Cloud Software Group

NetScaler VPX

Version: 14.1.FIPS

FIPS 140-3 Non-Proprietary Security Policy

FIPS Security Level: 1
Document Version: 0.4

Prepared for:

Cloud Software Group
851 Cypress Creek Road
Fort Lauderdale, FL 33309
United States of America

Phone: +1 954 267 3000
www.cloud.com

Prepared by:

Corsec Security, Inc.
12600 Fair Lakes Circle, Suite 210
Fairfax, VA 22033
United States of America

Phone: +1 703 267 6050
www.corsec.com

Table of Contents

List of Tables

List of Figures

1. General

1.1 Overview

This is a non-proprietary Cryptographic Module Security Policy for the NetScaler VPX (version: 14.1.FIPS) from Cloud Software Group (CSG). This Security Policy describes how the NetScaler VPX meets the security requirements of Federal Information Processing Standards (FIPS) Publication 140-3, which details the U.S. and Canadian government requirements for cryptographic modules. More information about the FIPS 140-3 standard and validation program is available on the National Institute of Standards and Technology (NIST) and the Canadian Centre for Cyber Security (CCCS) Cryptographic Module Validation Program (CMVP) website at http://csrc.nist.gov/groups/STM/cmvp.

This document also describes how to run the module in a secure Approved mode of operation. This policy was prepared as part of the Level 1 FIPS 140-3 validation of the module. The NetScaler VPX is referred to in this document as VPX or the module.

1.1.1 References

This document deals only with operations and capabilities of the module in the technical terms of a FIPS 140-3 cryptographic module security policy. More information is available on the module from the following sources:

- The CSG website www.cloud.com contains information on the full line of services and solutions from CSG.
- The search page on the CMVP website (https://csrc.nist.gov/Projects/cryptographic-module-validation-program/Validated-Modules/Search) can be used to locate and obtain vendor contact information for technical or sales-related questions about the module.

1.1.2 Document Organization

ISO/IEC 19790 Annex B uses the same section naming convention as *ISO/IEC 19790* section 7 - Security requirements. For example, Annex B section B.2.1 is named “General” and B.2.2 is named “Cryptographic module specification,” which is the same as *ISO/IEC 19790* section 7.1 and section 7.2, respectively. Therefore, the format of this Security Policy is presented in the same order as indicated in Annex B, starting with “General” and ending with “Mitigation of other attacks.” If sections are not applicable, they have been marked as such in this document.

1.2 Security Levels

The NetScaler VPX is validated at the FIPS 140-3 section levels shown in the table below.

Section	Title	Security Level
1	General	1
2	Cryptographic module specification	1
3	Cryptographic module interfaces	1
4	Roles, services, and authentication	3

Section	Title	Security Level
5	Software/Firmware security	1
6	Operational environment	1
7	Physical security	N/A
8	Non-invasive security	N/A
9	Sensitive security parameter management	1
10	Self-tests	1
11	Life-cycle assurance	1
12	Mitigation of other attacks	N/A
	Overall Level	1

Table 1: Security Levels

2. Cryptographic Module Specification

2.1 Description

2.1.1 Purpose and Use

The NetScaler product line optimizes delivery of applications over the Internet and private networks. It is an Application Delivery Controller (ADC) that performs application-specific traffic analysis to intelligently distribute, optimize, and secure L4-L7[1] network traffic for web-applications.

The NetScaler VPX is a virtual appliance consisting of a control plane processing function (providing all configuration and management processing functions) and multiple data planes which provide data packet processing functions. All configuration and management activities are performed via the web-based GUI[2], REST[3]ful Nitro API[4], and CLI[5] interfaces. The GUI includes a configuration utility for configuring the appliance as well as a statistical utility called Dashboard.

In a typical installation (see Figure 1 for an illustration of a typical “two-arm” topology), the NetScaler VPX is installed in a data center on-premises or in a public cloud (such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP)) between the clients and the servers so that client requests and server responses pass through it. Virtual servers configured on the appliance provide connection points that clients use to access the applications behind the appliance. In this case, the appliance owns public IP[6] addresses that are associated with its virtual servers, while the real servers are isolated in a private network. The internal customer network hosts all load-balancing and authentication services, such as LDAP[7], Kerberos, and SAML[8]. The module’s features are enabled, and the configured policies are then applied to incoming and outgoing traffic.

[1] L4-L7 – Layer 4 – Layer 7
[2] GUI – Graphical User Interface
[3] REST – Representational State Transfer
[4] API – Application Programming Interface
[5] CLI – Command Line Interface
[6] IP – Internet Protocol
[7] LDAP – Lightweight Directory Access Protocol
[8] SAML – Security Assurance Markup Language

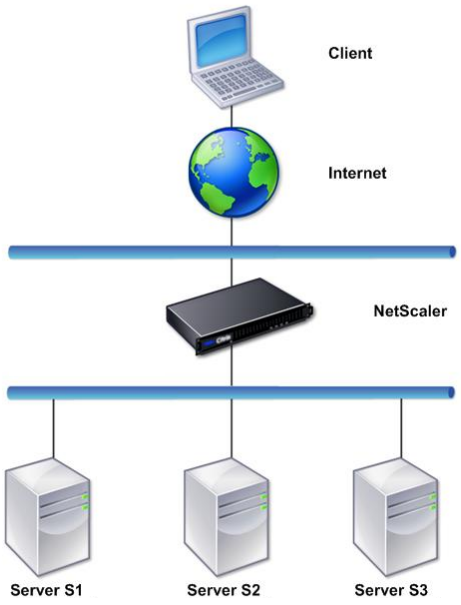

Figure 1. Typical VPX "Two-Arm" Topology

The feature set can be broadly categorized as consisting of switching features, security and protection features, and server-farm optimization features:

- Switching features – When deployed in front of application servers, the NetScaler ensures optimal distribution of traffic by the way in which it directs client requests. Administrators can segment application traffic according to information in the body of an HTTP[9] or TCP[10] request, and on the basis of L4–L7 header information such as URL[11], application data type, or cookie. Numerous load balancing algorithms and extensive server health checks improve application availability by ensuring that client requests are directed to the appropriate servers.

- Security and protection features – NetScaler's security and protection features protect web applications from Application Layer attacks. NetScaler allows legitimate client requests and can block malicious requests. It provides built-in defenses against denial-of-service (DoS) attacks and supports features that protect against legitimate surges in application traffic that would otherwise overwhelm the servers. An available built-in firewall protects web applications from Application Layer attacks, including buffer overflow exploits, SQL[12] injection attempts, cross-site scripting attacks, and more. In addition, the firewall provides identity theft protection by securing confidential corporate information and sensitive customer data.

- Optimization features – Optimization features offload resource-intensive operations, such as SSL[13] processing, data compression, client keep-alive, TCP buffering, and the caching of static and dynamic content from servers. This improves the performance of the servers in the server farm and therefore

[9] HTTP – Hypertext Transfer Protocol
[10] TCP – Transmission Control Protocol
[11] URL – Universal Resource Locator
[12] SQL – Structured Query Language
[13] SSL – Secure Sockets Layer

speeds up applications. NetScaler supports several transparent TCP optimizations, which mitigate problems caused by high latency and congested network links, accelerating the delivery of applications while requiring no configuration changes to clients or servers.

The NetScaler VPX combines all of these capabilities are into a single, integrated virtual appliance.

2.1.2 Module Type

The module is a Software module.

2.1.3 Module Embodiment

The module has a Multi-Chip Standalone embodiment.

2.1.4 Cryptographic Boundary

The module's cryptographic boundary comprises a virtual appliance (shown by the red dotted line in Figure 2) consisting of the VPX virtual appliance software and FreeBSD operating system acting as the guest OS.

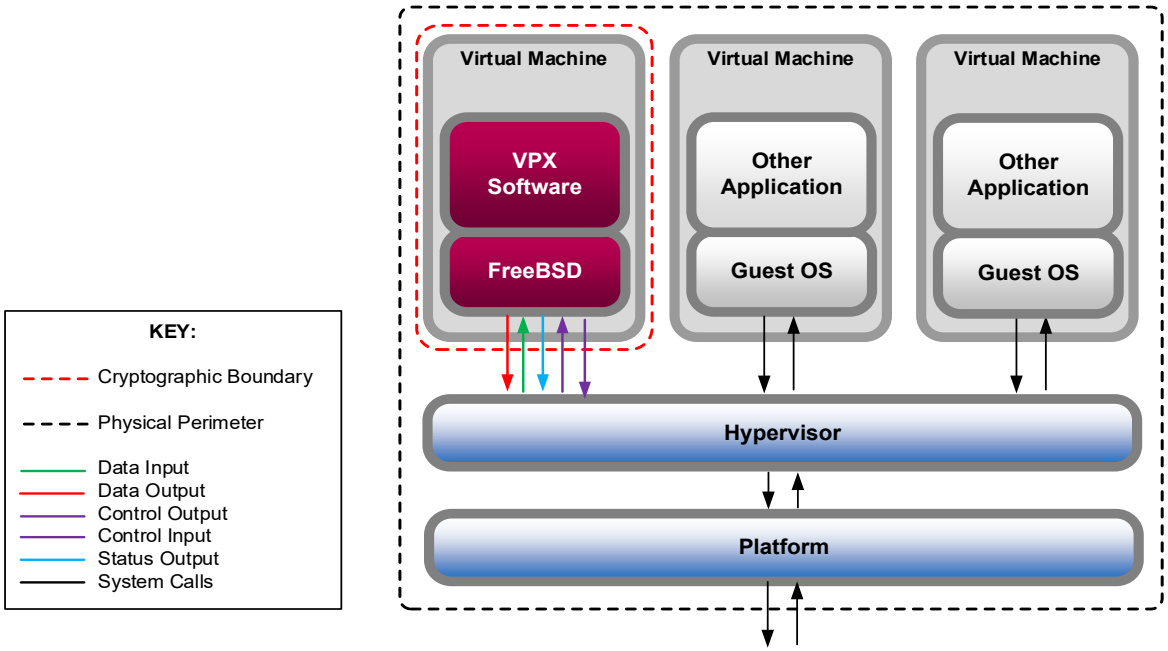

Figure 2. NetScaler VPX Cryptographic Boundary

2.1.5 Tested Operational Environment's Physical Perimeter (TOEPP)

As a virtual appliance, the software module has no physical characteristics; however, the module makes use of the physical interfaces of the server hosting the virtual environment upon which the module is installed. The

hypervisor controls and directs all interactions between the module and the operator and is responsible for mapping the module's virtual interfaces to the host server's physical interfaces.

The module's TOEPP is the hard enclosure of the host server on which it runs. The module was tested on the platforms listed in section 2.2, and each platform consists of a motherboard, a multi-core Intel Xeon CPU, random access memory (RAM), a power supply, and interface ports.

Figure 3 displays the hardware components of the server used for testing (the dashed line surrounding the hardware components represents the TOEPP) and identifies the hardware with which the processor interfaces.

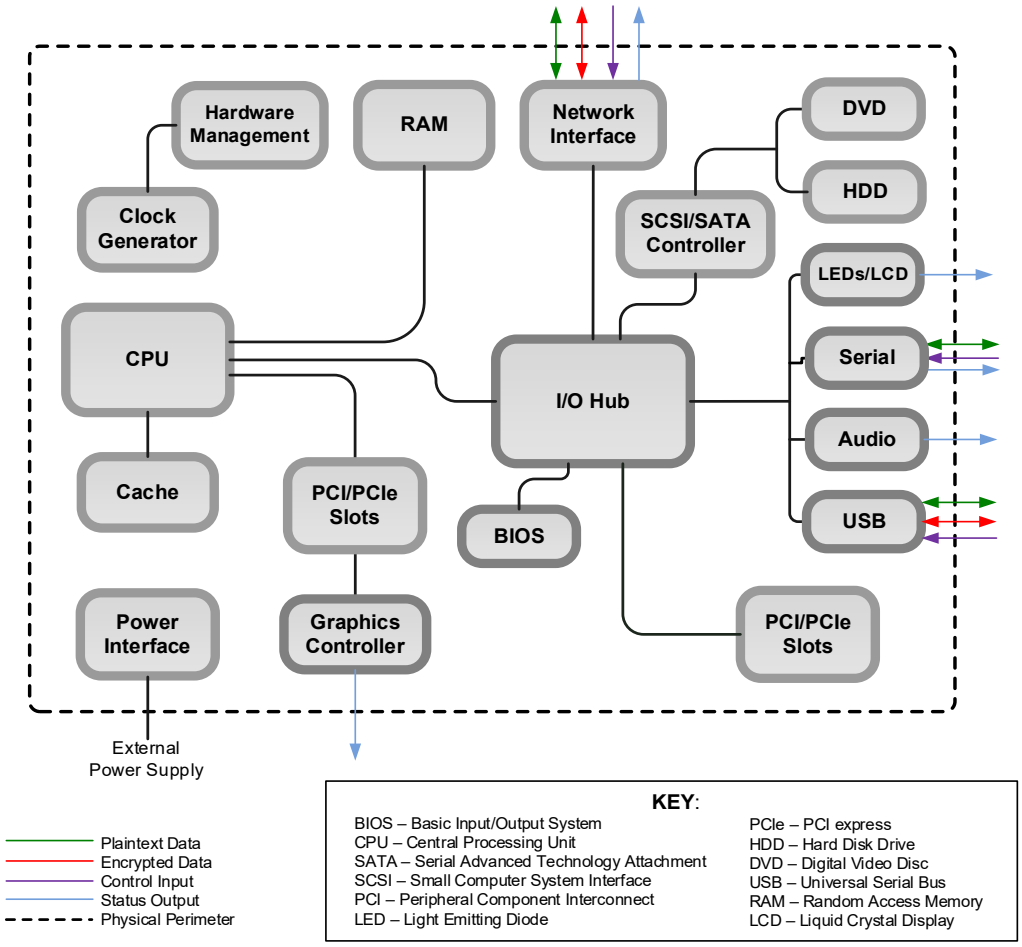

Figure 3. GPC Block Diagram

2.2 Tested and Vendor Affirmed Module Version and Identification

2.2.1 Tested Module Identification – Hardware

Not applicable. The module is a software module.

2.2.2 Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets)

The table below lists the executable code sets of the module.

Package or File Name	Software/ Firmware Version	Features	Integrity Test
build-14.1-47.107.fips_nc_64.tgz	14.1.FIPS	Contains FreeBSD OS and NetScaler VPX software in an image.	2048-bit RSA digital signature verification with SHA2-512

Table 2: Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets)

2.2.3 Tested Module Identification – Hybrid Disjoint Hardware

Not applicable. The module does not have any hybrid disjoint hardware.

