QRadar Security Intelligence Platform (NDcPP10) Security Target

Version 0.8
September 19, 2017

Prepared for:

IBM, Corporation

1 New Orchard Road,
Armonk, New York 10504

Prepared By:

Gossamer
Laboratories

www.gossamersec.com

LIST OF TABLES

1. Security Target 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 QRadar Security Intelligence Platform provided by IBM, Corporation. The TOE is being evaluated as a network devices.

The Security Target contains the following additional sections:

- Conformance Claims (Section 2)
- Security Objectives (Section 3)
- Extended Components Definition (Section 4)
- Security Requirements (Section 5)
- TOE Summary Specification (Section 6)

Conventions

The following conventions have been applied in this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
  - Iteration: allows a component to be used more than once with varying operations. In the ST, iteration is indicated by a parenthetical number placed at the end of the component. For example FDP_ACC.1(1) and FDP_ACC.1(2) indicate that the ST includes two iterations of the FDP_ACC.1 requirement.
  - Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]). Note that an assignment within a selection would be identified in italics and with embedded bold brackets (e.g., [***[selected-assignment]***]).
  - Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., [***selection***]).
  - Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …").
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

1.1 Security Target Reference

ST Title – QRadar Security Intelligence Platform (NDcPP10) Security Target

ST Version – Version 0.8

ST Date – September 19, 2017

1.2 TOE Reference

TOE Identification – IBM, Corp. QRadar Security Intelligence Platform, Version 7.2.7

Dell Model 3128C which uses the x86 64-bit CPU architecture

TOE Developer – IBM, Corp.

Evaluation Sponsor – IBM, Corp.

1.3 TOE Overview

The Target of Evaluation (TOE) is QRadar Security Intelligence Platform.

IBM Security QRadar Security Intelligence Platform is also known as the IBM QRadar Security Information and Event Management (SIEM). The QRadar SIEM is a network device intended to detect potential threats through the review of audit and event data collected from network sources. The TOE is the QRadar SIEM. The TOE is administered either locally or remotely. The QRadar product consolidates log source event data from thousands of devices endpoints and applications distributed throughout a network.

1.4 TOE Description

IBM QRadar SIEM consolidates log source event data from device endpoints and applications that are distributed throughout a network. QRadar performs intermediate normalization and correlation activities on this raw data and can forward data to another network server when so configured. Communication with network peers for outbound log/event data is accomplished using TLS protected communication channels. QRadar is capable of providing an X.509v3 certificate to authenticate itself as part of an outbound TLS connection.

The QRadar SIEM utilizes a cryptographic security kernel library internally (referred to as CSK version 1.0). The CSK version 1.0 has received CAVP algorithm certs as shown in Table 6-1 CSK CAVP Certificates.

QRadar provides its cryptographic features through a Java implementation (QCrypto), which utilizes bridge software to invoke OpenSSL cryptographic functions. The OpenSSL library included in the TOE is OpenSSL 1.0.1e. The Cryptographic Security Kernel (CSK) version 1.0 utilizes OpenSSL library to provide cryptographic functions. Thus, all cryptographic functions except those associated with a TOE update validation are provided by OpenSSL (including those originating within the Java implementation).

The TOE includes support for GNU Privacy Guard (GPG), which is a public key software package. GPG is used to verify signatures on product updates.

1.4.1 TOE Architecture

The evaluated product is a single All-in-one device running QRadar SIEM w/ QFlow enabled. A QRadar QFlow collectors collect network traffic passively through network taps and span ports. A QFlow collector can detect and collect information from networked applications. The All-in-One device is a self-contained appliance running the QRadar SIEM in a Red Hat RHEL 6.7 environment. The appliance makes only those interfaces offered by QRadar available.

The IBM All-In-One: Dell 3128C, model utilizes an x86 64-bit CPU architecture, with 4 network interface cards, and varying amounts of memory.

The All-In-One device can connect to an external audit server allowing QRadar to transmit audit and event data to an external server. All outbound audit data is transferred using TLS protected communication channels.

An IBM QRadar All-In-One device provides a trusted path to remote administrators using an HTTPS protected web GUI or SSH protected Command Line Interface (CLI). The QRadar system offers a CLI at the local console and remotely via SSH as an administrative interface. QRadar also offers a web interface for additional administrative functionality. A single device will have four (4) network connections which can be used either for remote management, receipt of event/syslog data, transmission of audit data, or other network support traffic (e.g., NTP, DNS). A REST API interface is offered by QRadar and can be protected by HTTPS/TLS.

1.4.1.1 Physical Boundaries

The TOE is composed of one physical component that is accessed and managed by administrators from computers in the environment.

The TOE can be configured to forward its audit records to an external syslog server in the environment. All audit records sent to the external syslog server are protected using a TLS channel.

The TOE can be configured to synchronize it internal clock using an NTP server in the operational environment.

1.4.1.2 Logical Boundaries

This section summarizes the security functions provided by QRadar:

- Security audit
- Cryptographic support
- Identification and authentication
- Security management
- Protection of the TSF
- TOE access
- Trusted path/channels

1.4.1.2.1 Security audit

The TOE generates logs for a wide range of security relevant events. The TOE can be configured to store the logs locally so they can be accessed by an administrator and also to send the logs to a designated network peer using TLS to protect data while in transit. The TOE is also capable of acting as a log storage device and receiving TLS protected communication from network peers sending audit/event data.

1.4.1.2.2 Cryptographic support

The TOE utilizes NIST validated cryptographic algorithms to support key management, random bit generation, encryption/decryption, digital signature and secure hashing and key-hashing features in support of higher level cryptographic protocols including SSH and TLS.

1.4.1.2.3 Identification and authentication

The TOE requires users to be identified and authenticated before they can use functions mediated by the TOE, with the exception of reading the login banner. The TOE authenticates administrative users. In order for an administrative user to access the TOE, a user account including a user name and password must be created for the user.

1.4.1.2.4 Security management

The TOE provides Command Line Interface (CLI) commands and an HTTP over TLS (HTTPS) Graphical User Interface (GUI) to access the wide range of security management functions to manage its security policies. All administrative activity and functions including security management commands are limited to authorized users (i.e., administrators) only after they have provided acceptable user identification and authentication data to the TOE. The security management functions are controlled through the use of privileges associated with roles that can be assigned to TOE users.

1.4.1.2.5 Protection of the TSF

The TOE implements a number of features designed to protect itself to ensure the reliability and integrity of its security features.

It protects particularly sensitive data such as stored passwords and cryptographic keys so that they are not accessible even by an administrator. It also provides its own timing mechanism to ensure that reliable time information is available (e.g., for log accountability).

Note that the TOE is a single appliance, and as such, no intra-TOE communication is subject to any risks that may require special protection (e.g., cryptographic mechanisms).

The TOE includes functions to perform self-tests so that it might detect when it is failing. It also includes support for GNU Privacy Guard (GPG) is a public key software package. GPG is used to verify signatures on product updates. The GPG signature of an update is verified against a published GPG key for IBM which is installed in the TOE.

