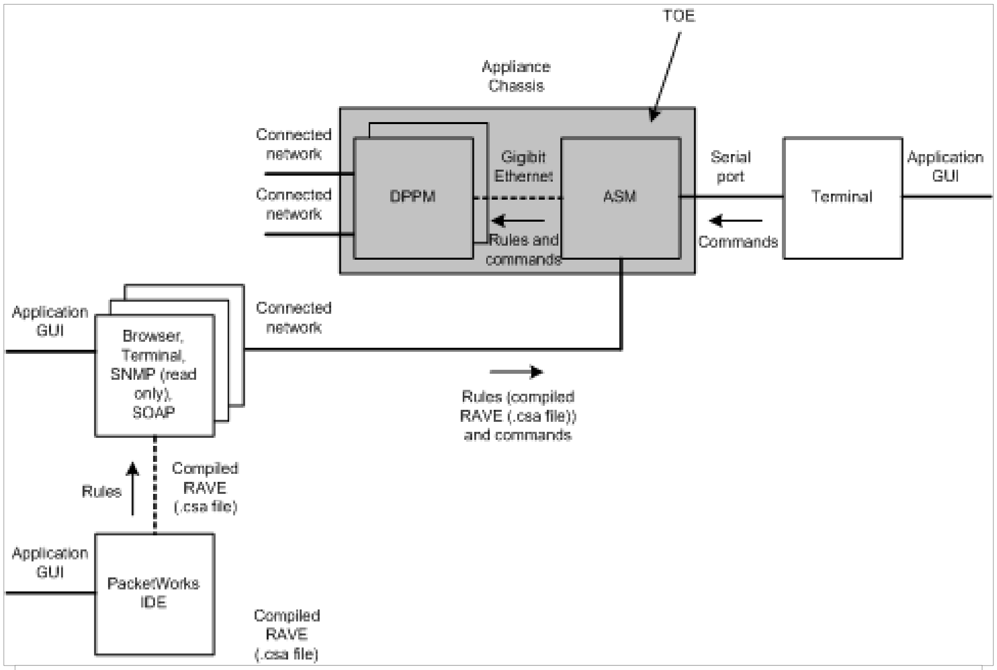

Illustration 2 Physical Boundary of TOE

Note: the serial port connection must be disabled in the evaluated configuration.

The hardware of the TOE consists of three major components: one or two Deep Packet Processing Modules, a single Application Server Module and a single 2RU Chassis enclosure which houses them.

The DPPM performs its operations of controlling the network information flow based on the RAVE application independently from any other DPPM or the ASM. The RAVE application is loaded into the DPPM and the DPPM implements the behavior specified by the RAVE application. Therefore, the operation of any DPPM and ASM is identical, irrespectively of the connections of the DPPMs or ASMs:

- One chassis supports one or two DPPMs – both are controlled by the ASM, which is also present in the chassis and operate independently.
- DPPMs of the same or separate chassis can be chained using the TCS functionality. TCS performs a pre-selection of network packets based on the TCS rules. If a packet is matched, it is forwarded to the configured DPPM for handling. The initial DPPM as well as the chained DPPM enforce the RAVE application logic independent from each other after the TCS selected the DPPM that shall handle the packet. This allows an even higher bandwidth for controlling the network flow.

The DPPM includes the CloudShield's Packet Operating System (CPOS™) run-time operating system that dynamically controls the DPPM resources to perform the packet operations (e.g. packet read, database table lookup, string search, variable update, packet replicate and capture, packet write, etc.) required by the RAVE applications. A new RAVE application can only be loaded from the ASM.

The ASM executes a version of RHEL with CloudShield Mandatory Access Control System.

1.5.2.2.1 It Supported hardware

All CS-2000 installations include the following components:

- CS-2000 Chassis Enclosure
- Application Server Module (ASM as well as ASM2 where the ASM2 provides newer CPU and newer hardware components – the software executed by both, the ASM and ASM2, is identical)
- Power Supply Modules
- Fan Tray Unit

All CS-2000 installations include at least one Deep Packet Processing Module. The available types supported by the TOE are:

- DPPM-500: Deep Packet Processing Module for GbE
- DPPM-510: Deep Packet Processing Module for GbE (includes high-speed interconnect support)
- DPPM-600: Deep Packet Processing Module for Packet over SONET and SDH (POS)
- DPPM-800: Deep Packet Processing Module for 10G Ethernet

The following components are disallowed in the evaluated configuration:

- Bypass Control Module (BCM)

2 Conformance Claims

2.1 Common Criteria

The ST is [CC] Part 2 conformant and Part 3 conformant. CC version 3.1 revision 3 is applied.

2.2 Packages

The ST claims an Evaluation Assurance Level of EAL4 augmented by ALC_FLR.3.

2.3 Protection Profiles

This Security Target does not claim conformance with any Protection Profile.

3 Security Problem Definition

3.1 Introduction

The TOE Security Problem Definition consists of the threats to security, organizational security policies, and usage assumptions as they relate to CloudShield. This section describes the security aspects of the environment in which the TOE is intended to be used and the manner in which it is expected to be employed.

3.2 Threats

The assumed security threats are listed below.

The IT assets to be protected comprise the information stored, processed or transmitted by the TOE. The term "information" is used here to refer to all data held within the TOE, including data in transit between instances of the TOE.

The TOE counters the general threat of unauthorized access to information, where "access" includes disclosure, modification and destruction.

The threat agents can be categorized as either:

- unauthorized users of the TOE, i.e. individuals who have not been granted the right to access the system; or
- authorized users of the TOE, i.e. individuals who have been granted the right to access the system.

The threat agents are assumed to originate from a well-managed user community in a non-hostile working environment, and hence the product protects against threats of security vulnerabilities that might be exploited in the intended environment for the TOE with medium level of expertise and effort. The TOE protects against straightforward or intentional breach of TOE security by attackers possessing a basic attack potential.

3.2.1 Threats countered by the TOE

T.COMPROT An attacker (possibly, but not necessarily, an unauthorized user of the TOE) may intercept a communication link between the ASM part of the TOE and another trusted IT product to read or modify information transferred between the TOE and the other trusted IT product in a way that cannot be detected by the TOE or the other trusted IT product.

T.IA An attacker (possibly, but not necessarily, an unauthorized user of the TOE) may impersonate an authorized user of the TOE. This includes the threat of an authorized user that tries to impersonate another authorized user without knowing the authentication information.

T.MEDIAT Information is sent through the TOE, which violates allowed information flow rules specified for and enforced by the Deep Packet Processing Module (DPPM).

T.ROLE An authorized user may gain access to resources or perform operations for which no rights have been granted.

3.2.2 Threats countered by the operational environment

TE.RULE_SET_PROTECTION

An unauthorized user may be able to inject new RAVE applications or modify RAVE applications during creation, modification and prior to upload to the TOE causing the TSF to enforce improper information flow rules.

3.3 Organizational Security Policies

P.ACCOUNTABILITY

The users of the system shall be held accountable for their actions within the system.

P.DPPM_LOG The operating environment is responsible for instrumenting auditing of the RAVE applications. This includes auditing changes to the state of the RAVE application and generating audit records from information gathered from the RAVE applications.

3.4 Assumptions

This section contains assumptions regarding the security environment and the intended usage of the TOE. The TOE is assured to provide effective security measures in a cooperative non-hostile environment only if it is installed, managed, and used correctly. The operational environment must be managed in accordance with assurance requirements documentation for delivery, operation, and user/administrator guidance. The following specific conditions are assumed to exist in an environment where the TOE is employed.