2.2.4 Tested Operational Environments – Software, Firmware, Hybrid

The module was tested and found to be compliant with FIPS 140-3 requirements on the environments listed in the table below.

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
FreeBSD 11.4	Dell PowerEdge R630	Intel Xeon E5-2680	Yes	VMware ESXi 8	14.1.FIPS
FreeBSD 11.4	Dell PowerEdge R630	Intel Xeon E5-2680	No	VMware ESXi 8	14.1.FIPS

Table 3: Tested Operational Environments - Software, Firmware, Hybrid

2.2.5 Vendor-Affirmed Operational Environments – Software, Firmware, Hybrid

The table below specifies the vendor-affirmed operational environments claimed.

Operating System	Hardware Platform
FreeBSD 11.4 on Citrix XenServer 7.1 LTSR	Dell PowerEdge R630
FreeBSD 11.4 on Microsoft Hyper-V	Dell PowerEdge R630
FreeBSD 11.4 on KVM w/ Ubuntu 16.04.05 LTS	Dell PowerEdge R630
FreeBSD 11.4 on Citrix XenServer 8.2 LTSR	Lenovo SR630

Table 4: Vendor-Affirmed Operational Environments - Software, Firmware, Hybrid

The module is also supported when running in the following hypervisors on cloud computing platforms:

- Microsoft Azure's hypervisor
- Amazon Web Services' Nitro hypervisor
- Google Cloud Platform's KVM hypervisor

2.3 Excluded Components

The module does not exclude any components from the requirements.

2.4 Modes of Operation

2.4.1 Modes List and Description

The module supports the modes of operation listed in the table below.

Mode Name	Description	Type	Status Indicator
Approved	When installed, initialized, and operated according to the Security Policy, the Approved mode is the only supported mode of operation of the module.	Approved	Global Indicator

Table 5: Modes List and Description

2.5 Algorithms

2.5.1 Approved Algorithms

The module includes the following cryptographic libraries that provide basic cryptographic functionalities and support secure networking protocols:

- NetScaler Control Plane Cryptographic Library version 2.1 (Cert. A7126)
- NetScaler Data Plane Cryptographic Library version 1.0 (Cert. A3943)
- NetScaler CPU Jitter Entropy Source version 3.4.0 (Cert. A3513)

The module implements the Approved algorithms listed in the table below.

Data Plane

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A3943	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-GCM	A3943	Direction - Decrypt, Encrypt IV Generation - Internal IV Generation Mode - 8.2.1 Key Length - 128, 256	SP 800-38D
ECDSA KeyGen (FIPS186-4)	A3943	Curve - P-224, P-256, P-384, P-521 Secret Generation Mode - Testing Candidates	FIPS 186-4
ECDSA KeyGen (FIPS186-5)	A3943	Curve - P-224, P-256, P-384, P-521 Secret Generation Mode - testing candidates	FIPS 186-5
ECDSA KeyVer (FIPS186-4)	A3943	Curve - P-224, P-256, P-384, P-521	FIPS 186-4
ECDSA KeyVer (FIPS186-5)	A3943	Curve - P-224, P-256, P-384, P-521	FIPS 186-5
ECDSA SigGen (FIPS186-4)	A3943	Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512	FIPS 186-4
ECDSA SigGen (FIPS186-5)	A3943	Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512	FIPS 186-5
ECDSA SigVer (FIPS186-4)	A3943	Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512	FIPS 186-4

Algorithm	CAVP Cert	Properties	Reference
ECDSA SigVer (FIPS186-5)	A3943	Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-224, SHA2-256, SHA2-384, SHA2-512	FIPS 186-5
Hash DRBG	A3943	Prediction Resistance - No, Yes Mode - SHA2-256	SP 800-90A Rev. 1
HMAC-SHA-1	A3943	Key Length - Key Length: 8-512 Increment 8	FIPS 198-1
HMAC-SHA2-224	A3943	Key Length - Key Length: 8-512 Increment 8	FIPS 198-1
HMAC-SHA2-256	A3943	Key Length - Key Length: 8-512 Increment 8	FIPS 198-1
HMAC-SHA2-384	A3943	Key Length - Key Length: 8-512 Increment 8	FIPS 198-1
HMAC-SHA2-512	A3943	Key Length - Key Length: 8-512 Increment 8	FIPS 198-1
KAS-ECC-SSC Sp800-56Ar3	A3943	Domain Parameter Generation Methods - P-224, P-256, P-384, P-521 Scheme - ephemeralUnified - KAS Role - initiator, responder	SP 800-56A Rev. 3
KDF TLS (CVL)	A3943	TLS Version - v1.0/1.1, v1.2 Hash Algorithm - SHA2-256, SHA2-384	SP 800-135 Rev. 1
KTS-IFC	A3943	Modulo - 4096 Key Generation Methods - rsakpg1-basic Scheme - KTS-OAEP-basic - KAS Role - initiator, responder Key Transport Method - Key Length - 384	SP 800-56B Rev. 2
RSA SigGen (FIPS186-4)	A3943	Signature Type - PKCS 1.5 Modulo - 2048, 3072, 4096	FIPS 186-4
RSA SigGen (FIPS186-5)	A3943	Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5	FIPS 186-5
RSA SigVer (FIPS186-2)	A3943	Signature Type - PKCS 1.5 Modulo - 4096	FIPS 186-4
RSA SigVer (FIPS186-4)	A3943	Signature Type - PKCS 1.5 Modulo - 1024, 2048, 3072, 4096	FIPS 186-4
RSA SigVer (FIPS186-5)	A3943	Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5	FIPS 186-5
SHA-1	A3943	Message Length - Message Length: 8-51200 Increment 8	FIPS 180-4
SHA2-224	A3943	Message Length - Message Length: 8-51200 Increment 8	FIPS 180-4
SHA2-256	A3943	Message Length - Message Length: 8-51200 Increment 8	FIPS 180-4
SHA2-384	A3943	Message Length - Message Length: 8-65536 Increment 8	FIPS 180-4
SHA2-512	A3943	Message Length - Message Length: 8-65536 Increment 8	FIPS 180-4
TLS v1.2 KDF RFC7627 (CVL)	A3943	Hash Algorithm - SHA2-256, SHA2-384	SP 800-135 Rev. 1
TLS v1.3 KDF (CVL)	A3943	HMAC Algorithm - SHA2-256, SHA2-384 KDF Running Modes - DHE, PSK, PSK-DHE	SP 800-135 Rev. 1

Table 6: Approved Algorithms - Data Plane

Control Plane

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A7126	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-CFB128	A7126	Direction - Decrypt, Encrypt Key Length - 128	SP 800-38A
AES-CTR	A7126	Direction - Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-GCM	A7126	Direction - Decrypt, Encrypt IV Generation - Internal IV Generation Mode - 8.2.1 Key Length - 128, 256	SP 800-38D

Algorithm	CAVP Cert	Properties	Reference
Counter DRBG	A7126	Prediction Resistance - No, Yes Mode - AES-256 Derivation Function Enabled - No, Yes	SP 800-90A Rev. 1
ECDSA KeyGen (FIPS186-5)	A7126	Curve - P-224, P-256, P-384, P-521 Secret Generation Mode - testing candidates	FIPS 186-5
ECDSA KeyVer (FIPS186-5)	A7126	Curve - P-224, P-256, P-384, P-521	FIPS 186-5
ECDSA SigGen (FIPS186-5)	A7126	Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-256, SHA2-384, SHA2-512	FIPS 186-5
ECDSA SigVer (FIPS186-4)	A7126	Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA-1	FIPS 186-4
ECDSA SigVer (FIPS186-5)	A7126	Curve - P-224, P-256, P-384, P-521 Hash Algorithm - SHA2-256, SHA2-384, SHA2-512	FIPS 186-5
HMAC-SHA-1	A7126	Key Length - Key Length: 8-512 Increment 8	FIPS 198-1
HMAC-SHA2-256	A7126	Key Length - Key Length: 8-512 Increment 8	FIPS 198-1
HMAC-SHA2-384	A7126	Key Length - Key Length: 8-512 Increment 8	FIPS 198-1
HMAC-SHA2-512	A7126	Key Length - Key Length: 8-512 Increment 8	FIPS 198-1
KAS-ECC-SSC Sp800-56Ar3	A7126	Domain Parameter Generation Methods - P-224, P-256, P-384, P-521 Scheme - ephemeralUnified - KAS Role - initiator, responder	SP 800-56A Rev. 3
KAS-FFC-SSC Sp800-56Ar3	A7126	Domain Parameter Generation Methods - ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, MODP-2048, MODP-3072, MODP-4096, MODP-6144 Scheme - dhEphem - KAS Role - initiator, responder	SP 800-56A Rev. 3
KDF IKEv1 (CVL)	A7126	Authentication Method - Pre-shared Key Preshared Key Length - Preshared Key Length: 64-504 Increment 8 Diffie-Hellman Shared Secret Length - Diffie-Hellman Shared Secret Length: 2048 Hash Algorithm - SHA-1, SHA2-256, SHA2-384, SHA2-512	SP 800-135 Rev. 1
KDF IKEv2 (CVL)	A7126	Diffie-Hellman Shared Secret Length - Diffie-Hellman Shared Secret Length: 2048 Derived Keying Material Length - Derived Keying Material Length: 1056-3072 Increment 8 Hash Algorithm - SHA-1, SHA2-256, SHA2-384, SHA2-512	SP 800-135 Rev. 1
KDF SNMP (CVL)	A7126	Password Length - Password Length: 64-248 Increment 8	SP 800-135 Rev. 1
KDF SSH (CVL)	A7126	Cipher - AES-128, AES-192, AES-256 Hash Algorithm - SHA2-256, SHA2-512	SP 800-135 Rev. 1
KTS-IFC	A7126	Modulo - 2048 Key Generation Methods - rsakpg1-basic Scheme - KTS-OAEP-basic - KAS Role - initiator, responder Key Transport Method - Key Length - 384	SP 800-56B Rev. 2
PBKDF	A7126	Iteration Count - Iteration Count: 10-10000 Increment 1 Password Length - Password Length: 8-128 Increment 1	SP 800-132
RSA KeyGen (FIPS186-5)	A7126	Key Generation Mode - probableWithProbableAux Modulo - 2048, 3072, 4096 Primality Tests - 2powSecStr Private Key Format - standard	FIPS 186-5
RSA SigGen (FIPS186-5)	A7126	Modulo - 2048, 3072 Signature Type - pkcs1v1.5	FIPS 186-5
RSA SigVer (FIPS186-4)	A7126	Signature Type - PKCS 1.5 Modulo - 2048, 3072	FIPS 186-4
RSA SigVer (FIPS186-5)	A7126	Modulo - 2048, 3072 Signature Type - pkcs1v1.5	FIPS 186-5

Algorithm	CAVP Cert	Properties	Reference
Safe Primes Key Generation	A7126	Safe Prime Groups - ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, MODP-2048, MODP-3072, MODP-4096, MODP-6144	SP 800-56A Rev. 3
Safe Primes Key Verification	A7126	Safe Prime Groups - ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, MODP-2048, MODP-3072, MODP-4096, MODP-6144	SP 800-56A Rev. 3
SHA-1	A7126	Message Length - Message Length: 8-51200 Increment 8	FIPS 180-4
SHA2-256	A7126	Message Length - Message Length: 8-51200 Increment 8	FIPS 180-4
SHA2-384	A7126	Message Length - Message Length: 8-65536 Increment 8	FIPS 180-4
SHA2-512	A7126	Message Length - Message Length: 8-65536 Increment 8	FIPS 180-4
TLS v1.2 KDF RFC7627 (CVL)	A7126	Hash Algorithm - SHA2-256, SHA2-384	SP 800-135 Rev. 1

Table 7: Approved Algorithms - Control Plane

CPU Jitter Entropy Source

Algorithm	CAVP Cert	Properties	Reference
SHA3-256	A3513	Message Length - Message Length: 0-65528 Increment 8	FIPS 202

Table 8: Approved Algorithms - CPU Jitter Entropy Source

2.5.2 Vendor Affirmed Algorithms

The table below lists the vendor-affirmed security methods.

Name	Properties	Implementation	Reference
CKG (Control Plane - VA)	CKG:Symmetric	NetScaler Control Plane Cryptographic Library	SP 800-133 Rev. 2, sections 4 and 6.3
CKG (Data Plane - VA)	CKG:Symmetric	NetScaler Data Plane Cryptographic Library	SP 800-133 Rev. 2, section 4
CKG (KEK)	CKG:Combining keys and other data	NetScaler Control Plane Cryptographic Library	NIST SP 800-133rev2, Section 6.3

Table 9: Vendor-Affirmed Algorithms

2.5.3 Non-Approved, Allowed Algorithms

The module does not implement any non-Approved algorithms allowed in the Approved mode of operation.

N/A for this module.

2.5.4 Non-Approved, Allowed Algorithms with No Security Claimed

The table below lists the non-Approved algorithms implemented by the module that are allowed for use in the Approved mode of operation with no security claimed.

Name	Caveat	Use and Function
MD5 (NetScaler Control Plane Cryptographic Library)	N/A	Message digest in TLS 1.0/1.1 handshake on the control plane
MD5 (NetScaler Data Plane Cryptographic Library)	N/A	Message digest in TLS 1.0/1.1 handshake on the data plane

Table 10: Non-Approved, Allowed Algorithms with No Security Claimed

2.5.5 Non-Approved, Not Allowed Algorithms

The module does not include any non-Approved algorithms that are not allowed in the Approved mode of operation.

N/A for this module.

2.6 Security Function Implementations

The table below lists the security function implementations for this module.