1.4.1.2.6 TOE access

The TOE can be configured to display a logon banner before a user session is established. The TOE also enforces inactivity timeouts for local and remote sessions.

1.4.1.2.7 Trusted path/channels

The TOE protects communication channels between itself and remote administrators using HTTPS/TLS and SSH. The SSH protocol is used to protect administrative connections utilizing the TOE's command line interface (CLI). Additionally, a web-based GUI is available for remote administration and is protected using HTTP over TLS (HTTPS).

The TOE also protects communication with network peers using TLS. Protected communication includes the TOE's outbound connection to an external audit server.

1.4.2 TOE Documentation

The TOE offers a series of documents that describe the installation process for the TOE as well as guidance for subsequent use and administration of the applicable security features. The following is a list of such documents.

- IBM QRadar Common Criteria for NIAP, Version 7.2.7

2. Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 4, September 2012.
  - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1 Revision 4, September 2012.
  - Part 3 Conformant
- Package Claims:
  - collaborative Protection Profile for Network Devices, Version 1.0, 27 February 2015 (NDcPP10)

2.1 Conformance Rationale

The ST conforms to the NDcPP10. As explained previously, the security problem definition, security objectives, and security requirements have been drawn from the PP.

This ST incorporates the following NIAP technical decisions:

- 116
- 125
- 130
- 154
- 156
- 167
- 169

3. Security Objectives

The Security Problem Definition may be found in the NDcPP10 and this section reproduces only the corresponding Security Objectives for operational environment for reader convenience. The NDcPP10 offers additional information about the identified security objectives, but that has not been reproduced here and the NDcPP10 should be consulted if there is interest in that material.

In general, the NDcPP10 has defined Security Objectives appropriate for network devices and as such are applicable to the QRadar Security Intelligence Platform TOE.

3.1 Security Objectives for the Operational Environment

OE.ADMIN_CREDENTIALS_SECURE The administrator's credentials (private key) used to access the TOE must be protected on any other platform on which they reside.

OE.NO_GENERAL_PURPOSE There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.

OE.NO_THRU_TRAFFIC_PROTECTION The TOE does not provide any protection of traffic that traverses it. It is assumed that protection of this traffic will be covered by other security and assurance measures in the operational environment.

OE.PHYSICAL Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.

OE.TRUSTED_ADMIN TOE Administrators are trusted to follow and apply all guidance documentation in a trusted manner.

OE.UPDATES The TOE firmware and software is updated by an administrator on a regular basis in response to the release of product updates due to known vulnerabilities.

4. Extended Components Definition

All of the extended requirements in this ST have been drawn from the NDcPP10. The NDcPP10 defines the following extended requirements and since they are not redefined in this ST the NDcPP10 should be consulted for more information in regard to those CC extensions.

Extended SFRs:

- FAU_STG_EXT.1: Protected Audit Event Storage
- FAU_STG_EXT.3: Display warning for local storage space
- FCS_HTTPS_EXT.1: HTTPS Protocol
- FCS_RBG_EXT.1: Random Bit Generation
- FCS_SSHS_EXT.1: SSH Server Protocol
- FCS_TLSC_EXT.2: TLS Client Protocol with authentication
- FCS_TLSS_EXT.1: TLS Server Protocol
- FIA_PMG_EXT.1: Password Management
- FIA_UAU_EXT.2: Password-based Authentication Mechanism
- FIA_UIA_EXT.1: User Identification and Authentication
- FIA_X509_EXT.1: X.509 Certificate Validation
- FIA_X509_EXT.2: X.509 Certificate Authentication
- FIA_X509_EXT.3: X.509 Certificate Requests
- FPT_APW_EXT.1: Protection of Administrator Passwords
- FPT_SKP_EXT.1: Protection of TSF Data (for reading of all symmetric keys)
- FPT_TST_EXT.1: TSF testing
- FPT_TUD_EXT.1: Trusted update
- FTA_SSL_EXT.1: TSF-initiated Session Locking

5. Security Requirements

This section defines the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) that serve to represent the security functional claims for the Target of Evaluation (TOE) and to scope the evaluation effort.

The SFRs have all been drawn from the NDcPP10. The refinements and operations already performed in the NDcPP10 are not identified (e.g., highlighted) here, rather the requirements have been copied from the NDcPP10 and any residual operations have been completed herein. Of particular note, the NDcPP10 made a number of refinements and completed some of the SFR operations defined in the Common Criteria (CC) and that PP should be consulted to identify those changes if necessary.

The SARs are also drawn from the NDcPP10 which includes all the SARs for EAL 1. However, the SARs are effectively refined since requirement-specific 'Assurance Activities' are defined in the NDcPP10 that serve to ensure corresponding evaluations will yield more practical and consistent assurance than the EAL 1 assurance requirements alone. The NDcPP10 should be consulted for the assurance activity definitions.

5.1 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by QRadar Security Intelligence Platform TOE.

Requirement Class	Requirement Component
FAU: Security audit	FAU_GEN.1: Audit Data Generation
	FAU_GEN.2: User identity association
	FAU_STG_EXT.1: Protected Audit Event Storage
	FAU_STG_EXT.3: Display warning for local storage space
FCS: Cryptographic support	FCS_CKM.1: Cryptographic Key Generation
	FCS_CKM.2: Cryptographic Key Establishment
	FCS_CKM.4: Cryptographic Key Destruction
	FCS_COP.1(1): Cryptographic Operation (AES Data Encryption/Decryption)
	FCS_COP.1(2): Cryptographic Operation (Signature Generation and Verification)
	FCS_COP.1(3): Cryptographic Operation (Hash Algorithm)
	FCS_COP.1(4): Cryptographic Operation (Keyed Hash Algorithm)
	FCS_HTTPS_EXT.1: HTTPS Protocol
	FCS_RBG_EXT.1: Random Bit Generation
	FCS_SSHS_EXT.1: SSH Server Protocol
	FCS_TLSC_EXT.2: TLS Client Protocol with authentication
	FCS_TLSS_EXT.1: TLS Server Protocol
FIA: Identification and authentication	FIA_PMG_EXT.1: Password Management
	FIA_UAU.7: Protected Authentication Feedback
	FIA_UAU_EXT.2: Password-based Authentication Mechanism
	FIA_UIA_EXT.1: User Identification and Authentication
	FIA_X509_EXT.1: X.509 Certificate Validation
	FIA_X509_EXT.2: X.509 Certificate Authentication
	FIA_X509_EXT.3: X.509 Certificate Requests
FMT: Security management	FMT_MOF.1(1): Management of security functions behavior - Trusted Update
	FMT_MOF.1(3): Management of security functions behavior - Audit
	FMT_MOF.1(4): Management of security functions behavior - Audit