3.4.1 Physical Aspects

A.PHYSICAL The TOE is protected from unauthorized physical access. The application development environment is physically secured to a level of protection appropriate for the eventual deployment environment of the product.

3.4.2 Personnel Aspects

A.MANAGE Administrative users are competent to manage the TOE securely. In addition, administrative users are competent to utilize the RAVE programming language.

The operating system of the application development environment is patched regularly for known vulnerabilities. The RAVE compiler and equivalent tools are under the protection of an integrity checking mechanism, and ensure the compilation tools used are the vendor-approved versions and in vendor-approved configurations. The mechanism used to transfer compiled and bundled application programs to the target product ensure the integrity of the files transferred.

A.UTIL Users connecting to the ASM are competent to utilize the TOE securely, and trusted and abide by the instructions set forth in the TOE documentation.

Users making use of the IDE must ensure this IDE and the associated RAVE code compiler to be:

- securely protected,
- its integrity is ensured.

3.4.3 Connectivity Aspects

A.PEER Any other systems with which the ASM part of the TOE communicates is assumed to be under the same management control and operate under the same security policy constraints.

A.SEPARATION In case of an in-line setup of the TOE, the information flow control functionality of the TOE establishes the only physical or logical network connection between the different networks that are to be protected by the information flow control rules enforced by the TOE.

4 Security Objectives

This section defines the security objectives of CloudShield and its supporting environment. Security objectives reflect the stated intent to counter identified threats and/or comply with any organizational security policies identified. All of the identified threats and organizational policies are addressed under one of the categories below.

4.1 Security Objectives for the TOE

O.AUDIT

The TOE shall generate audit records for security relevant events and make that information available to administrators.

O.COMPROT

The TOE must protect its functions and TSF data from tampering and ensure that its security functions cannot be bypassed.

O.DPPM.LOG

The TOE shall provide logging capability for RAVE application developers to implement audit record generation.

O.FLOW

The TOE shall control the flow of information and enforce the configured information flow policy rules for the TOE.

O.IA

The TOE must ensure that only authorized users can access the TOE functions and they must be successfully identified and authenticated before any TOE security functions can be accessed.

O.SECURE_POLICY

The TOE must provide functionality that enables authorized administrators to use the TOE security management functions and must ensure that only authorized administrators are able to access such functionality.

4.2 Security Objectives for the TOE Environment

OE.DPPM.LOG

The operating environment shall instrument auditing of the RAVE applications. This includes auditing changes to the state of the RAVE application and generating audit records from information gathered RAVE applications.

OE.MANAGE

The TOE must be installed, configured, and managed by competent, trusted and trained users in accordance with the applicable guidance documentation. The TOE must be under the same management domain as the computer systems connected to the ASM part of the TOE. The administrator must set up the network in a way that prevents any physical or logical connections between the networks to be protected by the TOE which are not established through the TOE in case of an in-line setup of the TOE.

Administrative users are competent to manage the TOE securely. In addition, administrative users are competent to utilize the RAVE programming language. The operating system of the application development environment is patched regularly for known vulnerabilities. The RAVE compiler and equivalent tools are under the protection of an integrity checking mechanism, and ensure the compilation tools used are the vendor-approved versions and in vendor-approved configurations. The mechanism used to transfer compiled and bundled application programs to the target product ensure the integrity of the files transferred.

Users connecting to the ASM are competent to utilize the TOE securely, and trusted and abide by the instructions set forth in the TOE documentation. Users making use of the IDE must ensure this IDE and the associated RAVE code compiler to be: securely protected, and its integrity is ensure.

OE.PHYSICAL

The TOE must be protected from unauthorized physical access. The application development environment is physically secured to a level of protection appropriate for the eventual deployment environment of the product.

OE.RULE_SET_PROTECTION

The RAVE applications are protected from unauthorized modification or injecting of new applications during creation, modification and prior to being uploaded to the TOE.

4.3 Security Objective Rationale

The following tables provide a mapping of security objectives to the Security Problem Definition, illustrating that each security objective covers at least one threat, assumption or policy and that each threat, assumption or policy is covered by at least one security objective.

4.3.1 Security Objectives Coverage

Table 1 Mapping objectives to threats, assumptions and policies

Objective	Threat / Policy
O.AUDIT	P.ACCOUNTABILITY
O.COMPROT	T.COMPROT
O.DPPM.LOG	P.DPPM.LOG
O.FLOW	T.MEDIAT
O.IA	T.IA
O.SECURE_POLICY	T.IA, T.ROLE

Table 2 Mapping objectives for the environment to threats, assumptions and policies

Environmental Objective	Threat / Assumption / Policy
OE.DPPM.LOG	P.DPPM.LOG
OE.MANAGE	A.MANAGE, A.UTIL, A.PEER, A.SEPARATION
OE.PHYSICAL	P.PHYSICAL
OE.RULE_SET_PROTECTION	TE.RULE_SET_PROTECTION

4.3.2 Security Objectives Sufficiency

T.COMPROT: The threat of compromising or modifying communication between remote trusted entities and the ASM part of the TOE without being detected is removed by O.COMPROT requiring the ability to set up an Inter-TSF trusted channel between the TOE and the remote trusted entity which allows the data flowing through the channel to be protected against disclosure and undetected modification.

T.IA: The threat of impersonation of an authorized user by an attacker is sufficiently diminished by O.IA which requires the identification and authentication of users when obtaining access to the TOE. O.SECURITY_POLICY ensures that only administrators have the ability to add new users or modify the security attributes of existing users. Both security objectives together ensure that no unauthorized user can impersonate as an authorized user.

T.MEDIAT: The threat of relaying information by the TOE not allowed by the information flow policy is removed by O.FLOW which ensures that the TOE controls the entire information flow.

T.ROLE: The threat of gaining access to resources or performing unauthorized operations is removed by O.SECURE_POLICY which specifies that the TOE provides management capabilities which allow the specification of access restrictions.

TE.RULE_SET_PROTECTION: The threat of compromising the RAVE applications is removed by OE.RULE_SET_PROTECTION which requires that the environment is controlled in a way that prevents unauthorized modification and injection of new RAVE applications.

P.ACCOUNTABILITY: The policy to provide a means to hold users accountable for their activities is implemented by O.AUDIT providing the TOE with such functionality and the capability to review the audit trail.

P.DPPM.LOG: The policy to ensure instrumentation of RAVE applications to provide auditing is implemented by OE.DPPM.LOG requirement such measure from the IT environment. To support the IT in instrumenting RAVE applications to generate audit records, the TOE must provide the auditing capability as defined by O.DPPM.LOG.

A.PHYSICAL: The assumption on physical protection of the TOE is covered by the objective OE.PHYSICAL demanding the physical protection of the TOE.

A.MANAGE: The assumption on competent administrative users managing the TOE, using the RAVE language and using the RAVE development environment with its compiler is covered by OE.MANAGE requiring such users.

A.UTIL: The assumption on competent, trusted and guidance-following users of the ASM operating the TOE and protecting and ensuring integrity of the development environment is covered by OE.MANAGE requiring such users.

A.PEER: The assumption on the same management control and security policy constraints for systems connected to the ASM part of the TOE is covered by OE.MANAGE requiring such global administration approach.

A.SEPARATION: The assumption that the TOE is the only connection of the protected networks is covered by OE.MANAGE requiring such a network topology.