Name	Type	Description	Properties	Algorithms
AES for Disk Encryption	BC-UnAuth	Used for encrypting/decrypting passwords and passphrases using KEK	Publication:NIST SP 800-38A	AES-GCM: (A7126) Key Length: 256
AES for Data Encryption	BC-UnAuth	Used for general encryption/decryption using the AES Key	Publication:NIST SP 800-38A	AES-CBC: (A3943)
CKG (KEK)	CKG	Used for generation of the KEK by combining multiple keys/other data	Publication:SP 800-133 Rev. 2, Section 6.3, NIST SP 800-90B	Counter DRBG: (A7126) SHA3-256: (A3513)
AES for TLS Tickets	BC-UnAuth	Used for encryption and decryption of TLS session resumption tickets using the TLS Ticket Encryption Key	Publication:NIST SP 800-38A	AES-CBC: (A3943) Key Length: 128 Hash DRBG: (A3943)
HMAC for TLS Tickets	MAC	Used for authentication of TLS session resumption tickets for using the TLS Ticket Authentication Key	Publication:FIPS 198-1	HMAC-SHA2-256: (A3943) SHA2-256: (A3943)
Key Generation for SSH Authentication (CP)	AsymKeyPair-KeyGen	Used for generation of SSH Public Key and SSH Private Key for authenticating SSH sessions	Publication:FIPS 186-5	Counter DRBG: (A7126) ECDSA KeyGen (FIPS186-5): (A7126) RSA KeyGen (FIPS186-5): (A7126)

Name	Type	Description	Properties	Algorithms
DH Key Agreement for SSH (CP)	KAS-Full	Used for derivation of the SSH Session Key and SSH Authentication Key (includes shared secret computation and key derivation)	Publication:NIST SP 800-56 Rev. 3 Key Strength:Key establishment methodology provides between 112 and 256 bits of encryption strength Caveat:No part of the SSH protocol, other than the KDF, has been tested by the CAVP and CMVP. IG:IG D.F Scenario 2 path (2), split Key confirmation:no Key derivation:IG 2.4.B SP 800-135rev1 CVL	KDF SSH: (A7126) KAS-FFC-SSC Sp800-56Ar3: (A7126) Safe Primes Key Generation: (A7126) Safe Primes Key Verification: (A7126)
ECDH Key Agreement for SSH (CP)	KAS-Full	Used for derivation of the SSH Session Key and SSH Authentication Key (includes shared secret computation and key derivation)	Publication:FIPS 186-5, NIST SP 800-56 Rev. 3 Key Strength:Key establishment methodology provides between 112 and 256 bits of encryption strength Caveat:No part of the SSH protocol, other than the KDF, has been tested by the CAVP and CMVP. IG:IG D.F Scenario 2 path (2), split Key confirmation:no Key derivation:IG 2.4.B SP 800-135rev1 CVL	KAS-ECC-SSC Sp800-56Ar3: (A7126) Curve: P-256, P-384, P-521 KDF SSH: (A7126) ECDSA KeyGen (FIPS186-5): (A7126) Curve: P-256, P-384, P-521 ECDSA KeyVer (FIPS186-5): (A7126) Curve: P-256, P-384, P-521 Counter DRBG: (A7126) SHA2-256: (A7126) SHA2-384: (A7126) SHA2-512: (A7126)
AES for SSH (CP)	BC-Auth	Used for encryption and decryption of SSH session packets using the SSH Session Key	Publication:NIST SP 800-38A	Counter DRBG: (A7126) AES-GCM: (A7126) Key Length: 128, 256 AES-CTR: (A7126) Key Length: 128, 256
HMAC for SSH (CP)	MAC	Used for authentication of SSH session packets using the SSH Authentication Key	Publication:FIPS 198-1, SP 800-107 Rev. 1 Caveat:The module supports the truncation of HMAC SHA-1 to 96 bits according to NIST SP 800-107 Rev.1	HMAC-SHA2-256: (A7126) HMAC-SHA2-512: (A7126) SHA2-256: (A7126) SHA2-512: (A7126)

Name	Type	Description	Properties	Algorithms
DH Key Agreement for IKE (CP)	KAS-Full	Used for derivation of the IKEv1/IKEv2 Session Keys and IKEv1/IKEv2 Authentication Keys (includes key pair generation, DH shared secret computation, and protocol key derivation)	Publication:NIST SP 800-56 Rev. 3 Key Strength:Key establishment methodology provides between 112 and 176 bits of encryption strength. Caveat:No part of the IKEv1 and IKEv2 protocols, other than the KDFs, have been tested by the CAVP and CMVP. IG:IG D.F Scenario 2 path (2), split Key confirmation:no Key derivation:IG 2.4.B SP 800-135rev1 CVL	Safe Primes Key Generation: (A7126) Safe Prime Groups: MODP-2048, MODP-3072, MODP-6144 Safe Primes Key Verification: (A7126) Safe Prime Groups: MODP-2048, MODP-3072, MODP-6144 KDF IKEv1: (A7126) KDF IKEv2: (A7126) KAS-FFC-SSC Sp800-56Ar3: (A7126) Domain Parameter Generation Methods: MODP-2048, MODP-3072, MODP-6144 Counter DRBG: (A7126) SHA-1: (A7126) SHA2-256: (A7126) SHA2-384: (A7126) SHA2-512: (A3943)
AES for IKE/IPsec (CP)	BC-UnAuth	Used for encryption and decryption of IKE and /IPsec session packets using the IKEv1/IKEv2/IPsec Session Keys	Publication:NIST SP 800-38A	AES-CBC: (A7126) Counter DRBG: (A7126)
HMAC for IKE/IPsec (CP)	MAC	Used for authentication of IKE and /IPsec session packets using the IKEv1/IKEv2/IPsec Authentication Keys	Publication:FIPS 198-1 Caveat:The module supports the truncation of HMAC SHA-1 to 96 bits according to NIST SP 800-107 Rev1	HMAC-SHA-1: (A7126) HMAC-SHA2-256: (A7126) HMAC-SHA2-384: (A7126) HMAC-SHA2-512: (A7126) SHA-1: (A7126) SHA2-256: (A7126) SHA2-384: (A7126)
HMAC for Message Authentication	MAC	Used for message authentication using HMAC key	Publication:FIPS 198-1 Caveat:The module supports the truncation of HMAC SHA-1 to 96 bits according to NIST SP 800-107 Rev.1	HMAC-SHA-1: (A3943) SHA-1: (A3943) SHA2-224: (A3943) HMAC-SHA2-256: (A3943) SHA2-256: (A3943) HMAC-SHA2-384: (A3943) SHA2-384: (A3943) HMAC-SHA2-512: (A3943) HMAC-SHA2-224: (A3943) SHA2-512: (A3943)

Name	Type	Description	Properties	Algorithms
SigGen for TLS Authentication (CP)	DigSig-SigGen	Used to generate ECDSA/RSA signatures during the TLS handshake on the control plane	Publication:FIPS 186-5	ECDSA SigGen (FIPS186-5): (A7126) RSA SigGen (FIPS186-5): (A7126)
SigVer for TLS Authentication (CP)	DigSig-SigVer	Used to verify ECDSA/RSA signatures during the TLS handshake on the control plane	Publication:FIPS 186-5	ECDSA SigVer (FIPS186-5): (A7126) RSA SigVer (FIPS186-5): (A7126)
SigVer for TLS Authentication (CP) (legacy)	DigSig-SigVer	Used to verify ECDSA/RSA signatures during the TLS handshake on the control plane	Publication:FIPS 186-4	ECDSA SigVer (FIPS186-4): (A7126) RSA SigVer (FIPS186-4): (A7126)
SigGen for TLS Authentication (DP)	DigSig-SigGen	Used to generate ECDSA/RSA signatures during the TLS handshake on the data plane	Publication:FIPS 186-4, FIPS 186-5	ECDSA SigGen (FIPS186-5): (A3943) RSA SigGen (FIPS186-5): (A3943) ECDSA SigGen (FIPS186-4): (A3943) RSA SigGen (FIPS186-4): (A3943)
SigVer for TLS Authentication (DP)	DigSig-SigVer	Used to verify ECDSA/RSA signatures during the TLS handshake on the data plane	Publication:FIPS 186-5	ECDSA SigVer (FIPS186-5): (A3943) RSA SigVer (FIPS186-5): (A3943)
SigVer for TLS Authentication (DP) (legacy)	DigSig-SigVer	Used to verify ECDSA/RSA signatures during the TLS handshake on the data plane	Publication:FIPS 186-4	ECDSA SigVer (FIPS186-4): (A3943) Hash Algorithm: SHA-1 Curve: P-224, P-256, P-384, P-521 RSA SigVer (FIPS186-4): (A3943) Properties: Modulo: 1024 Hash Pair: Hash Algorithm: SHA-1

Name	Type	Description	Properties	Algorithms
DH Key Agreement for TLS (CP)	KAS-Full	Used while deriving the TLS Session Key and TLS Authentication Key (includes key pair generation, DH shared secret computation, and protocol key derivation ion)	Publication:RFC 7627 Key Strength:Key establishment methodology provides between 112 and 176 bits of encryption strength. Caveat:No part of the TLS protocols, other than the KDF, has been tested by the CAVP and CMVP. IG:IG D.F Scenario 2 path (2), split Key confirmation:no Key derivation:IG 2.4.B SP 800-135rev1 CVL	TLS v1.2 KDF RFC7627: (A7126) KAS-ECC-SSC Sp800-56Ar3: (A7126) Domain Parameter Generation Methods: ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144 Counter DRBG: (A7126) Safe Primes Key Generation: (A7126) Safe Primes Key Verification: (A7126) Safe Prime Groups: ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144 SHA2-256: (A7126) Safe Prime Groups: ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144 SHA2-384: (A7126)
ECDH Key Agreement for TLS (CP)	KAS-Full	Used to derive the TLS Session Key and TLS Authentication Key (includes key pair generation, ECDH shared secret computation, and protocol key derivation)	Publication:FIPS 186-5, RFC 7627 Key Strength:Key establishment methodology provides between 112 and 256 bits of encryption strength. Caveat:No part of the TLS protocols, other than the KDFs, have been tested by the CAVP and CMVP. IG:IG D.F Scenario 2 path (2), split Key confirmation:no Key derivation:: IG 2.4.B SP 800-135rev1 CVL	TLS v1.2 KDF RFC7627: (A3943) KAS-ECC-SSC Sp800-56Ar3: (A7126) Counter DRBG: (A7126) ECDSA KeyGen (FIPS186-5): (A7126) ECDSA KeyVer (FIPS186-5): (A7126) SHA2-256: (A7126) SHA2-384: (A7126)

Name	Type	Description	Properties	Algorithms
ECDH Key Agreement for TLS (DP)	KAS-Full	Used to derive the TLS Session Key and TLS Authentication Key (includes key pair generation, ECDH shared secret computation, and protocol key derivation)	Publication:FIPS 186-4, FIPS 186-5, RFC 7627, RFC 8446 Key Strength:Key establishment methodology provides between 112 and 256 bits of encryption strength Caveat:No part of the TLS protocols, other than the KDFs, have been tested by the CAVP and CMVP. IG:IG D.F Scenario 2 path (2), split Key confirmation:no Key derivation:IG 2.4.B SP 800-135rev1 CVL	TLS v1.2 KDF RFC7627: (A3943) TLS v1.3 KDF: (A3943) KAS-ECC-SSC Sp800-56Ar3: (A3943) Hash DRBG: (A3943) ECDSA KeyGen (FIPS186-4): (A3943) ECDSA KeyGen (FIPS186-5): (A3943) ECDSA KeyVer (FIPS186-4): (A3943) ECDSA KeyVer (FIPS186-5): (A3943) SHA2-256: (A3943) SHA2-384: (A3943) KDF TLS: (A3943)
RSA Key Transport for TLS (CP)	KTS-Encap	Used for transporting a TLS Session Key and TLS Authentication Key (includes key pair generation and key encapsulation)	Publication:FIPS 186-5, SP 800-56B Rev. 2 Key Strength:Key establishment methodology provides 112 bits of encryption strength	KTS-IFC: (A7126) Modulo: 2048 Counter DRBG: (A7126) RSA KeyGen (FIPS186-5): (A7126)
RSA Key Transport for TLS (DP)	KTS-Encap	Used for transporting a TLS Session Key and TLS Authentication Key (includes key pair generation and key encapsulation)	Publication:FIPS 186-5, SP 800-56B Rev. 2 Key Strength:Key establishment methodology provides 112 bits of encryption strength	KTS-IFC: (A3943) Hash DRBG: (A3943) RSA KeyGen (FIPS186-5): (A7126) Modulo: 2048
AES for TLS (CP)	BC-Auth BC-UnAuth	Used for encryption and decryption of TLS session packets using the TLS Session Key	Publication:FIPS 800-38A, SP 800-38D	AES-CBC: (A7126) Key Length: 128, 256 AES-GCM: (A7126)
AES for TLS (DP)	BC-Auth BC-UnAuth	Used for encryption and decryption of TLS session packets using the TLS Session Key	Publication:FIPS 800-38A, SP 800-38D	AES-CBC: (A3943) Key Length: 128, 256 AES-GCM: (A3943) Key Length: 128, 256
HMAC for TLS (CP)	MAC	Used for authentication of TLS session packets using the TLS Authentication Key	Publication:FIPS 198-1 Caveat:: The module supports the truncation of HMAC SHA-1 to 96 bits according to NIST SP 800-107 Rev.1	HMAC-SHA-1: (A7126) HMAC-SHA2-256: (A7126) HMAC-SHA2-384: (A7126) SHA-1: (A7126) SHA2-256: (A7126) SHA2-384: (A7126)

Name	Type	Description	Properties	Algorithms
HMAC for TLS (DP)	MAC	Used for authentication of TLS session packets using the TLS Authentication Key	Publication:FIPS 198-1 Caveat:The module supports the truncation of HMAC SHA-1 to 96 bits according to NIST SP 800-107 Rev.1	HMAC-SHA-1: (A3943) HMAC-SHA2-256: (A3943) HMAC-SHA2-384: (A3943) SHA-1: (A3943) SHA2-224: (A3943) SHA2-256: (A3943) SHA2-384: (A3943)
RSA SigGen for DNSSec	DigSig-SigGen	Used for generation of an RSA digital signature for DNSSec	Publication:FIPS 186-5	RSA SigGen (FIPS186-5): (A3943) HMAC-SHA2-256: (A3943) SHA2-256: (A3943)
RSA SigVer for DNSSec	DigSig-SigVer	Used to verify an RSA digital signature for DNSSec	Publication:FIPS 186-5	RSA SigVer (FIPS186-5): (A3943) HMAC-SHA2-256: (A3943) SHA2-256: (A3943)
RSA SigVer for DNSSec (legacy)	DigSig-SigVer	Used to verify an RSA digital signature for DNSSec (using a 1024-bit modulus and/or SHA-1)	Publication:FIPS 186-2, FIPS 186-4	RSA SigVer (FIPS186-2): (A3943) RSA SigVer (FIPS186-4): (A3943) HMAC-SHA2-256: (A3943) SHA2-256: (A3943)
PBKDF for PEM Key	PBKDF	Used for derivation of the PEM Key from the PEM Passphrase	Publication:SP 800-132	PBKDF: (A7126)
AES for Encrypting TLS Private Key	BC-UnAuth	Used for encryption of the TLS Private Key using the PEM Key	Publication:SP 800-38A	AES-CBC: (A7126) Key Length: 256
AES for SNMPv3	BC-UnAuth	Used for encryption and decryption of SNMPv3 packets using the SNMPv3 Privacy Key	Publication:SP 800-38A	AES-CFB128: (A7126) Key Length: 128 KDF SNMP: (A7126) Counter DRBG: (A7126)
HMAC for SNMPv3	MAC	Used for authentication of SNMPv3 packets using the SNMPv3 Authentication Key	Publication:FIPS 198-1, SP 800-107 Rev.1 Caveat:The module supports the truncation of HMAC SHA-1 to 96 bits according to NIST SP 800-107 Rev.1	HMAC-SHA-1: (A7126) KDF SNMP: (A7126) SHA-1: (A7126)
RSA SigVer for Software Load Integrity	DigSig-SigVer	Used to verification of a new software image prior to load	Publication:FIPS 186-5	RSA SigVer (FIPS186-5): (A7126) Capabilities: Signature Type: pkcs1v1.5 Properties: Modulo: 2048 Hash Pair: SHA2-512 SHA2-512: (A3943)