	FMT_MOF.1(7): Management of security functions behavior - Local Audit Space
	FMT_MTD.1(1): Management of TSF Data
	FMT_SMF.1: Specification of Management Functions
	FMT_SMR.2: Restrictions on Security Roles
FPT: Protection of the TSF	FPT_APW_EXT.1: Protection of Administrator Passwords
	FPT_SKP_EXT.1: Protection of TSF Data (for reading of all symmetric keys)
	FPT_STM.1: Reliable Time Stamps
	FPT_TST_EXT.1: TSF testing
	FPT_TUD_EXT.1: Trusted update
FTA: TOE access	FTA_SSL.3: TSF-initiated Termination
	FTA_SSL.4: User-initiated Termination
	FTA_SSL_EXT.1: TSF-initiated Session Locking
	FTA_TAB.1: Default TOE Access Banners
FTP: Trusted path/channels	FTP_ITC.1: Inter-TSF trusted channel
	FTP_TRP.1: Trusted Path

Table 5-1 TOE Security Functional Components

5.1.1 Security audit (FAU)

5.1.1.1 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 shut-down of the audit functions;

b) All auditable events for the not specified level of audit; and

c) All administrative actions comprising:

- Administrative login and logout (name of user account shall be logged if individual user accounts are required for administrators).
- Security related configuration changes (in addition to the information that a change occurred it shall be logged what has been changed).
- Generating/import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference shall be logged).
- Resetting passwords (name of related user account shall be logged).
- Starting and stopping services (if applicable).
- ***no other actions***];

d) Specifically defined auditable events listed in Table 1.

Table 5-2 Audit Events

Requirement	Auditable Events	Additional Content
FAU_GEN.1	None	None
FAU_GEN.2	None	None
FAU_STG_EXT.1	None	None
FAU_STG_EXT.3	Warning about low storage space for audit events.	None
FCS_CKM.1	None	None
FCS_CKM.2	None	None
FCS_CKM.4	None	None
FCS_COP.1(1)	None	None

FCS_COP.1(2)	None	None
FCS_COP.1(3)	None	None
FCS_COP.1(4)	None	None
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session.	Reason for failure.
FCS_RBG_EXT.1	None	None
FCS_SSHS_EXT.1	Failure to establish an SSH session. Successful SSH rekey.	Reason for failure. Non-TOE endpoint of connection (IP Address).
FCS_TLSC_EXT.2	Failure to establish a TLS Session.	Reason for failure.
FCS_TLSS_EXT.2	Failure to establish a TLS Session.	Reason for failure.
FIA_PMG_EXT.1	None	None
FIA_UAU.7	None	None
FIA_UAU_EXT.2	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UIA_EXT.1	All use of identification and authentication mechanism.	Provided user identity, origin of the attempt (e.g., IP address).
FIA_X509_EXT.1	Unsuccessful attempt to validate a certificate.	Reason for failure.
FIA_X509_EXT.2	None	None
FIA_X509_EXT.3	None	None
FMT_MOF.1(1)	Any attempt to initiate a manual update.	None
FMT_MOF.1(3)	Modification of the behavior of the transmission of audit data to an external IT entity.	None
FMT_MOF.1(4)	Modification of the behavior of the handling of audit data.	None
FMT_MOF.1(7)	Modification of the behavior of the audit functionality when Local Audit Storage Space is full.	None
FMT_MTD.1(1)	All management activities of TSF data.	None
FMT_SMF.1	None	None
FMT_SMR.2	None	None
FPT_APW_EXT.1	None	None
FPT_SKP_EXT.1	None	None
FPT_STM.1	Changes to time.	The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).
FPT_TST_EXT.1	None	None
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure).	None
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None
FTA_SSL.4	The termination of an interactive session.	None
FTA_SSL_EXT.1	Any attempts at unlocking of an interactive session.	None
FTA_TAB.1	None	None
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions.	Identification of the claimed user identity.

FAU_GEN.1.2

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

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and
b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, information specified in column three of Table 1.

5.1.1.2 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.

5.1.1.3 Protected Audit Event Storage (FAU_STG_EXT.1)

FAU_STG_EXT.1.1

The TSF shall be able to transmit the generated audit data to an external IT entity using a trusted channel according to FTP_ITC.1.

FAU_STG_EXT.1.2

The TSF shall be able to store generated audit data on the TOE itself.

FAU_STG_EXT.1.3

The TSF shall [***overwrite previous audit records according to the following rule: [overwrite audit data as defined by configured log file rotation settings]***] when the local storage space for audit data is full.

5.1.1.4 Display warning for local storage space (FAU_STG_EXT.3)

FAU_STG_EXT.3.1

The TSF shall generate a warning to inform the user before the local space to store audit data is used up and/or the TOE will lose audit data due to insufficient local space.

5.1.2 Cryptographic support (FCS)

5.1.2.1 Cryptographic Key Generation (FCS_CKM.1)

FCS_CKM.1.1

The TSF shall generate asymmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm: [

- ***RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.3,***
- FFC schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.1].

5.1.2.2 Cryptographic Key Establishment (FCS_CKM.2)

FCS_CKM.2.1

The TSF shall perform cryptographic key establishment in accordance with a specified cryptographic key establishment method: [

- ***RSA-based key establishment schemes that meets the following: NIST Special Publication 800-56B, 'Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography',***
- Finite field-based key establishment schemes that meets the following: NIST Special Publication 800-56A, 'Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography'].

5.1.2.3 Cryptographic Key Destruction (FCS_CKM.4)

FCS_CKM.4.1

The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method

- For plaintext keys in volatile storage, the destruction shall be executed by a **[*destruction of reference to the key directly followed by a request for garbage collection*];**
- For plaintext keys in non-volatile storage, the destruction shall be executed by the invocation of an interface provided by a part of the TSF that [***logically addresses the storage location of the key and performs a [single] overwrite consisting of [a new value of the key]***]

that meets the following: No Standard. (TD0130 applied)

5.1.2.4 Cryptographic Operation (AES Data Encryption/Decryption) (FCS_COP.1(1))

FCS_COP.1(1).1

The TSF shall perform encryption/decryption in accordance with a specified cryptographic algorithm AES used in [***CBC***] mode and cryptographic key sizes [***128 bits, 256 bits***] that meet the following: AES as specified in ISO 18033-3, [***CBC as specified in ISO 10116***].

5.1.2.5 Cryptographic Operation (Signature Generation and Verification) (FCS_COP.1(2))

FCS_COP.1(2).1

The TSF shall perform cryptographic signature services (generation and verification) in accordance with a specified cryptographic algorithm [

- ***RSA Digital Signature Algorithm and cryptographic key sizes (modulus) [2048 bits]***] that meet the following: [***For RSA schemes: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5, ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3***]. (TD0116 applied)

5.1.2.6 Cryptographic Operation (Hash Algorithm) (FCS_COP.1(3))

FCS_COP.1(3).1

The TSF shall perform cryptographic hashing services in accordance with a specified cryptographic algorithm [***SHA-1, SHA-256, SHA-512***] that meet the following: ISO/IEC 10118-3:2004.

5.1.2.7 Cryptographic Operation (Keyed Hash Algorithm) (FCS_COP.1(4))

FCS_COP.1(4).1

The TSF shall perform keyed-hash message authentication in accordance with a specified cryptographic algorithm [***HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-512***] and cryptographic key sizes [***160, 256, 512***] and message digest sizes [***160, 256, 512***] bits that meet the following: ISO/IEC 9797-2:2011, Section 7 'MAC Algorithm 2'.