5 Extended Components Definition

No extended components are defined with this ST.

6 Security Requirements

6.1 TOE Security Functional Requirements

This section specifies the security functional requirements (SFRs) for the TOE. Functional requirements components in this ST were drawn from CC Part 2.

All operations except iterations which are performed in this ST are marked in bold within each of the requirements.

6.1.1 Security Audit (FAU)

6.1.1.1 Audit Data Generation (FAU_GEN.1)

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) the events identified in the following table.

Component	Event
FAU_SAR.1	Reading of information from the audit records.
FIA_UAU.1	All use of the authentication mechanism
FIA_UID.1	All use of the user identification mechanism
FMT_MOF.1	All modifications in the behavior of the functions of the TSF, including every change to the RAVE applications (loading, changing and removing of applications) except the modifications to the CAM (internal database memory) performed through JSON.
FMT_MTD.1	All modifications to the values of TSF data
FMT_SMF.1	Use of the management functions.
FMT_SMR.1	Modifications to the group of users that are part of a role
FTA_SSL.1	Locking of an interactive session.

Application Note: The JSON interface is designed to make extremely frequent updates to the CAM tables. Since auditing each update to the CAM would hinder the bandwidth of the interface, the TOE does not auditing CAM updates through JSON.

The requirement of audit generation falls to the client application in the TOE environment that exercises the JSON interface. To do this, the client must take into consideration the effects of its changes on the CAM to produce meaningful audit records.

Therefore, the administrator of the TOE must ensure that any user allowed access to the JSON interface (through the CLI or through JSONSSL) complies with the organizational auditing requirements.

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 **ST**, **the administrator IP address**.

Application Note: The TOE is capable of maintaining two types of subjects:

1. users identified and authenticated to the ASM part of the TOE to perform the allowed tasks based on their role, and
2. Unauthenticated external IT entities as specified in FDP_IFC.1.

As listed in FAU_GEN.1.1, only the SFRs addressing the administration of the TOE are covered with auditing functionality. Therefore, FAU_GEN.1.2 applies to the operations performed by a user identified and authenticated to the ASM part of the TOE.

6.1.1.2 User Identity Association (FAU_GEN.2)

FAU_GEN.2.1 For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

6.1.1.3 Audit Review (FAU_SAR.1)

FAU_SAR.1.1 The TSF shall provide **authorized administrators** with the capability to read **all audit information** from the audit records.

FAU_SAR.1.2 The TSF shall provide the audit records in a manner suitable for the user to interpret the information.

6.1.1.4 Restricted Audit Review (FAU_SAR.2)

FAU_SAR.2.1 The TSF shall prohibit all users read access to the audit records, except those users that have been granted explicit read-access.

6.1.1.5 Selectable Audit Review (FAU_SAR.3)

FAU_SAR.3.1 The TSF shall provide the ability to apply **searches, sorting** of audit data based on **the following attributes:**

a) user ID

b) date/time

c) event type

d) **source IP address.**

6.1.1.6 Protected audit trail storage (FAU_STG.1)

FAU_STG.1.1 The TSF shall protect the stored audit records in the audit trail from unauthorized deletion.

FAU_STG.1.2 The TSF shall be able to **prevent** unauthorized modifications to the stored audit records in the audit trail.

6.1.2 Cryptographic Support (FCS)

6.1.2.1 Cryptographic Key Generation for symmetric keys (FCS_CKM.1)(1)

FCS_CKM.1.1(1) The TSF shall generate **symmetric** cryptographic keys in accordance with a specified cryptographic key generation algorithm **symmetric key generation using a random number generator** and **the following** specified cryptographic key sizes

a) 128 bit,

b) 168 bit,

c) 192 bit, and

d) **256 bit**

that meet the following: **ANSI X9.31 appendix A2.4 based on AES**.

Application Note: The symmetric keys are used for SSHv2 and TLS.

6.1.2.2 Cryptographic Key Generation for SSH DSS keys (FCS_CKM.1)(2)

FCS_CKM.1.1(2) The TSF shall generate **asymmetric** cryptographic keys in accordance with a specified cryptographic key generation algorithm **defined in FIPS 186-2** and **the following** specified cryptographic key sizes:

a) **768 bit (RSA keys),**

b) **1024 bit (DSA and RSA keys),**

c) **2048 bit (RSA keys),**

d) **4096 bit (RSA keys),**

e) **8192 bit (RSA keys),**

f) **16384 bit (RSA keys),**

g) **32768 bit (RSA keys)**

that meet the following: **generation of asymmetric public/private key pairs as defined in FIPS 186-2**.

Application Note: The generation of the keys apply to the use of SSH host keys and SSH user keys. The tool allowing the key generation is provided with ssh-keygen(1).

6.1.2.3 Cryptographic Key Generation for TLS RSA keys (FCS_CKM.1)(3)

FCS_CKM.1.1(3) The TSF shall generate asymmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm **RSA key generation** and specified cryptographic key sizes **1024 bit** that meet the following: **ANSI X9.31**.

Application Note: The tool allowing the key generation is provided with the set of applications invoked by openssl(1ssl).

6.1.2.4 DH Cryptographic Key Distribution for SSH (FCS_CKM.2)(1)

FCS_CKM.2.1(1) The TSF shall distribute **symmetric** cryptographic keys in accordance with a specified cryptographic key distribution method **diffie-hellman-group1-sha1 and diffie-hellman-group14-sha1** that meets the following: **SSH version 2.0 protocol defined with [SSHTRANS]**.

6.1.2.5 DSS Public Cryptographic Key Distribution for SSH (FCS_CKM.2)(2)

FCS_CKM.2.1(2) The TSF shall distribute **public parts of asymmetric** cryptographic keys in accordance with a specified cryptographic key distribution method **of cryptographic certificates for public DSS keys** that meets the following: **SSH version 2.0 protocol defined with [SSHTRANS]**.

6.1.2.6 Cryptographic Key Distribution for TLS (FCS_CKM.2)(3)

FCS_CKM.2.1(3) The TSF shall distribute **symmetric** cryptographic keys in accordance with a specified cryptographic key distribution method **Transport Layer Security handshake using RSA encrypted exchange of pre-master secrets** that meets the following: **[TLS]**.

6.1.2.7 SSH Cryptographic Operation (FCS_COP.1)(1)

FCS_COP.1.1(1) The TSF shall perform **encryption and decryption** in accordance with **the following** specified cryptographic algorithm and cryptographic key sizes:

a) **AES in CBC mode with 128 bit, 192 bit or 256 bit key size; and**

b) **HMAC-SHA1 with 160 bit hash size**

that meet the following: **SSH version 2.0 Transport Layer Protocol with the aforementioned cipher suites defined with [SSHTRANS]**.

6.1.2.8 TLS Signature Verification Cryptographic Operation (FCS_COP.1)(2)

FCS_COP.1.1(2) The TSF shall perform **digital signature verification** in accordance with a specified cryptographic algorithm **RSA** and cryptographic key sizes **1024 bit** that meet the following: **TLS version 1.0 Protocol with the aforementioned cipher suite defined with [TLS]**.

Application Note: For HTTPS as well as JSONSSL, the TOE is able to verify the SSL certificate presented by the client.