Name	Type	Description	Properties	Algorithms
RSA SigVer for Web GUI	DigSig-SigVer	Used for certificate-based authentication via the Web GUI	Publication:FIPS 186-4, FIPS 186-5	RSA SigVer (FIPS186-4): (A3943) Modulo: 1024/2048/3072/4096 Hash Pair: Hash Algorithm: SHA-1 SHA2-256: (A3943) SHA2-384: (A3943) SHA2-512: (A3943) RSA SigVer (FIPS186-5): (A3943)
Key Generation for TLS Ticket Keys (DP)	KAS-135KDF	Used for generation of the TLS Ticket Encryption Key and TLS Ticket Authentication Key	Publication:SP 800-135 Rev. 1	Hash DRBG: (A3943)

Table 11: Security Function Implementations

2.7 Algorithm Specific Information

2.7.1 AES-GCM

The AES-GCM IV[14] is used in the following protocols:

- For SSH, the module meets the (key/IV) pair uniqueness requirements from *NIST SP 800-38D*. The protocol's implementation is contained within the boundary of the module, and the generated IV is only used in the context of the AES-GCM encryption executing the provisions of the SSH protocol. The mechanism for IV generation falls into scenario 1 in *FIPS 140-3 IG* C.H and is compliant with *RFC 4252*, *RFC 4253*, and *RFC 5647*.

  A new IV parameter is generated by the module for each AES-GCM encryption. The IV consists of a 32-bit fixed field and a 64-bit invocation counter. The fixed field of this IV remains the same for the duration of the session. The invocation counter is treated as a 64-bit integer and is incremented by one when performing an AES-GCM encryption.

  If the invocation counter reaches its maximum value 2^64 - 1, the next AES-GCM encryption is performed with the invocation counter set to either 0 or 1. No more than 2^64 - 1 AES-GCM encryptions are performed in the same session. When a session is terminated for any reason, it is the responsibility of the module operator to derive a new key and a new initial IV.

- For TLS v1.2, the module supports acceptable AES-GCM cipher suites from section 3.3.1.1 of *NIST SP 800-52rev2* and meets the (key/IV) pair uniqueness requirements from *NIST SP 800-38D*. The protocol's implementation is contained within the boundary of the module, and the generated IV is only used in the context of the AES-GCM encryption executing the provisions of the TLS 1.2 protocol.

  The mechanism for IV generation falls into scenario 1 in *FIPS 140-3 IG* C.H and is compliant with *RFC 5288*. The IV is a random 96-bit value generated with entropy provided by the module's Approved entropy

[14] IV – Initialization Vector

source. The 64-bit counter portion of the IV is strictly increasing. The counter portion of the IV does not exhaust the maximum number of possible values for a given session key. This condition is implicitly ensured by the design of the TLS protocol, in which the counter is denied exhaustion by the control exerted by the protocol's (and hence also the module's) management logic (wherein the counter is incremented per each TLS record). This management logic also implies that the probability of an exhaustion of all 2^64 - 1 values of the counter for the same TLS session in a realistic time frame is not significant.

- For TLS v1.3, the module supports acceptable AES-GCM cipher suites from section 3.3.1.2 of *NIST SP 800-52rev2* and meets the (key/IV) pair uniqueness requirements from *NIST SP 800-38D*. The protocol's implementation is contained within the boundary of the module, and the generated IV is only used in the context of the AES-GCM encryption executing the provisions of the TLS 1.3 protocol.

  The mechanism for IV generation falls into scenario 5 in *FIPS 140-3 IG* C.H and is compliant with *RFC 8446*. Each session employs a "per-record nonce", a 64-bit sequence number (or IV) maintained separately for reading and writing records. Each sequence number is set to 0 at the beginning of a connection and whenever the key is changed (the first record transmitted under a particular traffic key uses sequence number 0), and the appropriate sequence number is incremented by one after reading or writing each record. Because the size of sequence numbers is 64 bits, the IV should not exhaust the maximum number of possible values for a given session key. If the IV exhaustion condition is observed, this will trigger a session termination or a re-key due to session re-establishment.

The module also supports internal IV generation using the module's Approved DRBG. Per section 8.2.2 of *NIST SP 800-38D*, the IV is at least 96 bits in length. The Approved DRBG generates outputs such that the (key/IV) pair collision probability is less than 2^-32. The mechanism for IV generation falls into scenario 2 in *FIPS 140-3 IG* C.H

If the module's power is lost and then restored, the CO shall establish a new key for AES-GCM encryption.

2.7.2 ECDSA

The module implements ECDSA signature functions that were CAVP-tested against *FIPS PUB 186-4*. These tests are mathematically identical to the *FIPS PUB 186-5* tests. Thus, as allowed per Additional Comment #3 under *FIPS 140-3 IG* C.K, compliance with the *FIPS PUB 186-5* tests is claimed.

2.7.3 PBKDF2

The module uses PBKDF2 option 1a from section 5.4 of *NIST SP 800-132* for key establishment.

The PBKDF implementation takes an input salt that is a minimum of 128 bits in length, with a password/passphrase containing at least 8 characters and an iteration count of 10,000, producing a random value of 256 bits for AES keys. The underlying pseudorandom function used in this derivation is HMAC SHA2-256.

As specified in *NIST SP 800-132*, keys derived from passwords/passphrases are used only in storage applications.

2.7.4 RSA

The module implements RSA signature functions that were CAVP-tested against *FIPS PUB 186-4*. These tests are mathematically identical to the *FIPS PUB 186-5* tests. Thus, as allowed per Additional Comment #3 under *FIPS 140-3 IG* C.K, compliance with the *FIPS PUB 186-5* tests is claimed.

2.8 RNG and Entropy

The module's DRBG is seeded via entropy generated internally from CPU jitter. The module requests a minimum of 256 bits of entropy per call, and it generates symmetric SSPs with up to 256 bits in size and security strength and asymmetric SSPs with up to 6144 bits in size and 178 bits of security strength.

The table below specifies the module's entropy certificates.

Cert Number	Vendor Name
E52	Cloud Software Group

Table 12: Entropy Certificates

The table below specifies the module's entropy sources.

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
NetScaler CPU Jitter Entropy Source	Non-Physical	FreeBSD 11.4	256 bits	A request for 256 bits of entropy results in 256 bits of entropy per output sample, or full entropy.	SHA3-256 (A3513)

Table 13: Entropy Sources

2.9 Key Generation

The module implements the following Approved key generation methods specified in *NIST SP 800-140D Rev. 2*:

- Counter DRBG + CKG
- Hash DRBG + CKG
- RSA key generation
- ECDSA key generation
- DH key generation
- ECDH key generation

In compliance with *NIST SP 800-133*, the module uses its Approved DRBGs to generate cryptographic keys and seeds used for the generation of cryptographic keys. The resulting symmetric key or generated seed is an unmodified output from the DRBG.

2.10 Key Establishment

2.10.1 Key Agreement Schemes

The module implements the following Approved key agreement methods specified in *FIPS 140-3 IG D.F*:

- KAS-ECC-SSC + SSH KDF
- KAS-ECC-SSC + TLS 1.0/1.1 KDF
- KAS-ECC-SSC + TLS 1.2 KDF
- KAS-ECC-SSC + TLS 1.3 KDF
- KAS-ECC-SSC + SSH KDF
- KAS-FFC-SSC + TLS 1.0/1.1 KDF
- KAS-FFC-SSC + TLS 1.2 KDF
- KAS-ECC-SSC + TLS 1.3 KDF

The module performs assurances for its key agreement schemes as specified in the following sections of *NIST SP 800-56A Rev 3:*

- Section 5.5.2 (for assurances of domain parameter validity)
- Section 5.6.2.1 (for assurances required by the key pair owner)
- Section 5.6.2.2.2 (for recipient assurance of ephemeral public key validity)

Key confirmation is not supported by the module.

2.10.2 Key Transport Methods

The module implements the following Approved key agreement methods specified in *FIPS 140-3 IG D.G*:

- AES + MAC key wrap/unwrap
- AES-CCM key wrap/unwrap
- AES-GCM key wrap/unwrap
- RSA key encapsulation/decapsulation

2.11 Industry Protocols

The module supports the following industry protocols which use cryptography in the Approved mode of operation:

- IKEv1
- IKEv2
- SNMP
- SSH
- TLS v1.0/v1.1
- TLS v1.2
- TLS v1.3

The KDFs associated with these protocols shall only be used within the context of their respective protocols. No parts of these protocols, other than the Approved cryptographic algorithms and the KDFs, have been tested by the CAVP and CMVP.

2.12 Additional Information

Algorithms designated as “Legacy” can only be used on data that was generated prior to the Legacy Date specified in *FIPS 140-3 IG C.M*.

3. Cryptographic Module Interfaces

3.1 Ports and Interfaces

The module supports the following logical interfaces:

- Data Input
- Data Output
- Control Input
- Control Output
- Status Output

As a virtual appliance, VPX has no physical characteristics. Its interfaces are logical; the hypervisor provides virtualized ports and interfaces that map to the host server's physical ports and interfaces. The module relies on the physical and electrical characteristics, manual controls, and physical indicators of the host server. The table below contains a mapping of the physical and logical interfaces of the module.

Physical Port	Logical Interface(s)	Data That Passes
N/A	Data Input	Network traffic (ingress)
N/A	Data Output	Network traffic (egress)
N/A	Control Input	Initial configuration data from a connected computer; management data used to remotely manage the appliance
N/A	Control Output	Control information is sent to remote machines supporting LDAP and RADIUS in order for the module to communicate with these machines.
N/A	Status Output	Status information used to remotely monitor the appliance

Table 14: Ports and Interfaces

4. Roles, Services, and Authentication

4.1 Authentication Methods

Module operators are required to authenticate to the module for assumption of an authorized role. The module supports identity-based authentication. Role assumption is implicit, as module operator roles are assigned to their account.

Each session remains active (logged in) and secured until the operator logs out or is automatically logged out from inactivity (inactivity default is 900 seconds). When the module is powered off, results of any previous authentication will be cleared. Here, module operators are required to re-authenticate to assume a new role.

The strength objectives of the authentication mechanisms are as follows:

- For each attempt to use an authentication mechanism, the probability shall be less than one in 1,000,000 that a random attempt will succeed or a false acceptance will occur.

- For multiple attempts to use an authentication mechanism during a one-minute period, the probability shall be less than one in 100,000 that a random attempt will succeed or a false acceptance will occur.

To meet the stated strength objectives, the password policies shall be configured by the Crypto Officer such that all passwords shall require:

- A minimum of 8 characters
- At least 1 lowercase letter
- At least 1 uppercase letter
- At least 1 digit
- At least 1 special character (~, `, !, @, #, $, %, ^, &, *, -, _, =, +, {, }, [,], |, \, :, <, >, /, ., ,, " ")

The Crypto Officer shall configure the password policies during module initialization. Once set, the module enforces the password policies on all subsequent attempts to change a password. See section 11.1.2.1 below for instructions on setting password policies via the Web GUI.

The strength calculations for each of the authentication mechanisms are provided in the table below.

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
Password	The minimum length of the password is eight characters, with 90 different case-sensitive alphanumeric characters and symbols possible for usage. The password must contain: At least 8 characters, At least one lowercase letter, At least one uppercase letter, At least one digit, At least one special character (~, `, !, @, #, $, %, ^, &, *, -, _, =, +, {, }, [,], |, \, :, <, >, /, ., ,, " ")	Username / Password	The minimum length of the password is eight characters, with 90 different case-sensitive alphanumeric characters and symbols possible for usage. The probability of a random attempt falsely succeeding is: =1 per 908 possible passwords, =1 per 4.3x1015	The fastest network connection supported by the module is 1000 Mbps. At most (1x10^9 bits/second × 60 seconds) = 6x10^10 = 60,000,000,000 bits of data can be transmitted in one minute. The minimum password is 64 bits (8 bits per character x 8 characters), meaning 9.375x10^8 passwords can be passed to the module (assuming there is no overhead). This equates to a 1:4,591,650 chance of a random attempt will succeed, or a false acceptance will occur in a one-minute period.
Certificate	The module supports 2048-bit RSA digital certificate authentication during Web GUI/HTTPS (TLS) access. This equates to a 112-bit symmetric key.	RSA SigVer for Web GUI	Using conservative estimates and equating a 2048-bit RSA key to a 112-bit symmetric key, the probability of a random attempt falsely succeeding is: 1 per 2^122, 1 per 5.19 * 10^33	The fastest network connection supported by the module is 1000 Mbps. At most (1x10^9 bits/second × 60 seconds) = 6x10^10 = 60,000,000,000 bits of data can be transmitted in one minute. Thus, at most 60,000,000,000 / 2048 or 2.93x10^7 certificates can be passed to the module in a one-minute period (assuming there is no overhead), Given that there can be 60,000,000,000 bits of data transmitted to the module in one minute and that a certificate contains a 2048-bit RSA key, then at most 60,000,000,000 / 2048 or 2.93x10^7 certificates can be passed to the module in a one-minute period (assuming there is no overhead), meaning if one key has a 1:5.19x10^33 chance of succeeding then in a one minute period there is a 2.93x10^7:5.19x10^33, or 1:1.77x10^26 chance of a random attempt succeeding

Table 15: Authentication Methods

For first-time access, the module comes with a factory-set default login ID (“nsroot”) and password (“nsroot”) for the CLI, RESTful Nitro API, and web GUI. These credentials are used by the CO for initial setup and configuration of the device. During initial configuration, the CO shall disable local authentication to prevent any further use of the default root account (see section 11.1.2.4 below for instructions).