5.1.2.8 HTTPS Protocol (FCS_HTTPS_EXT.1)

FCS_HTTPS_EXT.1.1

The TSF shall implement the HTTPS protocol that complies with RFC 2818.

FCS_HTTPS_EXT.1.2

The TSF shall implement HTTPS using TLS.

FCS_HTTPS_EXT.1.3 The TSF shall establish the connection only if [***the peer initiates handshake***]. (TD0125 applied)

5.1.2.9 Random Bit Generation (FCS_RBG_EXT.1)

FCS_RBG_EXT.1.1

The TSF shall perform all deterministic random bit generation services in accordance with ISO/IEC 18031:2011 using [***CTR_DRBG (AES)***].

FCS_RBG_EXT.1.2

The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [***[one] software-based noise source***] with a minimum of [***256 bits***] of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011 Table C.1 'Security Strength Table for Hash Functions', of the keys and hashes that it will generate.

5.1.2.10 SSH Server Protocol (FCS_SSHS_EXT.1)

FCS_SSHS_EXT.1.1

The TSF shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, 4254, and [***no other RFCs***].

FCS_SSHS_EXT.1.2

The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, password-based.

FCS_SSHS_EXT.1.3

The TSF shall ensure that, as described in RFC 4253, packets greater than [**256K**] bytes in an SSH transport connection are dropped.

FCS_SSHS_EXT.1.4

The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: aes128-cbc, aes256-cbc, [***no other algorithms***].

FCS_SSHS_EXT.1.5

The TSF shall ensure that the SSH transport implementation uses [***ssh-rsa***] and [***no other public key algorithms***] as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SSHS_EXT.1.6

The TSF shall ensure that the SSH transport implementation uses [***hmac-sha1***] and [***no other MAC algorithms***] as its MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHS_EXT.1.7

The TSF shall ensure that [***diffie-hellman-group14-sha1***] and [***no other methods***] are the only allowed key exchange methods used for the SSH protocol.

FCS_SSHS_EXT.1.8

The TSF shall ensure that within SSH connections the same session keys are used for a threshold of no longer than one hour, and no more than onegigabyte of transmitted data. After either of the thresholds are reached a rekey needs to be performed. (TD0167 applied)

5.1.2.11 TLS Client Protocol with authentication (FCS_TLSC_EXT.2)

FCS_TLSC_EXT.2.1

The TSF shall implement [***TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)***] supporting the following ciphersuites:

Mandatory Ciphersuites:

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268

Optional Ciphersuites: [

- ***TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268,***
- ***TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268,***
- ***TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268,***
- ***TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,***
- ***TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246,***
- ***TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,***
- ***TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246***].

FCS_TLSC_EXT.2.2

The TSF shall verify that the presented identifier matches the reference identifier according to RFC 6125.

FCS_TLSC_EXT.2.3

The TSF shall only establish a trusted channel if the peer certificate is valid.

FCS_TLSC_EXT.2.4

The TSF shall present the Supported Elliptic Curves Extension in the Client Hello with the following NIST curves: [***none***] and no other curves.

FCS_TLSC_EXT.2.5

The TSF shall support mutual authentication using X.509v3 certificates.

5.1.2.12 TLS Server Protocol (FCS_TLSS_EXT.1)

FCS_TLSS_EXT.1.1

The TSF shall implement [***TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)***] supporting the following ciphersuites:

Mandatory Ciphersuites:

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268

Optional Ciphersuites: [

- ***TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268,***
- ***TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268,***
- ***TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268,***
- ***TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,***
- ***TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246,***
- ***TLS_DHE_RSA_WITH_AES_128_CBC_ HA256 as defined in RFC 5246,***
- ***TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246***].

FCS_TLSS_EXT.1.2

The TSF shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0, and [***none***]. (TD0156 applied)

FCS_TLSS_EXT.1.3

The TSF shall generate key establishment parameters using RSA with key size 2048 bits and [***no other size***] and [***Diffie-Hellman parameters of size 2048 bits and [no other size]***].

5.1.3 Identification and authentication (FIA)

5.1.3.1 Password Management (FIA_PMG_EXT.1)

FIA_PMG_EXT.1.1

The TSF shall provide the following password management capabilities for administrative passwords:

a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [***”!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, and “)”***] ;

b) Minimum password length shall be settable by the Security Administrator, and shall support passwords of 15 characters or greater.

5.1.3.2 Protected Authentication Feedback (FIA_UAU.7)

FIA_UAU.7.1

The TSF shall provide only obscured feedback to the administrative user while the authentication is in progress at the local console.

5.1.3.3 Password-based Authentication Mechanism (FIA_UAU_EXT.2)

FIA_UAU_EXT.2.1

The TSF shall provide a local password-based authentication mechanism, [***[SSH public-key-based authentication mechanism]***] to perform administrative user authentication.

5.1.3.4 User Identification and Authentication (FIA_UIA_EXT.1)

FIA_UIA_EXT.1.1

The TSF shall allow the following actions prior to requiring the non-TOE entity to initiate the identification and authentication process: - Display the warning banner in accordance with FTA_TAB.1; - [***no other actions***]

FIA_UIA_EXT.1.2

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

5.1.3.5 X.509 Certificate Validation (FIA_X509_EXT.1)

FIA_X509_EXT.1.1

The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation.
- The certificate path must terminate with a trusted CA certificate.
- The TSF shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates.
- The TSF shall validate the revocation status of the certificate using [***a Certificate Revocation List (CRL) as specified in RFC 5759***]. (TD0169 applied)
- The TSF shall validate the extendedKeyUsage field according to the following rules:
  - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field
  - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field
  - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field
  - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.

FIA_X509_EXT.1.2

The TSF shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

5.1.3.6 X.509 Certificate Authentication (FIA_X509_EXT.2)

FIA_X509_EXT.2.1

The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [***TLS, HTTPS***], and [***no additional uses***].

FIA_X509_EXT.2.2

When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall [***not accept the certificate***].

5.1.3.7 X.509 Certificate Requests (FIA_X509_EXT.3)

FIA_X509_EXT.3.1

The TSF shall generate a Certificate Request Message as specified by RFC 2986 and be able to provide the following information in the request: public key and [***Common Name, Organization, Organizational Unit, Country***].

FIA_X509_EXT.3.2

The TSF shall validate the chain of certificates from the Root CA upon receiving the CA Certificate Response.

5.1.4 Security management (FMT)

5.1.4.1 Management of security functions behavior - Trusted Update (FMT_MOF.1(1))

FMT_MOF.1(1).1

The TSF shall restrict the ability to enable the functions to perform manual update to Security Administrators.

5.1.4.2 Management of security functions behavior - Audit (FMT_MOF.1(3))

FMT_MOF.1(3).1

The TSF shall restrict the ability to determine the behavior of, modify the behavior of the functions transmission of audit data to an external IT entity to Security Administrators.

5.1.4.3 Management of security functions behavior - Audit (FMT_MOF.1(4))

FMT_MOF.1(4).1

The TSF shall restrict the ability to determine the behavior of, modify the behavior of the functions handling of audit data to Security Administrators.