6.1.2.9 TLS Cryptographic Operation (FCS_COP.1)(3)

FCS_COP.1.1(3) The TSF shall perform **encryption and decryption** in accordance with **the following** specified cryptographic algorithm and cryptographic key sizes:

a) TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA

b) TLS_DHE_DSS_WITH_AES_128_CBC_SHA

c) TLS_DHE_DSS_WITH_AES_256_CBC_SHA

d) TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA

e) TLS_DHE_RSA_WITH_AES_128_CBC_SHA

f) TLS_DHE_RSA_WITH_AES_256_CBC_SHA

g) TLS_RSA_WITH_3DES_EDE_CBC_SHA

h) TLS_RSA_WITH_AES_128_CBC_SHA

i) TLS_RSA_WITH_AES_256_CBC_SHA

j) TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA

k) TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA

l) TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA

m) TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA

n) TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA

o) TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA

p) TLS_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA

q) TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA

r) TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA

s) TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA

t) TLS_ECDH_RSA_WITH_AES_128_CBC_SHA

u) **TLS_ECDH_RSA_WITH_AES_256_CBC_SHA**

that meet the following: **TLS version 1.0 Protocol with the aforementioned cipher suite defined with [TLS]**.

6.1.3 User Data Protection (FDP)

6.1.3.1 Complete information flow control (FDP_IFC.2)

FDP_IFC.2.1 The TSF shall enforce the **unauthenticated SFP** on

Subjects: unauthenticated external IT entities that send and receive information between each other which is mediated through the TOE;

Information: IP packets sent through the TOE from one subject to another;

and all operations that cause that information to flow to and from subjects covered by the SFP.

FDP_IFC.2.2 The TSF shall ensure that all operations that cause any information in the TOE to flow to and from any subject in the TOE are covered by an information flow control SFP.

6.1.3.2 Simple security attributes (FDP_IFF.1)

FDP_IFF.1.1 The TSF shall enforce the **unauthenticated SFP** based on the following types of subject and information security attributes:

Subject security attributes: a source physical DPPM network interface through which the IP packet entered the TOE;

Information security attributes: all data contained within each IP packet.

FDP_IFF.1.2 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold:

An IP packet received with a source DPPM network interface is provided to the configured RAVE application and the IP packet is acted upon by the rule set defined with the RAVE application. This RAVE application holds a rule set with the following potential operations (in addition to general programming language concepts of branching, loops, conditionals, mathematical and logical operations):

1. **Identifying of properties in the entire IP packet using one or more of the following provided concepts:**
   a. **String matching,**
   b. **Pattern matching search based on regular expressions,**
   c. **Stateful flow tracking matches,**
   d. **Statistical analysis matches,**
   e. **Timer-based matching;**
2. **Performing one or more actions an identified IP packet:**
   a. **Drop the IP packet (no information is communicated),**
   b. **Storing of information found within the IP packet or auxiliary information (such as maintaining counters or state information for identified IP packets) in the hardware database of the DPPM for offloading to the ASM,**
   c. **Unlimited modification of any part of the IP packet,**
   d. **Generation of a new IP packet without a limitation of its contents, and**
   e. **Forwarding of the IP packet to any DPPM network interface.**

FDP_IFF.1.3 The TSF shall enforce **no additional information flow control SFP rules**.

FDP_IFF.1.4 The TSF shall explicitly authorise an information flow based on the following rules: **no additional rules**.

FDP_IFF.1.5 The TSF shall explicitly deny an information flow based on the following rules: **no additional rules**.

Application Note: This SFR covers the RAVE programming interface, allowing and covering arbitrary RAVE applications. The evaluation ensures that the specification of the individual RAVE op codes and the sequence of these op codes are enforced by the TOE.

Application Note: The intention of the logging capability stated in 2.b. is to address the objective O.DPPM_LOG which requires the TOE to provide logging capability for RAVE application developers to implement audit record generation. The “auxiliary” information refers to meta data the DPPM maintains for each network packet. When log data is offloaded to the ASM, the format of the data is defined, such as storing the information as pcap files.

6.1.3.3 Object Residual Information Protection (FDP_RIP.2)

FDP_RIP.2.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon **the allocation of the resource** to **all objects**.

Application Note: This SFR applies to the fundamental objects like memory objects used to implement and support all other SFRs.

6.1.4 Identification and Authentication (FIA)

6.1.4.1 Authentication failure handling (FIA_AFL.1)

FIA_AFL.1.1 The TSF shall detect when **an administrator configurable positive number within the range of 1 and 8 (both values are included, the default is 3) of** unsuccessful authentication attempts occur related to **a user’s attempt to log on**.

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been **met or surpassed**, the TSF shall **lock the user’s account**.

6.1.4.2 User Attribute Definition (FIA_ATD.1)

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users:

a) User identifier;

b) Group memberships;

c) **Authentication data.**

6.1.4.3 Verification of Secrets (FIA_SOS.1)

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet **for each attempt to use the authentication mechanism, the probability that a random attempt will succeed is less than one in 1,000,000.**

6.1.4.4 User authentication before any action (FIA_UAU.2)

FIA_UAU.2.1 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

Application Note: Identification and authentication services are only applicable to connections to the ASM component of the TOE. External entities that send IP packets through the DPPM part of the TOE are not covered by any aspect of identification and authentication.

6.1.4.5 Multiple authentication mechanisms (FIA_UAU.5)

FIA_UAU.5.1 The TSF shall provide **the following authentication mechanisms** to support user authentication:

a) Username / password authentication;

b) SSL-certificate based authentication.

FIA_UAU.5.2 The TSF shall authenticate any user’s claimed identity according to the following rules:

a) SSH / KVM / HTTP: Username / password authentication is the only authentication method;

b) JSONSSL: SSL-certificate based authentication is the only authentication method;

c) HTTPS: If PKI is enabled and the client presents an SSL certificate, SSL-certificate based authentication is enforced; otherwise username/password authentication is enforced.

Application Note: For SSL-certificate based authentication, the DN of the certificate specifies the user ID. A user with that DN must be defined in the TOE with the assignment of the appropriate permissions to allow the user to successful interoperate with the TOE.

6.1.4.6 User identification before any action (FIA_UID.2)

FIA_UID.2.1 The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

Application Note: See application note of FIA_UAU.2.

6.1.4.7 User-Subject Binding (FIA_USB.1)

FIA_USB.1.1 The TSF shall associate the following user security attributes with subjects acting on the behalf of that user:

d) The user identity which is associated with auditable events;

e) **The group membership or memberships used to restrict the administrative actions allowed to be performed by the user.**

FIA_USB.1.2 The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users:

a) After successful identification and authentication, the user ID of the session for the user has to be set to the user ID specified for the user entry of the successfully authenticated user.

b) **After successful identification and authentication, the groups IDs of the session for the user has to be set to the group IDs specified for the user entry of the successfully authenticated user.**

FIA_USB.1.3 The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: **none**.

6.1.5 Security Management (FMT)

6.1.5.1 Management of security functions 30behavior (FMT_MOF.1)

FMT_MOF.1.1 The TSF shall restrict the ability to **determine the behavior30 of** the functions **audit** to **administrators associated with the security role**.

6.1.5.2 Management of security attributes (FMT_MSA.1)

FMT_MSA.1.1 The TSF shall enforce the **unauthenticated SFP** to restrict the ability to **modify** the security attributes **described with the RAVE application** to **administrators associated with the configuration role**.

6.1.5.3 Static Attribute Initialization (FMT_MSA.3)

FMT_MSA.3.1 The TSF shall enforce the **unauthenticated SFP** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow **nobody** to specify alternative initial values to override the default values when an object or information is created.