4.2 Roles

The module supports the following role(s):

- Crypto Officer (CO) – The CO role performs administrative services on the module, such as initialization, configuration, and monitoring of the module. The CO role includes the privileges listed under the read-only, operator, network, and sysadmin command policies.

- User – The User role can view the module status and employ module services (including IPsec[15], TLS, SSH, and SNMPv3 services). The User role includes the privileges listed under the read-only command policy.

The following table lists the supported roles.

Name	Type	Operator Type	Authentication Methods
Crypto Officer	Identity	CO	Password Certificate
User	Identity	User	Password Certificate

Table 16: Roles

Operators connect to the module via an SSH connection (using the CLI) or via a TLS connection (using the Web GUI or REST API). The module can support up to 10 operator sessions concurrently from multiple client devices. Each secure session for simultaneous operators is distinguished and kept separate by unique session information, which is provided by the session protocol and protected by the OS.

4.3 Approved Services

Descriptions of the services available are provided in the table below.

The module only supports Approved services. Thus, as allowed per section 2.4.C of *FIPS 140-3 Implementation Guidance*, the module provides indicators for the use of Approved services through a combination of an explicit indication (via a global Approved mode indicator) and an implicit indication (via the successful completion of the service).

The keys and Sensitive Security Parameters (SSPs) listed in the table indicate the type of access required using the following notation:

- G = Generate: The module generates or derives the SSP.
- R = Read: The SSP is read from the module (e.g., the SSP is output).
- W = Write: The SSP is updated, imported, or written to the module.
- E = Execute: The module uses the SSP in performing a cryptographic operation.
- Z = Zeroize: The module zeroizes the SSP.

[15] IPsec – Internet Protocol Security

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Configure system settings	Configure modes and features, system settings, and cloud parameters	Command Line Interface	Command and parameters	Command response/status output	AES for Disk Encryption AES for Data Encryption CKG (KEK)	Crypto Officer - AES Key: G - KEK : G,E - Hash DRBG Entropy: W,E - Hash DRBG Seed: G,E - Hash DRBG 'V' Value : G,E - Hash DRBG 'C' Value: G,E - KEK Fragment 1: G,E - KEK Fragment 2: G,E
Configure network settings	Configure network routing protocols	Command Line Interface	Command and parameters	Command response / status output	AES for Disk Encryption CKG (KEK)	Crypto Officer - ZebOS Router Password (Alphanumeric string): W - KEK : G,E,Z - KEK Fragment 1: G,E - KEK Fragment 2: G,E
Configure clustering	Configure an appliance to either be the cluster coordinator or a node in the cluster	Command Line Interface	Command and parameters	Command response / status output / control output	None	Crypto Officer - Cluster Password (Alphanumeric string): W
Manage data policy encryption keys	Add, edit, delete encryption keys	Command Line Interface	Command	Status output	AES for Disk Encryption AES for Data Encryption CKG (KEK)	Crypto Officer - AES Key: G - KEK : G,E,Z - Hash DRBG Entropy: W,E - Hash DRBG Seed: G,E - Hash DRBG 'V' Value : G,E - Hash DRBG 'C' Value: G,E - KEK Fragment 1: G,E - KEK Fragment 2: G,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Manage data policy HMAC keys	Add, edit, delete HMAC keys	Command Line Interface	Command	Status output	AES for Disk Encryption CKG (KEK) HMAC for Message Authentication	Crypto Officer - HMAC Key: G - KEK : G,E,Z - Hash DRBG Entropy: W,E - Hash DRBG Seed: G,E - Hash DRBG 'V' Value : G,E - Hash DRBG 'C' Value: G,E - KEK Fragment 1: G,E - KEK Fragment 2: G,E
Exchange routing information	Exchange routing update information using ZebOS, authenticate source of packets	Show Command O/P and Traffic	Command	Status output	AES for Disk Encryption	Crypto Officer - ZebOS Router Password (Alphanumeric string): E - KEK : E

Zeroize	Reboot the module (same as power cycle)	N/A	Command	Status output	None	Crypto Officer - PEM Passphrase: Z - PEM Key: Z - AES GCM Key: Z - AES GCM IV : Z - DH Public Key: Z - DH Peer Public Key: Z - DH Private Key: Z - ECDH Public Key : Z - ECDH Peer Public Key : Z - ECDH Private Key : Z - RSA Public Key: Z - RSA Private Key : Z - SSH Shared Secret: Z - SSH Session Key: Z - SSH Authentication Key: Z - IKEv1 PSK: Z - IKEv1 SKEYID: Z - IKEv1 SKEYID_e: Z - IKEv1 SKEYID_a: Z - IKEv1 SKEYID_d: Z - IKEv1 Session Key: Z - IKEv2 SKEYSEED: Z - IKEv2 Session Key: Z - IKEv2 Authentication Key: Z - TLS Pre-Master Secret: Z - TLS Master Secret: Z - TLS Session Key: Z - TLS Authentication Key: Z - TLS Ticket Encryption Key: Z - TLS Ticket Authentication Key: Z - Hash DRBG Entropy: Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- Hash DRBG Seed: Z - Hash DRBG 'V' Value : Z - Hash DRBG 'C' Value: Z - CTR DRBG Entropy: Z - CTR DRBG Seed: Z - CTR DRBG 'V' Value: Z - CTR DRBG 'Key' Value: Z - SNMPv3 Privacy Key: Z - SNMPv3 Authentication Key: Z - IKEv1 Authentication Key: Z - IPsec Session Key: Z - IPsec Authentication Key: Z - IKEv1 Shared Secret: Z - IKEv2 Shared Secret: Z
Zeroize KEK	Zeroize KEK	API return value	Command	Status output	None	Crypto Officer - KEK : Z
Configure Gateway	Configure Gateway global settings, virtual servers, portal themes, AAA groups and users, policies, and resources	Command Line Interface	Command and parameters	Command response / status output	AES for Disk Encryption CKG (KEK)	Crypto Officer - KEK : G,E,Z - KEK Fragment 1: G,E - KEK Fragment 2: G,E
Configure IPsec	Configure IPsec profile; configure CloudBridge Connector settings, network bridges, and IP tunnels; view IP tunnel details	Command Line Interface	Command and parameters	Command response / status output	AES for Disk Encryption CKG (KEK)	Crypto Officer - IKEv1 PSK: W - KEK : G,E,Z - KEK Fragment 1: G,E - KEK Fragment 2: G,E

Establish IKEv1 session/IPsec session	Establish an IKEv1 Session	Traffic	Command	Status output	AES for Disk Encryption CKG (KEK) DH Key Agreement for IKE (CP) AES for IKE/IPsec (CP) HMAC for IKE/IPsec (CP)	Crypto Officer - CTR DRBG Entropy: W,E - CTR DRBG Seed: G,E - CTR DRBG 'V' Value: G,E - CTR DRBG 'Key' Value: G,E - DH Private Key: G,E - DH Public Key: G,R - DH Peer Public Key: W,E - KEK : G,E,Z - IKEv1 Shared Secret: G,E - IKEv1 SKEYID: G,E - IKEv1 SKEYID_e: G,E - IKEv1 SKEYID_a: G,E - IKEv1 SKEYID_d: G,E - IKEv1 Session Key: G,E - IKEv1 Authentication Key: G,E - IPsec Session Key: G,E - IPsec Authentication Key: G,E - KEK Fragment 1: G,E - KEK Fragment 2: G,E User - CTR DRBG Entropy: W,E - CTR DRBG Seed: G,E - CTR DRBG 'V' Value: G,E - CTR DRBG 'Key' Value: G,E - DH Private Key: G,E - DH Public Key: G,R - DH Peer Public Key: W,E - IKEv1 PSK: W,E - IKEv1 Shared Secret: G,E - IKEv1 SKEYID:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						G,E - IKEv1 SKEYID_e: G,E - IKEv1 SKEYID_a: G,E - IKEv1 SKEYID_d: G,E - IKEv1 Session Key: G,E - IKEv1 Authentication Key: G,E - KEK : G,E,Z - IPsec Session Key: G,E - IPsec Authentication Key: G,E - KEK Fragment 1: G,E - KEK Fragment 2: G,E

Establish IKEv2 session/IPsec session	Establish an IKEv2 Session	Traffic	Command	Status output	AES for Disk Encryption CKG (KEK) DH Key Agreement for IKE (CP) AES for IKE/IPsec (CP) HMAC for IKE/IPsec (CP)	Crypto Officer - CTR DRBG Entropy: W,E - CTR DRBG Seed: G,E - CTR DRBG 'V' Value: G,E - CTR DRBG 'Key' Value: G,E - DH Private Key: G,E - DH Public Key: G,R - DH Peer Public Key: W,E - IKEv2 Shared Secret: W,E - IKEv2 SKEYSEED: G,E - IKEv2 Session Key: G,E - IKEv2 Authentication Key: G,E - KEK : G,E,Z - IPsec Session Key: G,E - IPsec Authentication Key: G,E - KEK Fragment 1: G,E - KEK Fragment 2: G,E User - CTR DRBG Entropy: W,E - CTR DRBG Seed: G,E - CTR DRBG 'V' Value: G,E - CTR DRBG 'Key' Value: G,E - DH Private Key: G,E - DH Public Key: G,R - DH Peer Public Key: W,E - IKEv2 Shared Secret: G,E - IKEv2 SKEYSEED: G,E - IKEv2 Session Key: G,E - IKEv2 Authentication Key: G,E - KEK : G,E,Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- IPsec Session Key: G,E - IPsec Authentication Key: G,E - KEK Fragment 1: G,E - KEK Fragment 2: G,E
Configure SSH	Configure SSH authentication settings; generate SSH keys	Command Line Interface	Command and parameters	Command response / status output	Key Generation for SSH Authentication (CP)	Crypto Officer - CTR DRBG Entropy: W,E - CTR DRBG Seed: G,E - CTR DRBG 'V' Value: G,E - CTR DRBG 'Key' Value: G,E - SSH Private Key: G - SSH Public Key: G

Establish SSH sessions	Establish an SSH session (includes authentication of the communicating parties, negotiation of cryptographic modes and parameters, establishment of shared secrets, and derivation of session keys)	Traffic	Command	Status output	DH Key Agreement for SSH (CP) ECDH Key Agreement for SSH (CP) AES for SSH (CP) HMAC for SSH (CP)	Crypto Officer - SSH Private Key: E - SSH Public Key: R - SSH Peer Public Key: W,E - CTR DRBG Entropy: R,E - CTR DRBG Seed: G,E - CTR DRBG 'V' Value: G,E - CTR DRBG 'Key' Value: G,E - DH Private Key: G,E - DH Public Key: G,R - DH Peer Public Key: W,E - ECDH Private Key : G,E - ECDH Public Key : G,R - ECDH Peer Public Key : W,E - SSH Shared Secret: G,E - SSH Session Key: G,E - SSH Authentication Key: G,W,E User - SSH Private Key: E - SSH Public Key: R - SSH Peer Public Key: W,E - CTR DRBG Entropy: R,E - CTR DRBG Seed: G,E - CTR DRBG 'V' Value: G,E - CTR DRBG 'Key' Value: G,E - DH Private Key: G,E - DH Public Key: G,R - DH Peer Public Key: W,E - ECDH Private Key : G,E - ECDH Public Key : G,R - ECDH Peer Public Key : W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- SSH Shared Secret: G,E - SSH Session Key: G,E - SSH Authentication Key: G,W,E
Zeroize SSH private keys	Zeroize SSH private keys	API return value	Command	Status output	None	Crypto Officer - SSH Private Key: Z
Configure SNMPv3	Configure SNMP communities, traps, managers, views, groups, users, alarms, and engine ID ; view SNMP OIDs	Command Line Interface	Command and parameters	Command response / status output	AES for Disk Encryption CKG (KEK)	Crypto Officer - SNMPv3 Privacy Passphrase : W - SNMPv3 Authentication Passphrase: W - KEK : G,E - KEK Fragment 1: G,E - KEK Fragment 2: G,E
SNMPv3 traps	Provides system condition information	Log files	None	Status output / control output	AES for SNMPv3 HMAC for SNMPv3	Crypto Officer - SNMPv3 Authentication Passphrase: E - SNMPv3 Privacy Passphrase : E - SNMPv3 Privacy Key: G,E - SNMPv3 Authentication Key: G,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Configure TLS	Configure TLS sessions and profiles	Command Line Interface	Command and parameters	Command response / status output	AES for Disk Encryption CKG (KEK) AES for TLS Tickets HMAC for TLS Tickets SigGen for TLS Authentication (CP) SigGen for TLS Authentication (DP) RSA SigGen for DNSSec RSA SigVer for DNSSec RSA SigVer for DNSSec (legacy) PBKDF for PEM Key AES for Encrypting TLS Private Key Key Generation for TLS Ticket Keys (DP)	Crypto Officer - CA Public Key: W - TLS Private Key: G,E - TLS Public Key: G,W - TLS Master Secret: G,E - TLS Ticket Encryption Key: W - TLS Ticket Authentication Key: W - PEM Passphrase: G,E - PEM Key: G - KEK : G,E,Z - CTR DRBG Entropy: W,E - CTR DRBG Seed: G,E - CTR DRBG 'V' Value: G,E - CTR DRBG 'Key' Value: G,E - Hash DRBG Entropy: G,E - Hash DRBG Seed: G,E - Hash DRBG 'V' Value : G,E - Hash DRBG 'C' Value: G,E - DNS Private KSK: W - DNS Public KSK: W - DNS Private ZSK: W - DNS Public ZSK: W - KEK Fragment 1: G,E - KEK Fragment 2: G,E