5.1.4.4 Management of security functions behavior - Local Audit Space (FMT_MOF.1(7))

FMT_MOF.1(7).1

The TSF shall restrict the ability to determine the behavior of, modify the behavior of the functions audit functionality when Local Audit Storage Space is full to Security Administrators.

5.1.4.5 Management of TSF Data (FMT_MTD.1(1))

FMT_MTD.1(1).1

The TSF shall restrict the ability to manage the TSF data to Security Administrators.

5.1.4.6 Specification of Management Functions (FMT_SMF.1)

FMT_SMF.1.1

The TSF shall be capable of performing the following management functions:

- Ability to administer the TOE locally and remotely;
- Ability to configure the access banner;
- Ability to configure the session inactivity time before session termination or locking;
- Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates;
- [***Ability to configure audit behavior,***
- ***Ability to configure the cryptographic functionality***].

5.1.4.7 Restrictions on Security Roles (FMT_SMR.2)

FMT_SMR.2.1

The TSF shall maintain the roles: - Security Administrator.

FMT_SMR.2.2

The TSF shall be able to associate users with roles.

FMT_SMR.2.3

The TSF shall ensure that the conditions - The Security Administrator role shall be able to administer the TOE locally; - The Security Administrator role shall be able to administer the TOE remotely are satisfied.

5.1.5 Protection of the TSF (FPT)

5.1.5.1 Protection of Administrator Passwords (FPT_APW_EXT.1)

FPT_APW_EXT.1.1

The TSF shall store passwords in non-plaintext form.

FPT_APW_EXT.1.2

The TSF shall prevent the reading of plaintext passwords

5.1.5.2 Protection of TSF Data (for reading of all symmetric keys) (FPT_SKP_EXT.1)

FPT_SKP_EXT.1.1

The TSF shall prevent reading of all pre-shared keys, symmetric keys, and private keys.

5.1.5.3 Reliable Time Stamps (FPT_STM.1)

FPT_STM.1.1

The TSF shall be able to provide reliable time stamps.

5.1.5.4 TSF testing (FPT_TST_EXT.1)

FPT_TST_EXT.1.1

The TSF shall run a suite of the following self-tests [***during initial start-up (on power on)***] to demonstrate the correct operation of the TSF: [***cryptographic known answer, self-tests***].

5.1.5.5 Trusted update (FPT_TUD_EXT.1)

FPT_TUD_EXT.1.1

The TSF shall provide Security Administrators the ability to query the currently executing version of the TOE firmware/software [***no other TOE firmware/software version***]. (TD0154 applied).

FPT_TUD_EXT.1.2

The TSF shall provide Security Administrators the ability to manually initiate updates to TOE firmware/software and [***no other update mechanism***].

FPT_TUD_EXT.1.3

The TSF shall provide means to authenticate firmware/software updates to the TOE using a [***digital signature***] prior to installing those updates.

5.1.6 TOE access (FTA)

5.1.6.1 TSF-initiated Termination (FTA_SSL.3)

FTA_SSL.3.1

Refinement: The TSF shall terminate a remote interactive session after a Security Administrator-configurable time interval of session inactivity.

5.1.6.2 User-initiated Termination (FTA_SSL.4)

FTA_SSL.4.1

Refinement: The TSF shall allow Administrator-initiated termination of the Administrator's own interactive session.

5.1.6.3 TSF-initiated Session Locking (FTA_SSL_EXT.1)

FTA_SSL_EXT.1.1

The TSF shall, for local interactive sessions, [***terminate the session***] after a Security Administrator-specified time period of inactivity.

5.1.6.4 Default TOE Access Banners (FTA_TAB.1)

FTA_TAB.1.1

Refinement: Before establishing an administrative user session the TSF shall display a Security Administrator-specified advisory notice and consent warning message regarding use of the TOE.

5.1.7 Trusted path/channels (FTP)

5.1.7.1 Inter-TSF trusted channel (FTP_ITC.1)

FTP_ITC.1.1

The TSF shall be capable of using [***TLS***] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: audit server, [***[none]***] that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.

FTP_ITC.1.2

The TSF shall permit the TSF, or the authorized IT entities to initiate communication via the trusted channel.

FTP_ITC.1.3

The TSF shall initiate communication via the trusted channel for [***export to an audit server***].

5.1.7.2 Trusted Path (FTP_TRP.1)

FTP_TRP.1.1

The TSF shall be capable of using [***SSH, TLS, HTTPS***] to provide a communication path between itself and authorized remote administrators that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from disclosure and provides detection of modification of the channel data.

FTP_TRP.1.2

The TSF shall permit remote administrators to initiate communication via the trusted path.

FTP_TRP.1.3

The TSF shall require the use of the trusted path for initial administrator authentication and all remote administration actions.

5.2 TOE Security Assurance Requirements

The SARs for the TOE are the components as specified in Part 3 of the Common Criteria. Note that the SARs have effectively been refined with the assurance activities explicitly defined in association with both the SFRs and SARs.

Requirement Class	Requirement Component
ADV: Development	ADV_FSP.1: Basic functional specification
AGD: Guidance documents	AGD_OPE.1: Operational user guidance
	AGD_PRE.1: Preparative procedures
ALC: Life-cycle support	ALC_CMC.1: Labelling of the TOE
	ALC_CMS.1: TOE CM coverage
ATE: Tests	ATE_IND.1: Independent testing - conformance

AVA: Vulnerability assessment	AVA_VAN.1: Vulnerability survey

Table 5-3 **Assurance Components**

5.2.1 Development (ADV)

5.2.1.1 Basic functional specification (ADV_FSP.1)

ADV_FSP.1.1d

The developer shall provide a functional specification.

ADV_FSP.1.2d

The developer shall provide a tracing from the functional specification to the SFRs.

ADV_FSP.1.1c

The functional specification shall describe the purpose and method of use for each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.2c

The functional specification shall identify all parameters associated with each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.3c

The functional specification shall provide rationale for the implicit categorisation of interfaces as SFR-non-interfering.

ADV_FSP.1.4c

The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

ADV_FSP.1.1e

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

ADV_FSP.1.2e

The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

5.2.2 Guidance documents (AGD)

5.2.2.1 Operational user guidance (AGD_OPE.1)

AGD_OPE.1.1d

The developer shall provide operational user guidance.

AGD_OPE.1.1c

The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2c

The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3c

The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4c

The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5c

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

AGD_OPE.1.6c

The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfil the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7c

The operational user guidance shall be clear and reasonable.

AGD_OPE.1.1e

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

5.2.2.2 Preparative procedures (AGD_PRE.1)

AGD_PRE.1.1d

The developer shall provide the TOE including its preparative procedures.

AGD_PRE.1.1c

The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

AGD_PRE.1.2c

The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

AGD_PRE.1.1e

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

AGD_PRE.1.2e

The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

5.2.3 Life-cycle support (ALC)

5.2.3.1 Labelling of the TOE (ALC_CMC.1)

ALC_CMC.1.1d

The developer shall provide the TOE and a reference for the TOE.