6.1.5.4 Management of user attributes (FMT_MTD.1) (1)

FMT_MTD.1.1(1) The TSF shall restrict the ability to **modify, delete, and create** the **user security attributes** to **administrators associated with the security role**.

6.1.5.5 Management of Authentication Data (FMT_MTD.1)(2)

FMT_MTD.1.1(2) The TSF shall restrict the ability to **modify** the **authentication data** to **the following:**

a) Administrators associated with the security role are authorized to manage user accounts; and

b) **Administrators for other functions authorized to modify their own authentication data.**

6.1.5.6 Specification of Management Functions (FMT_SMF.1)

FMT_SMF.1.1 The TSF shall be capable of performing the following **security** management functions:

a) Review and modification of the RAVE application executed by the DPPM;

b) Review of audit log;

c) User attribute management; and

a) **Authentication data management.**

6.1.5.7 Security Roles (FMT_SMR.1)

FMT_SMR.1.1 The TSF shall maintain the roles **which give various permissions to access one or more of the following entities:**

a) Hardware,

b) Network,

c) Software,

d) Security,

e) **Configuration, and**

f) **Database.**

FMT_SMR.1.2 The TSF shall be able to associate users with roles.

Application Note: Any role listed here but not specified in the other management SFRs cover management aspects that are not required by this ST, like updating the TOE software.

6.1.6 Protection of the TSF (FPT)

6.1.6.1 Reliable Time Stamps (FPT_STM.1)

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps for its own use.

6.1.7 TOE access (FTA)

6.1.7.1 TSF-initiated session locking (FTA_SSL.1)

FTA_SSL.1.1 The TSF shall lock an interactive session after **an administrator configurable value between 5 and 480 minutes (60 minutes is the default)** by:

a) clearing or overwriting display devices, making the current contents unreadable;

b) disabling any activity of the user's data access/display devices other than unlocking the session.

FTA_SSL.1.2 The TSF shall require the following events to occur prior to unlocking the session: **the user must re-authenticate with the correct identification and authentication data**.

6.1.7.2 Default TOE access banners (FTA_TAB.1)

FTA_TAB.1.1 Before establishing a user session, the TSF shall display an advisory warning message regarding 31unauthorized use of the TOE.

6.1.8 Trusted path/channels (FTP)

6.1.8.1 Inter-TSF trusted channel (FTP_ITC.1)

FTP_ITC.1.1 The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

FTP_ITC.1.2 The TSF shall permit **another trusted IT product** to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for **no functionality**.

Application Note: This SFR applies to the communication between the ASM and the administrative system.

6.2 Security Requirements Rationale

This section provides the rationale for the internal consistency and completeness of the security functional requirements defined in this Security Target.

6.2.1 Security Requirements Coverage

The following table shows that each security functional requirement addresses at least one objective.

Table 3 Mapping Security Functional Requirements to Objectives

SFR	Objectives
FAU_GEN.1	O.AUDIT
FAU_GEN.2	O.AUDIT
FAU_SAR.1	O.AUDIT

SFR	Objectives
FAU_SAR.2	O.AUDIT
FAU_SAR.3	O.AUDIT
FAU_STG.1	O.AUDIT
FCS_CKM.1(1)	O.COMPROT
FCS_CKM.1(2)	O.COMPROT
FCS_CKM.2(1)	O.COMPROT
FCS_CKM.2(2)	O.COMPROT
FCS_CKM.2(3)	O.COMPROT
FCS_COP.1(1)	O.COMPROT
FCS_COP.1(2)	O.COMPROT O.IA
FCS_COP.1(3)	O.COMPROT
FDP_IFC.2	O.FLOW
FDP_IFF.1	O.FLOW O.DPPM.LOG
FDP_RIP.2	O.COMPROT
FIA_AFL.1	O.IA
FIA_ATD.1	O.IA
FIA_SOS.1	O.IA
FIA_UAU.2	O.IA
FIA_UAU.5	O.IA
FIA_UID.2	O.IA
FIA_USB.1	O.IA
FMT_MOF.1	O.SECURE_POLICY
FMT_MSA.1	O.SECURE_POLICY
FMT_MSA.3	O.SECURE_POLICY
FMT_MTD.1(1)	O.SECURE_POLICY
FMT_MTD.1(2)	O.SECURE_POLICY
FMT_SMF.1	O.SECURE_POLICY
FMT_SMR.1	O.SECURE_POLICY
FPT_STM.1	O.AUDIT
FTA_SSL.1	O.IA
FTA_TAB.1	O.IA
FTP_ITC.1	O.COMPROT

O.AUDIT

The events to be audited are defined in [FAU_GEN.1], and are associated with the identity of the user that caused the event [FAU_GEN.2]. Authorized administrators are provided the capability to read the audit records [FAU_SAR.1] and to be able to select which audit information they want to review [FAU_SAR.3], while all other users are denied access to the audit records [FAU_SAR.2]. The TOE provides a time stamp for use by the audit facility [FPT_STM.1]. The protection of the audit trail ensures the integrity of the stored audit information [FAU_STG.1].

O.COMPROT

The TOE provides a cryptographically-protected communication channel between itself and another trusted IT product [FTP_ITC.1]. The cryptographic functions supporting that secure communication channel are: key generation [all iterations of FCS_CKM.1], key distribution [all iterations of FCS_CKM.2] and cryptographic protocol definition [all iterations of FCS_COP.1]. Destroying of keys is ensured by the residual information protection [FDP_RIP.2].

O.DPPM.LOG

The TOE provides logging capabilities for IP packets [FDP_IFF.1].

O.FLOW

The TOE provides an information flow control mechanism to control the communication between two or more networks attached to the TOE [FDP_IFC.2]. The rules enforced by the information flow control mechanism are driven by a rule set provided with a RAVE application [FDP_IFF.1].

O.IA

The TSF must ensure that only authorized users gain access to the TOE and its resources. Users authorized to access the TOE have to use an identification and authentication process [FIA_UID.2, FIA_UAU.2, FIA_UAU.5]. The TOE maintains various user security attributes [FIA_ATD.1], which are assigned to subjects after proper identification and authentication [FIA_USB.1]. The appropriate strength of the authentication mechanism is ensured [FIA_AFL.1, FIA_SOS.1]. To ensure the protection of the administrative interfaces, the TOE is able to lock interactive sessions [FTA_SSL.1]. The initial banner displayed by the TOE allows the user to obtain information about the TOE [FTA_TAB.1].

O.SECURE_POLICY

The TOE allows the management of different aspects of the TOE [FMT_SMF.1]:

- Management mechanisms of the audit facility are provided [FMT_MOF.1].
- Management of the rule set specified with a RAVE application [FMT_MSA.1, FMT_MSA.3].
- Management of user security attributes and authentication data [FMT_MTD.1(1), FMT_MTD.1(2)].

The TOE maintains different roles which protect various resources, covering a restricted access to the administration functionality listed above [FMT_SMR.1].

6.2.2 Security Requirements Dependency Analysis

The following table shows the dependencies between the different security functional requirements and if they are resolved in this Security Target.