Establish TLS session (CP)	Establish a TLS session on the control plane (includes server authentication, negotiation of cryptographic modes and parameters, establishment of shared secrets, and generation of session keys)	Traffic	Command	Status output	CKG (KEK) SigVer for TLS Authentication (CP) SigVer for TLS Authentication (CP) (legacy) DH Key Agreement for TLS (CP) ECDH Key Agreement for TLS (CP) RSA Key Transport for TLS (CP) AES for TLS (CP) HMAC for TLS (CP)	Crypto Officer - TLS Private Key: E - TLS Public Key: R - TLS Peer Public Key: W,E - DH Private Key: W,E - DH Public Key: G,R - DH Peer Public Key: W,E - ECDH Private Key : G,E - ECDH Public Key : G,W - ECDH Peer Public Key : W,E - RSA Private Key : G,E - RSA Public Key: G,E - TLS Pre-Master Secret: G,E - TLS Master Secret: G,E - TLS Session Key: G,E - TLS Authentication Key: G,E - AES GCM IV : G,E - KEK : G,E,Z - CTR DRBG Entropy: W,E - CTR DRBG Seed: G,E - CTR DRBG 'V' Value: G,E - CTR DRBG 'Key' Value: G,E - PEM Key: G,E - PEM Passphrase: W,E - CA Public Key: E - KEK Fragment 1: G,E - KEK Fragment 2: G,E User - TLS Private Key: E - TLS Public Key: R - TLS Peer Public Key: W,E - DH Private Key: W,E - DH Public Key: G,R

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- DH Peer Public Key: W,E - ECDH Private Key : R,E - ECDH Public Key : G,R - ECDH Peer Public Key : W,E - RSA Private Key : G,E - RSA Public Key: G,E - TLS Pre-Master Secret: G,E - TLS Master Secret: G,E - TLS Session Key: G,E - TLS Authentication Key: G,E - AES GCM IV : G,E - PEM Passphrase: W,E - PEM Key: G,E - KEK : G,E,Z - CTR DRBG Entropy: W,E - CTR DRBG Seed: G,E - CTR DRBG 'V' Value: G,E - CTR DRBG 'Key' Value: G,E - CA Public Key: E - KEK Fragment 1: G,E - KEK Fragment 2: G,E

Establish TLS session (DP)	Establish a TLS session on the data plane (includes server authentication, negotiation of cryptographic modes and parameters, establishment of shared secrets, and generation of session keys)	Traffic	Command	Status output	CKG (KEK) AES for TLS Tickets HMAC for TLS Tickets SigVer for TLS Authentication (DP) SigVer for TLS Authentication (DP) (legacy) ECDH Key Agreement for TLS (DP) RSA Key Transport for TLS (DP) AES for TLS (DP) HMAC for TLS (DP) Key Generation for TLS Ticket Keys (DP)	Crypto Officer - TLS Private Key: E - TLS Public Key: R - TLS Peer Public Key: W,E - ECDH Private Key : G,E - ECDH Public Key : G,R - RSA Private Key : G,E - RSA Public Key: G,E - TLS Pre-Master Secret: G,E - TLS Master Secret: G,E - TLS Session Key: G,E - TLS Authentication Key: G,E - AES GCM IV : G,E - PEM Passphrase: W,E - PEM Key: G,E - KEK : G,E,Z - Hash DRBG Entropy: G,E - Hash DRBG Seed: G,E - Hash DRBG 'V' Value : G,E - Hash DRBG 'C' Value: G,E - CA Public Key: E - ECDH Peer Public Key : W,E - KEK Fragment 1: G,E - KEK Fragment 2: G,E User - TLS Private Key: E - TLS Public Key: R - TLS Peer Public Key: W,E - ECDH Private Key : G,E - ECDH Public Key : G,R - ECDH Peer Public Key : W,E - RSA Private Key : G,E - RSA Public Key: G,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- TLS Pre-Master Secret: G,E - TLS Master Secret: G,E - TLS Session Key: G,E - TLS Authentication Key: G,E - AES GCM IV : G,E - PEM Passphrase: W,E - PEM Key: G,E - KEK : G,E,Z - Hash DRBG Entropy: G,E - Hash DRBG Seed: G,E - Hash DRBG 'V' Value : G,E - Hash DRBG 'C' Value: G,E - CA Public Key: E - KEK Fragment 1: G,E - KEK Fragment 2: G,E
Resume TLS session (DP)	Resume a TLS session on the data plane (includes decryption and authentication of the TLS ticket and use of TLS Session Key and TLS Authentication Key)	Traffic	Command	Status output	AES for TLS Tickets HMAC for TLS Tickets AES for TLS (DP) HMAC for TLS (DP)	Crypto Officer - TLS Session Key: W,E - TLS Authentication Key: W,E - AES GCM IV : W,E - TLS Ticket Encryption Key: W,E - TLS Ticket Authentication Key: W,E User - TLS Session Key: W,E - TLS Authentication Key: W,E - AES GCM IV : W,E - TLS Ticket Encryption Key: W,E - TLS Ticket Authentication Key: W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Show status	Show the system status	N/A	Command	Status output	None	Crypto Officer User
Perform self-tests on demand	Perform pre-operational self-tests	Log file	Command	Status output	None	Crypto Officer
Show versioning information	Show module name and version	Console Output	Command	Module name, version	None	Crypto Officer User
Load software image	Update the module's software to a new version	Log files	Command	Status output	RSA SigVer for Software Load Integrity	Crypto Officer - Software Load Integrity Key : W
Authenticate via RADIUS	Used for operator logins to the module using RADIUS	Traffic	Command	Status output	None	Unauthenticated - Operator Password: W
Authenticate via LDAP	Used for operator logins to the module using LDAP	Traffic	Command	Status output	CKG (KEK) RSA SigVer for Web GUI	Unauthenticated - Operator Password: W - LDAP Admin Password: R,E - SSH Public Key: E - TLS Public Key: E - AES Key: E - AES GCM Key: E - AES GCM IV : E - KEK : G,E,Z - KEK Fragment 1: G,E - KEK Fragment 2: G,E

Table 17: Approved Services

The module offers additional non-security-relevant services for module configuration and administration. These services do not employ security services, nor do they access SSPs. For more information regarding these services, refer to the Getting started with NetScaler article found on the online product documentation portal.

4.4 Non-Approved Services

The module does not provide any non-Approved services.

4.5 External Software/Firmware Loaded

The module provides the capability of upgrading its software by downloading a complete software image from an external source. To update the module software, the following steps must be performed:

- Download a new software load integrity key (for use with the next upgrade).
- Download the software upgrade package.
- Run the installation script.

When the installation script is finished, it will prompt the CO to restart the module. Upon restart, the module will load the new image into memory for execution and perform its pre-operational integrity test using a 2048-bit RSA

digital signature to verify the image's software components before executing. If the test is passed, the module will begin execution of the new image. If the test is failed, the bootup process will abort, and the module will be rendered inoperable. To recover, CSG customer support will need to be contacted for assistance.

Prior to execution of the new image, module operators shall zeroize all keys in non-volatile memory using the methods described in section 9.3 below. All keys in volatile memory are zeroized on module reboot.

In order to maintain the module's validation, only validated images may be loaded. The new image shall include updated version information to represent the newly loaded image.

5. Software/Firmware Security

5.1 Integrity Techniques

At module start-up, all software components within the cryptographic boundary are verified using an Approved integrity technique implemented within the cryptographic module itself. The module verifies a single encompassing 2048-bit RSA digital signature with a SHA2-512 hash to ensure the integrity of all of its software components. Unsuccessful verification of any component will cause the module to enter a critical error state.

5.2 Initiate on Demand

The pre-operational integrity test can be launched on demand by the following methods:

- power-cycling the module's host device
- issuing the `reboot` CLI command
- issuing the `reboot` API method
- via the Web GUI by navigating to **Configuration > System > System Information** and clicking the **Reboot** button

6. Operational Environment

6.1 Operational Environment Type and Requirements

The NetScaler VPX is a software cryptographic module that executes in a Modifiable operational environment.

The module runs on FreeBSD OS 11.4, which acts as the guest OS on top of the virtualization layer provided by VMware's ESXi 8 hypervisor. The VMware hypervisor runs directly on the host server's hardware, with no need for an underlying operating system. Only the module's signed image can be executed, and all software upgrades are digitally signed.

All services provided by the module are provided by the module's software and external interfaces. The module's processor executes the software on the tested configurations specified in section 2.2.4 of this document.

7. Physical Security

The cryptographic module is a multi-chip standalone software module and does not include physical security mechanisms. Therefore, per section G.3 of the *Implementation Guidance for FIPS PUB 140-3 and the CMVP*, this section is not applicable.

8. Non-Invasive Security

This section is not applicable. There are currently no approved non-invasive mitigation techniques references in Annex F of *ISO/IEC 19790*.

9. Sensitive Security Parameters Management

9.1 Storage Areas

The table below lists sensitive security parameters (SSPs) storage areas for this module. Section 9.4 below selects from the storage areas listed here and specifies the associated storage area in the “Storage” column for each SSP.

Storage Area Name	Description	Persistence Type
On Disk	SSPs are stored on Host Device's disk	Static
In Volatile Memory	SSPs are stored in volatile memory	Dynamic

Table 18: Storage Areas

9.2 SSP Input-Output Methods

The table below lists SSP input and output methods for this module. Section 9.4 below selects from the methods listed here and specifies the associated method in the “Inputs/Outputs” column for each SSP.

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
Imported plaintext form via public key certificate	External	On Disk	Plaintext	Automated	Electronic	
Imported encrypted form via SSH	External	On Disk	Encrypted	Automated	Electronic	AES for SSH (CP)
Imported encrypted form via TLS (CP)	External	On Disk	Encrypted	Automated	Electronic	AES for TLS (CP)
Imported encrypted form via TLS (DP)	External	On Disk	Encrypted	Automated	Electronic	AES for TLS (DP)
Imported plaintext via local console	External	On Disk	Plaintext	Automated	Electronic	
Imported plaintext	External	On Disk	Plaintext	Automated	Electronic	
Imported encrypted form via RSA key transport (CP)	External	On Disk	Encrypted	Automated	Electronic	RSA Key Transport for TLS (CP)
Imported encrypted form via RSA key transport (DP)	External	On Disk	Encrypted	Automated	Electronic	RSA Key Transport for TLS (DP)
Exported encrypted form via part of config backup file	On Disk	External	Encrypted	Automated	Electronic	AES for Disk Encryption
Exported plaintext form via public key certificate	On Disk	External	Plaintext	Automated	Electronic	
Exported encrypted form via RSA key transport (CP)	On Disk	External	Encrypted	Automated	Electronic	RSA Key Transport for TLS (CP)
Exported encrypted form via RSA key transport (DP)	On Disk	External	Encrypted	Automated	Electronic	RSA Key Transport for TLS (DP)

Table 19: SSP Input-Output Methods

9.3 SSP Zeroization Methods

The table below lists SSP zeroization methods for this module. Section 9.4 below selects from the methods listed here and specify the associated method in the “Zeroization” column for each SSP.

Zeroization Method	Description	Rationale	Operator Initiation
Completion of TLS key derivation	Temporary SSPs used for deriving TLS keys are zeroized.	Keys are overwritten with zeroes. Zeroization is immediate and non-interruptible. Zeroization occurs in a sufficiently small time period, preventing the recovery of the sensitive data between the time zeroization is initiated and the actual zeroization is completed.	N/A
Reboot/Remove power	All ephemeral keys are cleared from memory on module reboot or power removal.	Keys are overwritten with zeroes. Zeroization is immediate and non-interruptible. Zeroization occurs in a sufficiently small time period, preventing the recovery of the sensitive data between the time zeroization is initiated and the actual zeroization is completed.	The module operator reboots or removes power from the module.
Session termination	Ephemeral session keys are automatically zeroized when a secure protocol session is terminated.	Keys are overwritten with zeroes. Zeroization is immediate and non-interruptible. Zeroization occurs in a sufficiently small time period, preventing the recovery of the sensitive data between the time zeroization is initiated and the actual zeroization is completed.	N/A
Completion of KEK operation	The KEK resides in volatile memory and is automatically zeroized and freed after each use.	Keys are overwritten with zeroes. Zeroization is immediate and non-interruptible. Zeroization occurs in a sufficiently small time period, preventing the recovery of the sensitive data between the time zeroization is initiated and the actual zeroization is completed.	N/A
CLI command (KEK fragments)	KEK fragments are zeroized and KEK fragment files are deleted, rendering all passphrases and passwords stored in the non-volatile memory unrecoverable.	Keys are overwritten with zeroes. Zeroization is immediate and non-interruptible. Zeroization occurs in a sufficiently small time period, preventing the recovery of the sensitive data between the time zeroization is initiated and the actual zeroization is completed.	Crypto Officer issues the following CLI command: rm system csps -type KEK
CLI command (SSH private keys)	SSH private keys stored in non-volatile memory in plaintext form are zeroized.	Keys are overwritten with zeroes. Zeroization is immediate and non-interruptible. Zeroization occurs in a sufficiently small time period, preventing the recovery of the sensitive data between the time zeroization is initiated and the actual zeroization is completed.	Crypto Officer issues the following CLI command: rm system csps -type SSH_HOST_KEYS

Table 20: SSP Zeroization Methods

For CLI command-based methods, the success status is indicated by the display of the command line prompt after completion of the command without any error showing on the console. If the command fails, an error will show on the console before returning control to the module operator.

When the pre-operational integrity test is complete, the module zeroizes all temporary values used in the integrity test.

9.4 SSPs

The module supports the keys and other SSPs listed in the table below. As a virtual appliance running on a host server, all SSP imports and exports performed by the module are electronic and occur within the TOEPP.