ALC_CMC.1.1c

The TOE shall be labelled with its unique reference.

ALC_CMC.1.1e

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

5.2.3.2 TOE CM coverage (ALC_CMS.1)

ALC_CMS.1.1d

The developer shall provide a configuration list for the TOE.

ALC_CMS.1.1c

The configuration list shall include the following: the TOE itself; and the evaluation evidence required by the SARs.

ALC_CMS.1.2c

The configuration list shall uniquely identify the configuration items.

ALC_CMS.1.1e

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

5.2.4 Tests (ATE)

5.2.4.1 Independent testing - conformance (ATE_IND.1)

ATE_IND.1.1d

The developer shall provide the TOE for testing.

ATE_IND.1.1c

The TOE shall be suitable for testing.

ATE_IND.1.1e

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

ATE_IND.1.2e

The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

5.2.5 Vulnerability assessment (AVA)

5.2.5.1 Vulnerability survey (AVA_VAN.1)

AVA_VAN.1.1d

The developer shall provide the TOE for testing.

AVA_VAN.1.1c

The TOE shall be suitable for testing.

AVA_VAN.1.1e

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

AVA_VAN.1.2e

The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.1.3e

The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

6. TOE Summary Specification

This chapter describes the security functions:

- Security audit
- Cryptographic support
- Identification and authentication
- Security management
- Protection of the TSF
- TOE access
- Trusted path/channels

6.1 Security audit

The TOE acts as a network device generating audit data. The TOE can be configured to forward audit data to a remote network device. This forwarding can include its own audit data. Thus, the TOE can be configured to send its own audit data to an external network peer using a trusted channel protected by TLS.

All audit logs are stored in plain text and are archived and compressed when the audit log file reaches 200 MB. The current log file is named audit.log. When the file reaches 200 MB, the file is compressed and renamed to audit.1.gz. The file number increments each time that a log file is archived. QRadar stores up to 50 archived log files. These audit files are accessible only to authenticated administrators.

The Security audit function is designed to satisfy the following security functional requirements:

- FAU_GEN.1: Auditing is always being generated when the TOE is running. Thus, the start and stop of auditing corresponds to boot and shutdown of the system. The boot and shutdown of the system are audited. The TOE audits the events identified in Table 5-2.

  Audit records generated by the TOE include a date and time, event type, success or failure indication. For actions that can be assigned to a specific user or network entity, a subject identifier (e.g., username or network address) in included within the audit record. Audit records identified by Table 5-2 contain the additional information described by column 3 of Table 5-2.

- FAU_GEN.2: For actions that can be assigned to a specific user or network entity, a subject identifier (e.g., username or network address) in included within the audit record.
- FAU_STG_EXT.1: The TOE can store audit data as well as transmit audit data to an external audit server through a trusted channel protected by TLS. The external audit server can be any server supporting the syslog protocol tunneled within TLS. All audit logs when stored internally are stored in plain text. These logs are archived and compressed when the audit log file size reaches 200 MB. The current log file is named audit.log. When the file reaches 200 MB, the file is compressed and renamed to audit.log.1.gz. The file number increments each time that a log file is archived. QRadar stores up to 50 archived log files. These log file rotation settings are controlled by internal settings and are not available for modification through the administrative interface.
- FAU_STG_EXT.3: A warning is issued by the TOE before the local storage space for the audit log is full. This message is written to the audit trail and states that the audit log file has reached maximum capacity and will be overwritten.

6.2 Cryptographic support

The TOE utilizes cryptographic support from IBM QRadar Cryptographic Security Kernel (CSK) library, version 1.0. The CSK library version 1.0 that is included within the IBM QRadar software version 7.2 product is unchanged in

QRadar version 7.2.7. The CSK is a multi-algorithm library providing general-purpose cryptographic services. The CSK includes the libcrypto/Openssl library version 1.0.1e for underlying cryptographic operations. The purpose of the CSK library is to provide a single API for cryptographic functionality that can provide centralized control over FIPS-Approved mode status, provide availability of only CAVP/FIPS-Approved algorithms or vendor-affirmed implementations of non FIPS-Approved algorithms, and provide for centralized logging and reporting of the cryptographic engine. The TOE utilizes the CSK for a cryptographic operations it performs (including those associated with TLS, certificate operations, system integrity, and update verification). The libcrypto/Openssl library used by the CSK has obtained CAVP certificates as shown in the following table.

The TOE runs on the QRadar 3128C Appliance, which includes an Intel® Xeon® E5-2650 CPU and operating system support from Red Hat Enterprise Linux server release 6.7.

Table 6-1 CSK CAVP Certificates

Functions	Standards	SFR	CAVP Certificate
Encryption/Decryption			
AES CBC (128 and 256 bits)	FIPS Pub 197 NIST SP 800-38A ISO 18033-3 and ISO 10116	FCS_COP.1(1)	AES #4735
Cryptographic Signature Services & Key Generation			
RSA 2048-bit Signature Gen & Verify Key Gen	FIPS Pub 186-4 ISO/IEC 9796-2	FCS_CKM.1 FCS_COP.1(2)	RSA #2584
DSA Key Gen (FFC)	FIPS PUB 186-4	FCS_CKM.1	DSA #1267
Cryptographic hashing			
SHA-1/256/512	FIPS Pub 180-3 ISO/IEC 10118-3:2004	FCS_COP.1(3)	SHS #3880
Keyed-hash message authentication			
HMAC-SHA-1, HMAC_SHA2-256, HMAC-SHA2-512 (digest sizes and block sizes of 160, 256, and 512 bits)	FIPS Pub 198-1 FIPS Pub 180-3 ISO/IEC 9797-2:2011	FCS_COP.1(4)	HMAC #3152
Random bit generation			
CTR_DRBG (AES)	FIPS SP 800-90B ISO/IEC 18031:2011	FCS_RBG_EXT.1	DRBG #1621
Component Validation List			
FFC	NIST SP 800-56A	FCS_CKM.2	CVL #1368

The TOE also includes GNU GPG to perform verification of RSA signatures on product updates in support of FPT_TUD_EXT.1. GPG utilizes the libgcrypt library (version 1.4.5-11), which has obtained CAVP certs (see Table 6-2) to demonstrate functions supporting signature verification surrounding trusted updates.

Table 6-2 GPG libgcrypt library CAVP certificates

Functions	Standards	SFR	CAVP Certificate
Cryptographic signature services			
RSA 2048-bit Signature Gen & Verify	FIPS Pub 186-4 ISO/IEC 9796-2	FCS_COP.1(2)	RSA #2590
Cryptographic hashing			

SHA-1, SHS-256	FIPS Pub 180-3 ISO/IEC 10118-3:2004	FCS_COP.1(3)	SHS #3886

These supporting cryptographic functions are provided by the TOE to support the SSHv2 (compliant with RFCs 4251, 4252, 4253, and 4254), TLSv1.1 (compliant with RFC 4346), and TLSv1.2 (compliant with RFC 5289) secure communication protocols. The TOE supports the TLS ciphersuites identified below.