Table 4 Dependencies between Security Functional Requirements

Security Functional Requirement	Dependencies	Resolved (reference to SFR in case of ambiguities)
FAU_GEN.1	FPT_STM.1	Yes
FAU_GEN.2	FAU_GEN.1 FIA_UID.1	Yes FIA_UID.2
FAU_SAR.1	FAU_GEN.1	Yes
FAU_SAR.2	FAU_SAR.1	Yes
FAU_SAR.3	FAU_SAR.1	Yes
FAU_STG.1	FAU_GEN.1	Yes

FCS_CKM.1(1)	FCS_CKM.2 or FCS_COP.1
	FCS_CKM.4
		No
		FCS_CKM.2(1)
		FCS_CKM.2(3)
		FCS_COP.1(1)
		FCS_COP.1(3)

FCS_CKM.1(2)	FCS_CKM.2 or FCS_COP.1 FCS_CKM.4	No FCS_CKM.2(2)
FCS_CKM.1(3)	FCS_CKM.2 or FCS_COP.1 FCS_CKM.4	No FCS_COP.1(2)
FCS_CKM.2(1)	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 FCS_CKM.4	No FCS_CKM.1(1)
FCS_CKM.2(2)	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 FCS_CKM.4	No FCS_CKM.1(2)
FCS_CKM.2(3)	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 FCS_CKM.4	No FCS_CKM.1(1)
FCS_COP.1(1)	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 FCS_CKM.4	No FCS_CKM.1(1)
FCS_COP.1(2)	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 FCS_CKM.4	No FCS_CKM.1(3)
FCS_COP.1(3)	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 FCS_CKM.4	No FCS_CKM.1(1)
FDP_IFC.2	FDP_IFF.1	Yes
FDP_IFF.1	FDP_IFC.1 FMT_MSA.3	Yes FDP_IFC.2

Security Functional Requirement	Dependencies	Resolved (reference to SFR in case of ambiguities)
FDP_RIP.2	No dependencies	Yes
FIA_AFL.1	FIA_UAU.1	Yes FIA_UAU.2
FIA_ATD.1	No dependencies	Yes
FIA_SOS.1	No dependencies	Yes
FIA_UAU.2	FIA_UID.1	Yes FIA_UID.2
FIA_UAU.5	No dependencies	Yes
FIA_UID.2	No dependencies	Yes
FIA_USB.1	FIA_ATD.1	Yes
FMT_MOF.1	FMT_SMF.1 FMT_SMR.1	Yes
FMT_MSA.1	FDP_ACC.1 or FDP_IFC.1 FMT_SMR.1 FMT_SMF.1	Yes
FMT_MSA.3	FMT_MSA.1 FMT_SMR.1	Yes
FMT_MTD.1(1)	FMT_SMR.1 FMT_SMF.1	Yes
FMT_MTD.1(2)	FMT_SMR.1 FMT_SMF.1	Yes
FMT_SMF.1	No dependencies	Yes
FMT_SMR.1	FIA_UID.1	Yes FIA_UID.2
FPT_STM.1	No dependencies	Yes
FTA_SSL.1	FIA_UAU.1	Yes FIA_UAU.2
FTA_TAB.1	No dependencies	Yes
FTP_ITC.1	No dependencies	Yes

Comment

The SFRs of FCS_CKM.1, FCS_CKM.2, and FCS_COP.1 all depend on FCS_CKM.4 requiring the functionality of a key destruction mechanism. The TOE does not implement an explicit key destruction mechanism. The key destruction for the symmetric keys is implicitly performed by the residual information protection mechanism as specified with FDP_RIP.2, which ensures that memory and disk space is cleared before it is reassigned to another subject or object. Concerning the long-term public-private key pairs, the key destruction is performed by deleting the file. As already mentioned, the residual information protection mechanism would clear the disk blocks containing the deleted information of the key material prior to reassignment to subjects or objects. As the TOE generates all required keys itself, all other dependencies are satisfied.

There are no unresolved dependencies between security assurance requirements. This is because the evaluation assurance level EAL4 has been defined such that no unresolved dependencies exist. The additional assurance component ALC_FLR.3 has no dependencies and, therefore, there are no unresolved dependencies for assurance components.

6.3 TOE Security Assurance Requirements

The target evaluation assurance level for the product is EAL4 [CC] augmented by ALC_FLR.3, which is seen as appropriate for a controlled environment where attackers only have an enhanced-basic attack potential.

7 TOE Summary Specification

This chapter describes the CloudShield security functions and associated assurance measures.

7.1 Security audit

The TOE generate audit records for start-up and shutdown of the TOE, as well as for security and system configuration changes. Operations invoked via the WMI, CLI, and GODYN / JSON administrative interfaces provided by the ASM generate audit records.

Due to the update mechanism for CAM (a special memory for the SDB database) via JSON is designed to be a high-speed interface usable by another application only, the TOE does not audit any updates. The updates for CAM imply modifications of the RAVE application behavior. For implementing a full end-to-end auditing, the JSON client must be enabled to audit the modifications to its rule engine. Therefore, the administrator of the TOE must ensure that any user allowed access to the JSON interface (either via the CLI) or via JSONSSL complies with the organizational auditing requirements.

All records of the audit trail are stored in the ASM system database. The WMI and CLI interfaces can be used by authenticated administrators to read from and search through the audit trail.

The auditable events include:

System Access:

- All successful system login and user identification operations
- All unsuccessful system login and user identification operations, including the user identity provided – the TOE logs an unsuccessful login attempt only after an administrator-configurable number of username/password failures
- All successful user authentication operations
- All invalid user authentication attempts

User security management:

- Changes to a user’s account inactivity threshold
- Changes to user’s session inactivity threshold
- Changes to user’s password definition
- Changes made to enable/disable a user’s account
- User account lockout due to invalid login attempts

Security area:

- The startup and termination of the audit functions (specifically, of the TOE)
- Access to the security event records. This includes both successful and unauthorized attempts to access security events.
- Changes made in security profiles and attributes associated with a security group
- Changes in the authentication method used by the TOE
- SNMP Agent additions, modifications, restarts
- System services modifications to enable/disable or change timeout durations
- SSH/SSL Key regeneration operation
- Interface status changes to enable/disable or change timeout durations
- Changes to system banner to enable/disable or change message

Network area:

- System Network interface changes such as IP, Subnet mask, Administrative state.
- System Route table changes
- System Time changes

- NTP Server changes such as additions or modifications, every time synch occurs with NTP server – success/failure.
- Syslog Server changes such as address, controls for forwarding events.
- DNS Server configuration add/change

Hardware area:

- Provisioning (insertion) of modules
- De-provisioning (removal) of module
- Changes to admin state for modules/ports
- Changes to port configuration settings
- Reboot or shutdown of any module
- Execution of diagnostics

System configuration:

- Setup or modification of a remote backup server
- Performing a system backup operation
- Performing a system restore operation
- Performing an update of the system software

Application rule-set configuration:

- Upload of application rule-set file onto the system
- Deletion of application rule-set file from the system
- Import of an application rule-set to the system
- Upload of TCS (traffic control system available on the DPPM-800) configuration file onto the system
- Import of TCS configuration file to the system
- Changing the destination for packet capture files (database, syslog, or file – remote or local)
- Changing the size or frequency of packet capture file creation
- Deletion of packet capture files

Audit records include date and time of the event, type of event, description of the event, user identity (including source IP address if applicable), interface (if applicable), and the outcome of the event. The audit logs are stored as records in the CloudShield database which is located on the ASM. Authorized administrators can review the audit logs only after successfully identifying and authenticating themselves to the Web Management Interface (WMI) or CLI and choosing the appropriate options provided in the drop down menus. The logs are presented in a suitable format in order for the authorized administrators to interpret the information. The logs can be searched and sorted on the following selected fields: user ID, date/time, type of event, and source IP address. The audit logs are protected from unauthorized deletion. Only authorized administrators who have been identified and authenticated have access to the audit functions.