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
KEK Fragment 1	Key fragment hashed in combination with KEK Fragment 2 to generate KEK	N/A - N/A	Keying material - PSP	Counter DRBG (A7126)		CKG (KEK)
KEK Fragment 2	Key fragment hashed in combination with KEK Fragment 1 to generate KEK	N/A - N/A	Keying material - PSP	Counter DRBG (A7126)		CKG (KEK)
KEK	Symmetric key used for encryption and decryption of passwords and passphrases	256 bits - 256 bits	Symmetric Key - CSP	CKG (KEK)		AES for Disk Encryption
PEM Key	Symmetric key used for encryption and decryption of asymmetric private keys	256 bits - 256 bits	Symmetric Key - CSP		PBKDF for PEM Key	AES for Encrypting TLS Private Key
AES Key	Symmetric key used for encryption and decryption	Between 128 and 256 bits - Between 128 and 256 bits	Symmetric Key - CSP	Hash DRBG (A3943)		AES for Data Encryption
AES GCM Key	Symmetric key used for encryption and decryption	256 bits - 128 or 256 bits	Symmetric Key - CSP	Hash DRBG (A3943)		AES for Disk Encryption
HMAC Key	Key used for message authentication	Between 160 and 512 bits - Between 128 and 256 bits	Authentication - CSP	Hash DRBG (A3943)		HMAC for Message Authentication
CA Public Key	Public key used for TLS certificate authentication	[for ECDSA] Between 224 and 512 bits [for RSA] Between 2048 and 3072 bits - [for ECDSA] Between 224 and 512 bits [for RSA] Between 2048 and 3072 bits	Public/Private - PSP			SigVer for TLS Authentication (CP) SigVer for TLS Authentication (CP) (legacy) SigVer for TLS Authentication (DP) SigVer for TLS Authentication (DP) (legacy)

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
DH Private Key	Private key used for generating shared secrets in IKE, SSH, and TLS	SSH: Between 2048 and 6144 bits, TLS: Between 2048 and 4096 bits, IKE: 2048 bits - SSH: Between 112 and 176 bits, TLS: Between 112 and 150 bits, IKE: 112 bits	Public/Private - CSP	Safe Primes Key Generation (A7126)		DH Key Agreement for SSH (CP) DH Key Agreement for IKE (CP) DH Key Agreement for TLS (CP)
DH Public Key	Public key used by peer for generating shared secrets in IKE, SSH, and TLS	SSH: Between 2048 and 6144 bits, TLS: Between 2048 and 4096 bits, IKE: 2048 bits - SSH: Between 112 and 176 bits, TLS: Between 112 and 150 bits, IKE: 112 bits	Public/Private - PSP	Safe Primes Key Generation (A7126)		
DH Peer Public Key	Peer public key used for generating of shared secrets in IKE, SSH, and TLS	SSH: Between 2048 and 6144 bits, TLS: Between 2048 and 4096 bits, IKE: 2048 bits - SSH: Between 112 and 176 bits, TLS: Between 112 and 150 bits, IKE: 112 bits	Public/Private - PSP			DH Key Agreement for SSH (CP) DH Key Agreement for IKE (CP) DH Key Agreement for TLS (CP)
ECDH Private Key	Private key used for generating shared secrets in SSH and TLS	Between 224 and 512 bits - Between 112 and 256 bits	Public/Private - CSP	ECDSA KeyGen (FIPS186-4) (A3943) ECDSA KeyGen (FIPS186-5) (A3943) ECDSA KeyGen (FIPS186-5) (A7126)		ECDH Key Agreement for SSH (CP) ECDH Key Agreement for TLS (CP) ECDH Key Agreement for TLS (DP)

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
ECDH Public Key	Public key used by peer for generating shared secrets in SSH and TLS	Between 224 and 512 bits - Between 112 and 256 bits	Public/Private - PSP	ECDSA KeyGen (FIPS186-4) (A3943) ECDSA KeyGen (FIPS186-5) (A3943) ECDSA KeyGen (FIPS186-5) (A7126)		
RSA Private Key	Private key used in TLS key transport functions	2048 or 3072 bits - 112 or 128 bits	Public/Private - CSP	RSA KeyGen (FIPS186-5) (A7126)		RSA Key Transport for TLS (CP) RSA Key Transport for TLS (DP)
RSA Public Key	Public key used by peer in TLS key transport functions	2048 or 3072 bits - 112 or 128 bits	Public/Private - PSP	RSA KeyGen (FIPS186-5) (A7126)		RSA Key Transport for TLS (CP) RSA Key Transport for TLS (DP)
SSH Private Key	Private key used for authentication during SSH session negotiation; RBA Authentication for LDAP; GSLB configuration sync	RSA: 2048 or 3072, ECDSA: Between 224 and 512 bits - RSA: 112 or 128 bits, ECDSA: Between 112 and 256 bits	Public/Private - CSP	ECDSA KeyGen (FIPS186-5) (A7126) RSA KeyGen (FIPS186-5) (A7126)		DH Key Agreement for SSH (CP) ECDH Key Agreement for SSH (CP)
SSH Public Key	Public key used by peer for authentication during SSH session negotiation; RBA Authentication for LDAP; GSLB configuration sync	RSA: 2048 or 3072, ECDSA: Between 224 and 512 bits - RSA: 112 or 128 bits, ECDSA: Between 112 and 256 bits	Public/Private - PSP	ECDSA KeyGen (FIPS186-5) (A7126) RSA KeyGen (FIPS186-5) (A7126)		DH Key Agreement for SSH (CP) ECDH Key Agreement for SSH (CP)
SSH Session Key	Symmetric key used for encryption and decryption of SSH session packets	Between 128 and 256 bits - Between 128 and 256 bits	Symmetric Key - CSP		KDF SSH (A7126)	AES for SSH (CP)
SSH Authentication Key	Key used for authentication of SSH session packets	Between 160 and 512 bits - Between 128 and 256 bits	Authentication - PSP		KDF SSH (A7126)	HMAC for SSH (CP)
IKEv1 Shared Secret	Secret value used for deriving the IKEv1 SKEYID	-	Shared Secret - CSP		KAS-FFC-SSC Sp800-56Ar3 (A7126)	DH Key Agreement for IKE (CP)
IKEv1 PSK	Pre-shared key used for authentication of secure key exchange during IKEv1 Phase 1	-	Authentication - CSP			
IKEv1 SKEYID	Secret value used for deriving other IKEv1 secrets	-	Secret - CSP		KDF IKEv1 (A7126)	DH Key Agreement for IKE (CP)

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
IKEv1 SKEYID_e	Secret value for deriving IKE Session Key	-	Secret - CSP		KDF IKEv1 (A7126)	DH Key Agreement for IKE (CP)
IKEv1 SKEYID_a	Secret value for deriving IKE Authentication Key	-	Secret - CSP		KDF IKEv1 (A7126)	DH Key Agreement for IKE (CP)
IKEv1 SKEYID_d	Secret value used for deriving IPsec keys	-	Secret - CSP		KDF IKEv1 (A7126)	DH Key Agreement for IKE (CP)
IKEv1 Session Key	Symmetric key used for encryption and decryption of IKE messages	Between 128 and 256 bits - Between 128 and 256 bits	Symmetric Key - CSP		KDF IKEv1 (A7126)	AES for IKE/IPsec (CP)
TLS Session Key	Symmetric key used for encryption and decryption of TLS session packets	AES: 128 or 256 bits AES GCM: 128 or 256 bits - AES: 128 or 256 bits AES GCM: 128 or 256 bits	Symmetric Key - CSP		KDF TLS (A3943) TLS v1.2 KDF RFC7627 (A3943) TLS v1.2 KDF RFC7627 (A7126) TLS v1.3 KDF (A3943)	AES for TLS (CP) AES for TLS (DP)
TLS Authentication Key	Key used for authentication of TLS session packets	Between 160 and 384 bits - Between 128 and 256 bits	Authentication - CSP		KDF TLS (A3943) TLS v1.2 KDF RFC7627 (A3943) TLS v1.2 KDF RFC7627 (A7126) TLS v1.3 KDF (A3943)	HMAC for TLS (CP) HMAC for TLS (DP)
TLS Ticket Encryption Key	Symmetric key used for encryption and decryption of TLS session tickets	128 bits - 128 bits	Symmetric Key - CSP	Hash DRBG (A3943)		AES for TLS Tickets
TLS Ticket Authentication Key	Key used for authentication of TLS session tickets	- 256 bits	Authentication - CSP	Hash DRBG (A3943)		HMAC for TLS Tickets
SNMPv3 Privacy Key	Symmetric key used for encryption and decryption of SNMPv3 packets	128 bits - 128 bits	Symmetric Key - CSP		KDF SNMP (A7126)	AES for SNMPv3
SNMPv3 Authentication Key	Key used for authentication of SNMPv3 packets	160 bits - 128 bits	Authentication - CSP		KDF SNMP (A7126)	HMAC for SNMPv3
DNS Private KSK	Private key used to sign the DNS Public ZSK	Between 2048 and 4096 bits - Between 112 and 150 bits	Public/Private - CSP	RSA KeyGen (FIPS186-5) (A7126)		RSA SigGen for DNSSec
DNS Public KSK	Public key used to verify the DNS Public ZSK	Between 1024 and 4096 bits - Between 80 and 150 bits	Public/Private - PSP	RSA KeyGen (FIPS186-5) (A7126)		RSA SigVer for DNSSec RSA SigVer for DNSSec (legacy)

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
DNS Private ZSK	Private key used to sign records in a DNS zone	Between 2048 and 4096 bits - Between 112 and 150 bits	Public/Private - CSP	RSA KeyGen (FIPS186-5) (A7126)		RSA SigGen for DNSSec
Software Load Integrity Key	Public key used to verify the new software image prior to load	2048 bits - 112 bits	Public/Private - Neither			RSA SigVer for Software Load Integrity
PEM Passphrase	Passphrase used to derive PEM Key	-	Alphanumeric String - CSP			AES for Encrypting TLS Private Key
AES GCM IV	IV for AES GCM	96 and 128-bits -	Initialization Vector - CSP	Counter DRBG (A7126) KDF SSH (A7126) TLS v1.2 KDF RFC7627 (A3943) TLS v1.2 KDF RFC7627 (A7126) TLS v1.3 KDF (A3943)		AES for Disk Encryption AES for SSH (CP) AES for TLS (CP) AES for TLS (DP)
SSH Shared Secret	Shared secret used to derive the SSH Session Key and SSH Authentication Key	-	Shared Secret - CSP		KAS-ECC-SSC Sp800-56Ar3 (A7126) KAS-FFC-SSC Sp800-56Ar3 (A7126)	DH Key Agreement for SSH (CP) ECDH Key Agreement for SSH (CP)
TLS Pre-Master Secret	Shared secret used to derive the TLS Master Secret	-	Pre-Master Secret - CSP		KAS-ECC-SSC Sp800-56Ar3 (A3943) KAS-ECC-SSC Sp800-56Ar3 (A7126) KAS-FFC-SSC Sp800-56Ar3 (A7126)	DH Key Agreement for TLS (CP) ECDH Key Agreement for TLS (CP) ECDH Key Agreement for TLS (DP)
TLS Master Secret	Derivation of the TLS Session Key and TLS Authentication Key	-	Master Secret - CSP		KDF TLS (A3943) TLS v1.2 KDF RFC7627 (A3943) TLS v1.2 KDF RFC7627 (A7126) TLS v1.3 KDF (A3943)	DH Key Agreement for TLS (CP) ECDH Key Agreement for TLS (CP) ECDH Key Agreement for TLS (DP)
Hash DRBG Entropy	Entropy input for Hash DRBG	-	Entropy - CSP	SHA3-256 (A3513)		Hash DRBG (A3943)
Hash DRBG Seed	Seed material for Hash DRBG	-	DRBG Seed - CSP	Hash DRBG (A3943)		Hash DRBG (A3943)
Hash DRBG 'V' Value	Internal state value used with Hash DRBG	-	Internal State Value - CSP	Hash DRBG (A3943)		Hash DRBG (A3943)

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
Hash DRBG 'C' Value	Internal state value used with Hash DRBG	-	Internal State Value - CSP	Hash DRBG (A3943)		Hash DRBG (A3943)
CTR DRBG Entropy	Entropy input for CTR DRBG	-	Entropy - CSP	SHA3-256 (A3513)		Counter DRBG (A7126)
CTR DRBG Seed	Seed material for CTR DRBG	-	DRBG Seed - CSP	Counter DRBG (A7126)		Counter DRBG (A7126)
CTR DRBG 'V' Value	Internal state value used with CTR DRBG	-	Internal State Value - CSP	Counter DRBG (A7126)		Counter DRBG (A7126)
CTR DRBG 'Key' Value	Internal state value used with CTR DRBG	-	Internal State Value - CSP	Counter DRBG (A7126)		Counter DRBG (A7126)
SNMPv3 Privacy Passphrase	Derivation of the SNMPv3 Privacy Key	-	Alphanumeric String - CSP			AES for SNMPv3
SNMPv3 Authentication Passphrase	Derivation of the SNMPv3 Authentication Key	-	Alphanumeric String - CSP			HMAC for SNMPv3
LDAP Admin Password	Used to bind to the LDAP server	-	Alphanumeric String - CSP			
ZebOS Router Password (Alphanumeric string)	Router authentication	-	Alphanumeric String - CSP			
Cluster Password (Alphanumeric string)	Authentication to the cluster coordinator node in a cluster configuration	-	Alphanumeric String - CSP			
Operator Password	Authentication to the module via an external authentication service	-	Alphanumeric String - CSP			
ECDH Peer Public Key	Peer public key used for generating of shared secrets in SSH and TLS	Between 224 and 512 bits - Between 112 and 256 bits	Public/Private - PSP			ECDH Key Agreement for SSH (CP) ECDH Key Agreement for TLS (CP) ECDH Key Agreement for TLS (DP)
SSH Peer Public Key	Peer public key used for authentication during SSH session negotiation; RBA Authentication for LDAP; GSLB configuration sync	RSA: 2048 or 3072, ECDSA: Between 224 and 512 bits - RSA: 112 or 128 bits, ECDSA: Between 112 and 256 bits	Public/Private - PSP			DH Key Agreement for SSH (CP) ECDH Key Agreement for SSH (CP)
IKEv1 Authentication Key	Key used for authentication of IKE messages	Between 160 and 512 bits - Between 128 and 256 bits	Authentication - CSP		KDF IKEv1 (A7126)	HMAC for IKE/IPsec (CP)
IKEv2 Shared Secret	Secret value used for deriving the IKEv1 SKEYID	-	Shared Secret - CSP		KAS-FFC-SSC Sp800-56Ar3 (A7126)	DH Key Agreement for IKE (CP)

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
IKEv2 SKEYSEED		-	Shared Secret - CSP		KDF IKEv2 (A7126)	DH Key Agreement for IKE (CP)
IKEv2 Session Key	Symmetric key used for encryption and decryption of IKEv2 messages	Between 128 and 256 bits - Between 128 and 256 bits	Symmetric Key - CSP		KDF IKEv2 (A7126)	AES for IKE/IPsec (CP)
IKEv2 Authentication Key	Key used for authentication of IKEv2 messages	Between 160 and 512 bits - Between 128 and 256 bits	Authentication - CSP		KDF IKEv2 (A7126)	HMAC for IKE/IPsec (CP)
IPsec Session Key	Symmetric key used for encryption and decryption of IPsec message traffic	-	Symmetric Key - CSP		KDF IKEv1 (A7126) KDF IKEv2 (A7126)	AES for IKE/IPsec (CP)
IPsec Authentication Key	Key used for authentication of IPsec message traffic	-	Authentication - CSP		KDF IKEv1 (A7126) KDF IKEv2 (A7126)	HMAC for IKE/IPsec (CP)
TLS Private Key	Private key used for TLS session authentication Private key used for SAML authentication (RSA only) Private key used for OpenID authentication (RSA only)	RSA: 2048 or 4096, ECDSA: Between 224 and 512 bits - RSA: 112 or 150, ECDSA: Between 112 and 256 bits	Public/Private - CSP	ECDSA KeyGen (FIPS186-4) (A3943) ECDSA KeyGen (FIPS186-5) (A3943) ECDSA KeyGen (FIPS186-5) (A7126) RSA KeyGen (FIPS186-5) (A7126)		SigVer for TLS Authentication (CP) SigVer for TLS Authentication (DP)
TLS Public Key	Public key used by peer for TLS session authentication	RSA: 2048 or 4096, ECDSA: Between 224 and 512 bits - RSA: 112 or 150, ECDSA: Between 112 and 256 bits	Public/Private - PSP	ECDSA KeyGen (FIPS186-4) (A3943) ECDSA KeyGen (FIPS186-5) (A3943) ECDSA KeyGen (FIPS186-5) (A7126) RSA KeyGen (FIPS186-5) (A7126)		