The TOE supports the following TLS ciphersuites to connect with downstream audit servers.

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_ SHA256
- TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256
- TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256

The TOE allows inbound TLS connections using only the following ciphersuites.

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_ SHA256
- TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256
- TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256

The TOE supports the secret keys, private keys and CSPs as shown in Table 6-3 Cryptographic Keys.

Table 6-3 Cryptographic Keys

Key or CSP:	Zeroized upon:	Stored in:	Zeroized by:
SSH host RSA private key	Administrative action	/root/.ssh/id_rsa	Generating new key, or uninstalling the module
SSH host RSA public key	Administrative action	/root/.ssh/id_rsa.pub	Generating new key, or uninstalling the module
SSH session key	Connection Termination	RAM	API call, power cycle, or host reboot
TLS host RSA private key	Administrative action	/etc/httpd/conf/certs/cert.key	Generating new private key, or uninstalling the module
TLS host RSA digital certificate	Administrative action	/etc/httpd/conf/certs/cert.cert	Generating new certificate, or uninstalling the module

TLS pre-master secret	Process Restart (Manual)	RAM	API call, power cycle, or host reboot
TLS session key	Process Restart	RAM	API call, power cycle, or host reboot
DH Private Exponent	Process Restart	RAM	API call, power cycle, or host reboot
DH Public Key	Process Restart	RAM	API call, power cycle, or host reboot
Administrator/User Passwords	User-driven account management	On disk, in hashed form	Setting new password, or uninstalling the module
GPG Update key	Install/Setup and Updates	Hard coded, internal to shared library	Uninstalling the module
DRBG Seed	System Reboot	RAM	API call, power cycle, or host reboot
DRBG Value V	System Reboot	RAM	API call, power cycle, or host reboot
DRBG Constant C	System Reboot	RAM	API call, power cycle, or host reboot

The TOE stores all persistent secret and private keys on disk and stores all ephemeral keys in RAM (as indicated in the above table). Additionally, the TOE is designed to zeroize secret and private keys when they are no longer required by the TOE as detailed below. The TOE's zeroization has been subjected to FIPS 140 validation. Note that zeroization occurs as shown in column 4 of Table 6-3.

The Cryptographic support function is designed to satisfy the following security functional requirements:

- FCS_CKM.1: The TOE generates RSA keys of size 2048-bits using the CSK functions outlined in Table 6-1 CSK CAVP Certificates.
- FCS_CKM.2: The TOE generates RSA keys for use with both inbound and outbound TLS using key establishment schemes identified by Table 6-1 CSK CAVP Certificates.
- FCS_CKM.4: The TOE stores all persistent secret and private keys on disk and stores all ephemeral keys in RAM (as indicated in the above table). The TOE clears these keys (i.e., plaintext keys in volatile storage), by destroying the reference to the key and requesting a garbage collection action. Additionally, the TOE is designed to zeroize secret and private keys when they are no longer required by the TOE as detailed in Table 6-3 Cryptographic Keys. The TOE's zeroization has been subjected to FIPS 140 validation. Note that zeroization occurs as follows: 1) when deleted from disk, the previous value is overwritten once with zeroes; 2) when added or changed on disk, any old value is overwritten completely with the new value; and, 3) the zeroization of values in RAM is achieved by overwriting once with zeroes.
- FCS_COP.1(1): The TOE performs AES encryption and decryption. Refer to Table 6-1 for the specifics regarding this cryptographic functions and relevant CAVP certificates.
- FCS_COP.1(2): The TOE performs cryptographic signature generation and verification using 2048-bit RSA schemes per FIPS 186-4 as indicated by the CAVP certificates shown in Table 6-1.

- FCS_COP.1(3): The TOE performs cryptographic hashing using the algorithms shown in Table 6-1. Refer to Table 6-1 for the specifics regarding this cryptographic functions and relevant CAVP certificates.
- FCS_COP.1(4): The TOE performs keyed-hashing using the algorithms shown in Table 6-1. Refer to Table 6-1 for the specifics regarding this cryptographic functions and relevant CAVP certificates.
- FCS_HTTPS_EXT.1: An HTTPS/TLS connection is available which presents a Web GUI administrative interface. The TOE implements HTTPS per RFC 2818. A connection can be established only if the peer initiates the connection.
- FCS_RBG_EXT.1: The TOE includes the IBM CTR_DRBG (AES256). This DRBG is used for all keys generated in support of cryptographic operations for TLS and SSH. The DRBG is seeded by drawing 256-bits from the Linux Kernel RNG which is augmented by noise from the JitterEntropy software noise source.
- FCS_SSHS_EXT.1: The TOE supports SSHv2 as defined by RFCs 4251, 4252, 4253, and 4254. The TOE supports SSHv2 with AES (CBC) 128 or 256 bit ciphers, in conjunction with HMAC-SHA-1, RSA for authentication and with diffie-hellman-group14-sha1 for the key exchange method. While other ciphers and hashes are implemented in the product, they are disabled while the TOE is operating in FIPS mode. The TOE also supports RSA public-key authentication for users connecting to the TOE with a public-key.

  The SSHv2 authentication timeout period is 120 seconds allowing clients to retry only 5 times; both public-key and password based authentication can be configured; and packets are limited to 256K bytes. Whenever the timeout period or authentication retry limit is reached, the TOE closes the applicable TCP connection and releases the SSH session resources. As SSH packets are being received, the TOE uses a buffer to build all packet information. Once complete, the packet is checked to ensure it can be appropriately decrypted. However, if it is not complete when the buffer becomes full (256K bytes) the packet will be dropped. An SSH session is rekeyed by the TOE after being established one hour or after transmitting 1Gb of traffic whichever comes first. A client initiated rekey, resets both the time and traffic counters.
- FCS_TLSC_EXT.2: The TOE provides TLS protected export of audit/event data with mutual authentication using X.509v3 certificates. When a TLS server requests TLS authentication via certificates, the TOE is capable of providing a X509v3 certificate.

  The TOE includes support for TLSv1.1 and TLSv1.2 only. No older versions of TLS, and no version of SSL are supported. The TOE supports the ciphersuites identified above. When validating a certificate, the TOE ensures the distinguished name (DN) or Subject Alternate Name (SAN) fields in the certificate match the peer identifier DNS name. The TOE supports the use of wildcard values within a certificate's DN or SAN field. The TOE does not support certificate pinning.
- FCS_TLSS_EXT.1: An HTTPS/TLS connection is available which presents a Web GUI administrative interface. Thus, the TOE acts as a TLS server supporting TLSv1.1 and TLSv1.2 only. No older versions of TLS, and no version of SSL are supported. The TOE supports the ciphersuites identified above.

  TLS v1.2 is supported with AES (CBC) 128 or 256 bit ciphers, in conjunction with SHA-1 and SHA-256 and RSA. The TOE utilizes RSA w/ key sizes of 2048-bits and can support DH key exchanges with those RSA keys.