CloudShield provides a time clock which is used to stamp all audit records generated by the TOE.

The Security audit function is designed to satisfy the security functional requirements mapped to O.AUDIT.

7.2 Cryptographic support

The WMI, CLI, and GODYN / JSON administrative interfaces provide secure system management through the use of SSH and TLS (used for protecting HTTP connections) for protection. Publicly available Apache Web Server and OpenSSH code bases are used on the system. The TOE generates, stores, and uses cryptographic keys to support SSH and TLS operations. Specifically,

- The web server supports TLS with the ciphers and keys specified in the SFRs on cryptographic operation.
- SSHv2 clients are supported using ciphers and keys specified in the SFRs on cryptographic operation.

During installation, the TOE automatically generates two sets of asymmetric key pairs on the TOE, one for use with SSH and one for use with TLS. For initial system configuration, administrators use the default serial port interface to then enable both SSH and TLS-protected HTTP – no network service is available during the initial use of the TOE. From the WMI, users can elect to manually re-generate new random key pairs (i.e. no user input to adjust) for either SSH or TLS. Public and private key pairs are stored unencrypted in files on the local ASM disk (no method exists for users to access these files without physical access to the TOE). The asymmetric keys for the SSH protocol are generated internally with the ssh-keygen application that links with the OpenSSL library. This library uses the Linux /dev/urandom device to seed its internal deterministic random number generator. Also, the TLS asymmetric keys are generated using OpenSSL which again uses /dev/urandom as a seed source for its internal deterministic random number generator that provides the input for the key generation process.

A session key is generated per-session and resides in system RAM. The session keys are generated using the OpenSSL deterministic random number generator which is seeded from the Linux /dev/urandom noise source.

The TOE implements an implicit key destruction mechanism. The key destruction for the symmetric keys is implicitly performed by the residual information protection mechanism, which ensures that memory and disk space is cleared before it is reassigned to another subject or object. Concerning the long-term public-private key pairs, the key destruction is performed by deleting the file. As already mentioned, the residual information protection mechanism would clear the disk blocks containing the deleted information of the key material prior to reassignment to subjects or objects.

Encryption is not utilized on the DPPM interfaces nor supported for encrypting or decrypting network content. Management physical network interfaces are provided with separate interfaces on the ASM which results in the fact that encrypted management traffic will not be seen on the DPPM interfaces.

The Cryptographic support function is designed to satisfy the security functional requirements mapped to O.COMPROT.

7.3 Information flow control

When connected in-line, the DPPM blade implements a rule engine that the TOE uses to mediate network traffic between separate networks connected to two different DPPM blade ports. The ASM blade is used to configure DPPM blade information flow rule sets.

When connected in a tap configuration, the TOE does not mediate traffic. Instead it captures packets (i.e. duplicates of original packets provided via span port) that are traveling on a network and processes them. The DPPM blade makes information flow decisions when located either inline or as a tap.

Rule sets are created using the CloudShield PacketWorks IDE application (which is not part of the TOE) on a commodity PC in the operational environment and uploaded to the ASM blade using the WMI or CLI interfaces. Rule sets are created using the CloudShield PacketWorks IDE application by first creating a RAVE project or importing an existing one. These rule sets are linked into an Application Deployment Package (ADP). Using trusted IT products to initiate communication with the TOE, administrators use the WMI or CLI administrative interfaces to upload the ADP that was created using the IDE. Rule sets are saved in the ASM system database and then can subsequently be used to configure the capabilities of a DPPM blade using the WMI or CLI interfaces.

The PacketWorks IDE provides RAVE programmers with the ability to create rule sets that allow the TOE to perform operations to control or alter traffic flow, including the stateful filtering of packets. To ease the development effort for RAVE programmers, the IDE provides a mechanism that is called “virtual patch panel”. This mechanism allows different RAVE code fragments to be “patched together” or linked together to form a single coherent RAVE application.

When a network layer packet is received at a DPPM network interface, the contents of that packet along with metadata about the source DPPM network interface is provided to the configured rule sets. The rule set can inspect packets, modify packets, cause packets to be dropped, and cause packets to be forwarded. These general operations are embodied in the following (more specific) types of functions:

- Packet Routing Controls – RAVE programmers can implement operations to drop or forward packets through the TOE.
- Math and Logic Operations – The RAVE language supports basic math calculations, field comparisons, logical operations, field shifts, etc.
- Execution Controls – The RAVE language provides controls for execution flow through the rule set logic allowing programmers to implement subroutines and modify the logic execution based on value comparisons.

- System Time and Timers – There are three "time" categories that are important to the RAVE developer. Each has varying interactions with the other categories, however, in most cases, remain disconnected in normal usage. Each timer is used for different purposes and accessed differently.
    - Tick-based timers – Tick-based timers are useful for instrumentation of code, measuring application response time, flow duration, or identifying the time at which an event occurred in the network.
    - System Time – The most common use for System Time, within RAVE applications, is to be able to assemble logged packets when captured from across the network on disparate CloudShield systems. Additionally, System Time is used by applications running on the ASM to either commit a new application, or to perform an update using a Data API so that the operation is coordinated to occur on disparate systems at the same time.
    - Global Timers – The global timers are a collection of low granularity timers that are available as global variables and can be referenced by a RAVE application. The timers provide for control of multiple, seconds-based functionality such as flow expiration or seconds-based rate limiters. The timers do not synchronize their starting value with any other clocks.
- Automatic Protocol Decoding – Automatic Protocol Decoding provides RAVE applications the ability to decode and validate major protocol structures. Structures are decoded and validated and the offset to the layers are stored as are the type indicators if possible. The offsets are stored in the system parameters. Most Internet protocols are defined with a fixed format structure residing at the layer offset with an optional extension at the end of the given layer. By knowing the offset to the layer and the structure of the protocol, it is simple to decode fields within a protocol:
    - Layer 2 – Ethernet II, 802.3 Ethernet with or without SNAP, PPP/HDLC
    - Layers 2/3 – MPLS (up to 4 tags)
    - Layer 2/3 – ARP
    - Layer 3 – IPv4
    - Layer 4 – TCP, UDP, ICMP
    - Layers 5–7 – Start of Payload for TCP, UDP, ICMP
- Analyzing Unstructured Data – The CloudShield system and RAVE programming language can perform complex analysis of unstructured packet data. The language provides a means to finding keywords and patterns, as well as a way to map protocol structures onto layer 7. RAVE applications can perform searches against data using regular expression pattern-matching. The Search function takes data from the packet, global, or local variables, and sends it to the data search function. The data can be adapted using pre-processors to make the construction and execution of analysis rules simpler. While regular expressions are a powerful language, using pre-processors is important when designing for speed and pattern-memory consumption.
- Analyzing Structured Data – The CloudShield system and RAVE programming language can perform complex analysis of structured packet data using database table entries.
- Packet Manipulation – The RAVE programming language supports direct packet modification through mechanisms to remove and collapse packet data, expand packets with additional content in headers (encapsulation) and payload, overwrite packet data fields, replicate packets, memory copy/set, and memory pattern location. The TOE provides automated checksum recalculation support.
- Capture of Packet Headers and/or Payloads and/or packet meta data – The RAVE Log function provides the ability to instruct the TOE to capture data directly from the dataplane (the CPU processing the Ethernet frames). RAVE programmers may use this function as part of a RAVE ruleset to selectively capture packet data and meta data for storage in files locally on the ASM formatted such that standard packet analyzing tools can interpret them (pcap files). The captured data represents packet header and/or payload data, not to be confused with event or audit log data.