6.3 Identification and authentication

The Identification and authentication function is designed to satisfy the following security functional requirements:

- FIA_PMG_EXT.1: The TOE supports user management capabilities which allow the administrator to configure a minimum length password (including a length of 15 characters). The TOE also supports us use of uppercase letters, lowercase letters, numeric values and special characters for passwords. The special characters permitted to be used in passwords including "!", "@", "#", "$", %", "^", "&", "*", "(" and ")".
- FIA_UAU.7: The TOE obscures passwords when entered by administrators.

- FIA_UAU_EXT.2: The TOE authenticates administrators using a local password-based mechanism or using a public-key based authentication mechanism (Note public-key authentication is supported only for SSH connections).
- FIA_UIA_EXT.1: The TOE does not offer any services through the administrative interfaces prior to authenticating the connecting user other than the display of a TOE warning banner.
- FIA_X509_EXT.1: The TOE supports the use of CRLs to determine the revocation status of certificates. Certificates are validated as part of the authentication process when they are presented to the TOE and when they are loaded into the TOE. The following fields are verified as appropriate:
    - Expiration – certificate cannot be expired
    - Common Name - Needs to be FQN device name or IP. Wildcards are allowed only for 1 level of sub-domain and not allowed for the main domain.
    - CA Field – must be true if CA certificate
    - Key Usage - Need to have "Certificate Sign" incase of CA certificates and "Digital Signature" in case of identity certificates.
    - X509v3 Extended Key Usage - Need to rightly indicate whether it is for use as "server" certificate or "client" certificate. If incorrect, connection is not allowed
    - X509v3 CRL Distribution Points- Certificate must be valid per a current CRL. If this field is not present, CRL check is not performed.
    - Subject Alternative Method - Not a mandatory attribute. If present, the values stored in this will take priority over the CN in Subject attribute.
    - Basics Constraints - Attribute must be present and must have CA Field
- FIA_X509_EXT.2: The TOE uses X509v3 certificates for HTTPS and TLS communications. Certificates are validated as part of the authentication process when they are presented to the TOE and when they are loaded into the TOE. The TOE attempts to obtain a CRL distribution point(s) from the presented certificate. If none are found within the certificate, and all other checks deem the certificate to be valid, then the TOE accepts the certificate. If there are distribution points within the certificate, the TOE attempts to download the CRL. If either the download fails, the CRL is expired or the certificate is revoked, the TOE does not accept the certificate as valid.
- FIA_X509_EXT.3: An administrator can issue commands to generate a CSR on the TOE. The administrator is asked for information to complete the CSR, including the following:
    - Country Name (2 letter code) [XX]:
    - State or Province Name (full name) []:
    - Locality Name (eg, city) [Default City]:
    - Organization Name (eg, company) [Default Company Ltd]:
    - Organizational Unit Name (eg, section) []:
    - Common Name (eg, your name or your server's hostname) []:
    - Email Address []:

6.4 Security management

The Security management function is designed to satisfy the following security functional requirements:

- FMT_MOF.1(1): Only administrators can initiate a software update through the command line interface only.
- FMT_MOF.1(3): The TOE allows administrators to configure the use of external audit server by the TOE including the use of TLS to protect the communications.
- FMT_MOF.1(4): The TOE provides a capability to define audit data storage limits and retention policies.

- FMT_MOF.1(7): The TOE provides a capability to define audit data storage limits and retention policies that control TOE behavior as local storage becomes full.
- FMT_MTD.1(1): Only security administrators can login to the TOE via its command line or Web GUI interfaces.
- FMT_SMF.1: The TOE provides the ability for security administrators to securely configure and manage the TOE. Administrators can connect to the Command Line Interface (CLI) via either a remote SSH connection or using a locally connected terminal. Administrators perform a majority of administrative tasks using an HTTPS/TLS protected communication channel offering a Web-based GUI. The TOE offers the ability to configure a login banner and an inactivity timeout value that can be used for session termination or session locking. The TOE also provides mechanisms to support an administrator's ability to verify and install TOE updates; to configure audit behavior; and to configure cryptographic functionality.
- FMT_SMR.2: The TOE supports local administration via a CLI or a locally attached network device. Remote administration is provided through a CLI protected by SSH or through a web GUI protected by HTTPS/TLS. The majority of administrative functionality is available only through the web GUI.

6.5 Protection of the TSF

The TOE includes several mechanisms to support self-protection, including protection of sensitive data, accurate time, self-testing, and trusted updates.

The mechanism to support verification of TOE updates utilizes IBM GPG keys. The IBM public key is inserted into the root user's keyring during system hardening/setup. The update is delivered as an executable with an embedded payload. The embedded payload is a GPG signed package. When executed, the payload is verified and if valid it is extracted. Upon failure of the signature check the extracted payload is deleted; upon success there is an indicator that installation can proceed.

The Protection of the TSF function is designed to satisfy the following security functional requirements:

- FPT_APW_EXT.1: Administrator passwords are stored SHA-512 hashed and salted. No interface is offered by the TOE to present cleartext passwords.
- FPT_SKP_EXT.1: Pre-shared keys, symmetric keys, and private keys stored by the TOE are AES encrypted and never presented back to administrators as plaintext. The AES key protecting other keys is never presented or displayed in any tools or UI.
- FPT_STM.1: The TOE uses the system time obtained from the hardware as maintained by an optional NTP server for use in audit records and during certificate validation.

FPT_TST_EXT.1: The TOE runs a suite of self-tests during power-on. These tests include cryptographic known answer tests exercising the algorithms shown in Table 6-1 CSK CAVP Certificates and Table 6-2 GPG libgcrypt library CAVP certificates.

- FPT_TUD_EXT.1: IBM signs product updates using a GPG key that is installed on the TOE. The signature on an update is verified prior to installing the update (as described above). Administrators must manually obtain updates and must take explicit actions to install an update (the TOE does not automatically update itself). The TOE UI presents the currently running version upon request.

6.6 TOE access

The TOE access function is designed to satisfy the following security functional requirements:

- FTA_SSL.3: For remote sessions, the TOE provides an inactivity timeout mechanism that is configurable by an administrator. The timeout mechanism applies to the HTTPS-protected Web GUI and RESTAPI interface, as well as to the SSH-protected CLI interface.

- FTA_SSL.4: All administrative users can issue a logout from a remote administrative session using either the SSH, the Web GUI and the RESTAPI interfaces. Administrators can also logout from a local console session.
- FTA_SSL_EXT.1: The TOE provides an admin-specified timeout for inactivity at the local console.
- FTA_TAB.1: The TOE presents a warning banner to the user during login actions prior to completing authentication on the Web GUI, the SSH interface and the local console.

6.7 Trusted path/channels

The Trusted path/channels function is designed to satisfy the following security functional requirements:

- FTP_ITC.1: Communication with an external audit server (i.e., outbound) will utilize mutually authenticated TLS. The external audit server would be expected to accept TLS communication initiated by the TOE.
- FTP_TRP.1: The TOE provides multiple methods of remote administration. A command line interface is available remotely via an SSH protected channel. Additionally, an HTTPS/TLS connection is available which presents a Web GUI administrative interface and the RESTAPI interface. Administrators must initiate the connection to the TOE from a remote network entity.