  The RAVE Log function is performed asynchronously and works by setting a flag and associating the parameters with a data block handed off to a control plane processor (processor controlling the dataplane). In addition to the parameters specified and a pointer to the location of the packet, the time at which the Log function was called is stored for reporting in a pcap capture file.

  The RAVE programming language also supports replication of packets and the ability to forward these replicated packets out a DPPM port for off-system collection and storage. The RAVE programmer can elect to capture all or a portion of the packet data for analysis. To accelerate this capability, the DPPM supports a Log Accelerator subsystem for automatic replication, re-direction and packet encapsulation for sending the packet to the log server. The Log Accelerator sub-system provides a hardware co-processor on CloudShield DPPMs that makes copies of packets and modifies (encapsulates) them for routing, switching and load balancing without requiring RAVE software replication overhead. This function offers improved performance for RAVE applications that use the capture port for logging data by moving the overhead for these functions to hardware. Any RAVE application that requires the following functions can use the Log Accelerator function to perform the same function instead of implementing it in RAVE code:

  - Packet re-direct and Packet replication
  - Ethernet encapsulation of SONET and SDH packets
  - GRE encapsulation of Ethernet and/or SONET and/or SDH packets
  - Load balancing of log packets over multiple MAC addresses
  - MAC address re-write of Ethernet packets

- Fragment and Stream reassembly – The Stream Processor Accelerator (SPA) subsystem provides RAVE applications the ability to re-assemble streams, analyze and manipulate the aggregated stream in real-time. While the CS-2000 database and its large matrices provide the storage and functions required to support re-assembly, the SPA capability is focused on improving the performance, reducing the complexity and driving broad use of stream re-assembly in RAVE applications.
- External Event Reporting -- Rule sets can be created that generate alarms (implemented using the RAVE “Log” function) on the DPPM; the DPPM sends the alarm information to the ASM; the ASM sends the alarm information to an SNMP trap, local or remote syslog, local or remote log file, or a database according to rule configuration.

These functions can be combined in a RAVE application to define requirements that must be satisfied in order for the information flow to be allowed. That is, a received packet is used as input that is analyzed and/or manipulated by a RAVE application. The RAVE application determines whether the packet (i.e. the information flow) is to be retransmitted or not.

For example, a RAVE application can be written to only allow information flows from a presumed address if the Layer-4 protocol being used is ICMP. A more complex example would allow information flows to be redirected to a new destination address, whenever the original destination address in a Layer-4 UDP packet matched a certain subnet, changing the information flow from its original nature into a syslog entry.

The TOE ensures that memory used for storing objects is fully overwritten prior or during allocation.

The User data protection function is designed to satisfy the security functional requirements mapped to O.FLOW, O.DPPM.LOG as well as FDP_RIP.1.

7.4 Identification and authentication

All protocols listed in section 0 provided by the ASM blade require administrators to login before access to the requested management interfaces is allowed. The DPPM blade executing a rule set enforces an unauthenticated information flow control policy and does not authenticate network traffic senders and receivers on connected networks. The TOE defines administrators in terms of:

- user identity
- password
- group membership
- group permissions

There is only one pre-defined group called SU (for Super User). Administrators who are members of the SU group can access all WMI, CLI, and GODYN / JSON type administrative interfaces. The TOE provides the ability to create new groups and to assign groups permissions to perform individual management tasks. Users inherit the permissions of the groups to which they are assigned and must be assigned to at least one group when a user’s account is created.

The TOE provides an authentication mechanism that verifies that secrets meet the requirement: for each attempt to authenticate, the probability that a random attempt will succeed is less than one in 1,000,000. The mechanism provided accomplishes this by requiring that passwords contain at least eight characters but no more than 20 and the session is locked after three (three is the default – or administrator configurable integer between 1 and 8) unsuccessful authentication attempts or by using an SSL certificate. Administrators can also require that passwords contain at least one special character.

The administrator can configure to lock interactive sessions after a specified period of inactivity. A user has to re-authenticate with the username and password if the session was locked.

When gaining access to the TOE, a banner page is displayed informing the user about important data.

The Identification and authentication function is designed to satisfy the security functional requirements mapped to O.IA.

7.5 Security management

The ASM blade provides mechanisms to control access to TOE among a variety of administrative users. Users when they are created are placed into "groups", where each group is granted one of three levels of privilege (read/write, read-only, or none) to one or more of the TOE's permission types:

- Hardware-type permissions – This type of permission allows viewing health type information for DPPM and ASM blade hardware, and managing operational and administrative settings of system ports.
- Network-type permissions – This type of permission allows configuring network information for the ASM blade management ports.
- Software-type permissions – This type of permission allows viewing of exported variable data generated during rule set processing.
- Security-type permissions – This type of permission allows viewing and modifying user and group attributes, configuring network services for the ASM blade management ports, modifying the system banner message, and viewing audit trail data from the audit records.
- Database-type permissions – This type of permission allows reading configuration information directly from the ASM blade database. Please note: The TOE only supports read-only access for this type.
- Configuration-type permissions – This type of permission allows provisioning of import rule sets that permit or deny information flows, provisioning system maintenance activities, and managing data capture.

Access to the TOE is controlled by defining groups and their privileges, and by assigning users to one or more of the groups. The TOE only displays those interfaces that a user has permissions to. The TOE includes pre-defined groups. The groups that are pre-defined in the CS-2000 cannot be modified or removed. The Super User (SU) group is granted Read/Write privileges to all management areas. The super user is created by default to ensure that the customer has access to the system. There can only be one member of the Super User group logged into a system at a time. Groups can be defined in terms of a group name, read/write, read-only, or none to each of the TOE's permission types.

The WMI, CLI and GODYN / JSON type administrative interfaces provided by the ASM blade provide interfaces that can be used to manage DPPM and ASM blade functions. Administrators who are members of the SU group can access all WMI, CLI, and GODYN / JSON type administrative interfaces. Users possessing administrator-defined roles (i.e. users who are members of administrator-defined groups) can perform management tasks for which the group has been granted permission by the administrator.

The system restricts the ability to manage the security audit logs by a privilege setting assigned to an administrator role. Only authorized administrators who have been identified and authenticated have access to the audit functions. Using the WMI and CLI management interfaces, authorized security administrators have the ability to;

- View all information related to a security audit log. The system ensures that no user without the proper authorization is able to view the security audit logs. Any unauthorized attempt results in a security audit log. The logs may be sorted in ascending/descending order or by time, type, source IP address, or user name to facilitate searches.
- Generate a security audit log file to upload for off-system archival and analysis.
- Delete audit log entries and audit log files. The audit logs are protected from unauthorized deletion. Only authorized administrators who have been identified and authenticated have the ability to delete audit log records and files.

The Security management function is designed to satisfy the security functional requirements mapped to O.SECURE_POLICY.

8 References

[CC]	Common Criteria for Information Technology Security Evaluation, CCIMB-2007-09-001 to CCIMB-2007-09-003, Version 3.1 Revision 2, September 2007, Part 1 to 3
[SSHTRANS]	RFC 4253: The Secure Shell (SSH) Transport Layer Protocol, http://www.ietf.org/rfc/rfc4253.txt
[TLS]	RFC 2246: The TLS Protocol Version 1.0, http://www.ietf.org/rfc/rfc2246.txt