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
TLS Peer Public Key	Peer public key used for TLS session authentication	RSA: 2048 or 4096, ECDSA: Between 224 and 512 bits - RSA: 112 or 150, ECDSA: Between 112 and 256 bits	Public/Private - PSP			SigVer for TLS Authentication (CP) SigVer for TLS Authentication (CP) (legacy) SigVer for TLS Authentication (DP) SigVer for TLS Authentication (DP) (legacy)
DNS Public ZSK	Public key used to verify records in a DNS zone	Between 1024 and 4096 bits - Between 80 and 150 bits	Public/Private - PSP	RSA KeyGen (FIPS186-5) (A7126)		RSA SigVer for DNSSec RSA SigVer for DNSSec (legacy)

Table 21: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
KEK Fragment 1		On Disk:Plaintext	Until zeroization	CLI command (KEK fragments)	KEK Fragment 2:Used With KEK :Generates
KEK Fragment 2		On Disk:Plaintext	Until zeroization	CLI command (KEK fragments)	KEK Fragment 1:Used With KEK :Generates
KEK		In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power Completion of KEK operation	KEK Fragment 1:Generated from KEK Fragment 2:Generated from
PEM Key		On Disk:Encrypted	Until zeroization method is initiated	CLI command (KEK fragments)	KEK :Encrypted by PEM Passphrase (Alphanumeric string):Derived From
AES Key	Imported encrypted form via SSH Imported encrypted form via TLS (CP) Imported encrypted form via TLS (DP) Imported plaintext via local console Exported encrypted form via part of config backup file	On Disk:Encrypted	Until zeroization method is initiated	CLI command (KEK fragments)	KEK :Encrypted by
AES GCM Key		In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
HMAC Key		On Disk:Encrypted	Until zeroization method is initiated	CLI command (KEK fragments)	KEK :Encrypts
CA Public Key	Imported plaintext via local console Exported plaintext form via public key certificate	On Disk:Plaintext			
DH Private Key		In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power Session termination	DH Public Key:Paired With DH Peer Public Key:Used With SSH Shared Secret:Computes TLS Pre-Master Secret:Computes IKEv1 SKEYID:Computes
DH Public Key	Exported plaintext form via public key certificate	In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power Session termination	DH Private Key:Paired With
DH Peer Public Key	Imported plaintext form via public key certificate	In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power Session termination	DH Private Key:Used With SSH Shared Secret:Computes TLS Pre-Master Secret:Computes IKEv1 SKEYID:Computes
ECDH Private Key		In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power Session termination	ECDH Public Key :Paired With SSH Shared Secret:Computes TLS Pre-Master Secret:Computes ECDH Peer Public Key :Used With
ECDH Public Key	Exported plaintext form via public key certificate	In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power Session termination	ECDH Private Key :Paired With

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
RSA Private Key	Imported encrypted form via RSA key transport (CP) Imported encrypted form via RSA key transport (DP) Exported encrypted form via RSA key transport (CP) Exported encrypted form via RSA key transport (DP)	On Disk:Encrypted	Until zeroization method is initiated	CLI command (KEK fragments)	RSA Public Key:Paired With KEK :Encrypted By
RSA Public Key	Exported plaintext form via public key certificate	In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power Session termination	RSA Private Key :Paired With
SSH Private Key	Exported encrypted form via part of config backup file	On Disk:Plaintext	Until zeroization method is initiated	CLI command (SSH private keys)	SSH Public Key:Paired With
SSH Public Key	Exported plaintext form via public key certificate	In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power Session termination	SSH Private Key:Paired With
SSH Session Key		In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power Session termination	SSH Shared Secret:Derived From
SSH Authentication Key		In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power Session termination	SSH Shared Secret:Derived From
IKEv1 Shared Secret		In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power Session termination	DH Private Key:Derived From DH Peer Public Key:Derived From IKEv1 SKEYID:Derives
IKEv1 PSK	Exported encrypted form via part of config backup file Imported plaintext via local console	In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power Session termination	
IKEv1 SKEYID		In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power Session termination	IKEv1 Shared Secret:Derived From IKEv1 SKEYID_e:Derives IKEv1 SKEYID_a:Derives IKEv1 SKEYID_d:Derives

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
IKEv1 SKEYID_e		In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power Session termination	IKEv1 SKEYID:Derived From IKEv1 Session Key:Derives
IKEv1 SKEYID_a		In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power Session termination	IKEv1 SKEYID:Derived From IKEv1 Authentication Key:Derives
IKEv1 SKEYID_d		In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power Session termination	IKEv1 SKEYID:Derived From
IKEv1 Session Key		In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power Session termination	IKEv1 SKEYID_e:Derived From
TLS Session Key		In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power Session termination	TLS Master Secret:Derived From
TLS Authentication Key		In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power Session termination	TLS Master Secret:Derived From
TLS Ticket Encryption Key		In Volatile Memory:Plaintext On Disk:Encrypted	Until zeroization method is initiated	Reboot/Remove power CLI command (KEK fragments)	KEK :Encrypted by
TLS Ticket Authentication Key		In Volatile Memory:Plaintext On Disk:Encrypted	Until zeroization method is initiated	Reboot/Remove power CLI command (KEK fragments)	KEK :Encrypted by
SNMPv3 Privacy Key		In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power Session termination	SNMPv3 Privacy Passphrase (Alphanumeric string):Derived From
SNMPv3 Authentication Key		In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power Session termination	SNMPv3 Privacy Passphrase (Alphanumeric string):Derived From
DNS Private KSK	Imported encrypted form via SSH Imported encrypted form via TLS (CP) Imported encrypted form via TLS (DP) Exported encrypted form via part of config backup file	On Disk:Encrypted	Until zeroization method is initiated	CLI command (KEK fragments)	PEM Key:Encrypted by DNS Public KSK:Paired With

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
DNS Public KSK	Imported encrypted form via SSH Imported encrypted form via TLS (CP) Imported encrypted form via TLS (DP) Exported encrypted form via part of config backup file	On Disk:Plaintext			Private DNS KSK (RSA private key):Paired With
DNS Private ZSK	Imported encrypted form via SSH Imported encrypted form via TLS (CP) Imported encrypted form via TLS (DP) Exported encrypted form via part of config backup file	On Disk:Encrypted	Until zeroization method is initiated	CLI command (KEK fragments)	PEM Key:Encrypted by DNS Public ZSK:Paired With
Software Load Integrity Key	Imported plaintext	In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power Session termination	
PEM Passphrase	Exported encrypted form via part of config backup file Imported plaintext via local console	In Volatile Memory:Plaintext On Disk:Encrypted	Until zeroization method is initiated	Reboot/Remove power CLI command (KEK fragments)	KEK :Encrypted by
AES GCM IV		In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power	TLS Session Key:Paired With SSH Session Key:Paired With AES GCM Key:Paired With
SSH Shared Secret		In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power Session termination	DH Peer Public Key:Derived From DH Private Key:Derived From ECDH Peer Public Key :Derived by ECDH Private Key :Derived by SSH Session Key:Derives SSH Authentication Key:Derives

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
TLS Pre-Master Secret		In Volatile Memory:Plaintext	Until zeroization method is initiated	Completion of TLS key derivation Reboot/Remove power	DH Peer Public Key:Derived From DH Private Key:Derived From ECDH Peer Public Key :Derived by ECDH Private Key :Derived by TLS Master Secret:Derives
TLS Master Secret		In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power Session termination	TLS Pre-Master Secret:Derived From TLS Session Key:Derives TLS Authentication Key:Derives
Hash DRBG Entropy		In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power	Hash DRBG Seed:Derives
Hash DRBG Seed		In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power	Hash DRBG Seed:Derived From Hash DRBG 'V' Value (Internal state value):Derives Hash DRBG 'C' Value (Internal state value):Derives
Hash DRBG 'V' Value		In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power	Hash DRBG Seed:Derived From Hash DRBG 'C' Value (Internal state value):Paired With
Hash DRBG 'C' Value		In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power	Hash DRBG Seed:Derived From Hash DRBG 'V' Value :Paired With
CTR DRBG Entropy		In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power	CTR DRBG Seed:Derives
CTR DRBG Seed		In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power	CTR DRBG Seed:Derived From CTR DRBG 'V' Value:Derives CTR DRBG 'Key' Value (AES key):Derives
CTR DRBG 'V' Value		In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power	CTR DRBG Seed:Derived From CTR DRBG 'Key' Value (AES key):Paired With
CTR DRBG 'Key' Value		In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power	CTR DRBG Seed:Derived From CTR DRBG 'V' Value:Paired With

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
SNMPv3 Privacy Passphrase	Imported encrypted form via SSH Imported encrypted form via TLS (CP) Imported encrypted form via TLS (DP) Imported plaintext via local console Exported encrypted form via part of config backup file	On Disk:Encrypted	Until zeroization method is initiated	CLI command (KEK fragments)	KEK :Encrypted by
SNMPv3 Authentication Passphrase	Imported encrypted form via SSH Imported encrypted form via TLS (CP) Imported encrypted form via TLS (DP) Imported plaintext via local console Exported encrypted form via part of config backup file	On Disk:Encrypted	Until zeroization method is initiated	CLI command (KEK fragments)	KEK :Encrypted by
LDAP Admin Password	Imported encrypted form via SSH Imported encrypted form via TLS (CP) Imported encrypted form via TLS (DP) Imported plaintext via local console Exported encrypted form via part of config backup file	On Disk:Encrypted	Until zeroization method is initiated	CLI command (KEK fragments)	KEK :Encrypted by

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
ZebOS Router Password (Alphanumeric string)	Imported encrypted form via SSH Imported encrypted form via TLS (CP) Imported encrypted form via TLS (DP) Imported plaintext via local console Exported encrypted form via part of config backup file	On Disk:Encrypted	Until zeroization method is initiated	CLI command (KEK fragments)	KEK :Encrypted by
Cluster Password (Alphanumeric string)	Imported encrypted form via SSH Imported encrypted form via TLS (CP) Imported encrypted form via TLS (DP) Imported plaintext via local console Exported encrypted form via part of config backup file	On Disk:Encrypted	Until zeroization method is initiated	CLI command (KEK fragments)	KEK :Encrypted by
Operator Password	Imported encrypted form via SSH Imported encrypted form via TLS (CP) Imported encrypted form via TLS (DP) Exported encrypted form via part of config backup file	In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power	
ECDH Peer Public Key	Imported plaintext form via public key certificate	In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power Session termination	ECDH Private Key :Used With SSH Shared Secret:Computes TLS Pre-Master Secret:Computes
SSH Peer Public Key	Imported plaintext form via public key certificate	In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power Session termination	SSH Private Key:Used With

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
IKEv1 Authentication Key		In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power Session termination	IKEv1 SKEYID_a:Derived From
IKEv2 Shared Secret		In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power Session termination	DH Private Key:Derived From DH Peer Public Key:Derived From IKEv2 SKEYSEED:Derives
IKEv2 SKEYSEED		In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power Session termination	IKEv2 Shared Secret:Derived From
IKEv2 Session Key		In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power Session termination	IKEv2 SKEYSEED:Derived From
IKEv2 Authentication Key		In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power Session termination	IKEv2 SKEYSEED:Derived From
IPsec Session Key		In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power Session termination	IKEv1 SKEYID:Derived From IKEv1 Shared Secret:Derived From IKEv2 Shared Secret:Derived From
IPsec Authentication Key		In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power Session termination	IKEv1 SKEYID:Derived From IKEv1 Shared Secret:Derived From IKEv2 Shared Secret:Derived From
TLS Private Key	Imported encrypted form via SSH Imported encrypted form via TLS (CP) Imported encrypted form via TLS (DP) Imported plaintext via local console Exported encrypted form via part of config backup file	On Disk:Encrypted	Until zeroization method is initiated	CLI command (KEK fragments)	PEM Key:Encrypts TLS Public Key:Paired With TLS Peer Public Key:Used With
TLS Public Key	Exported plaintext form via public key certificate	In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power Session termination	TLS Private Key:Paired With

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
TLS Peer Public Key	Imported plaintext form via public key certificate	In Volatile Memory:Plaintext	Until zeroization method is initiated	Reboot/Remove power Session termination	TLS Private Key:Used With
DNS Public ZSK	Imported encrypted form via SSH Imported encrypted form via TLS (CP) Imported encrypted form via TLS (DP) Exported encrypted form via part of config backup file	On Disk:Plaintext			DNS Private ZSK:Paired With

Table 22: SSP Table 2

9.5 Transitions

The following list specifies applicable transition periods or timeframes where an algorithm or key length transitions from Approved to non-Approved:

- The module includes implementations of 1024-bit RSA signature verification that comply with *FIPS PUB 186-2* and *FIPS PUB 186-4*. These publications have been superseded by *FIPS PUB 186-5*, and these implementations are now allowed for legacy use only.

- The module includes implementations of SHA-1 for MAC generation and digital signature verification. SHA-1 will be non-Approved for all uses starting January 1, 2031.

- In compliance with *NIST SP 800-131A Rev. 2*, the module supports algorithms and key lengths that provide a minimum of 112 bits of security strength for applying cryptographic protection. Starting January 1, 2031, the minimum security strength for applying cryptographic protection will be 128 bits, and security strengths between 112 bits and 128 bits will be allowed for legacy use only to process information that is already protected.

10. Self-Tests

The module performs pre-operational self-tests and conditional self-tests. Pre-operational tests are performed between the time the cryptographic module is instantiated and before the module transitions to the operational state. Conditional self-tests are performed by the module during module operation when certain conditions exist. The following sections list the self-tests performed by the module, their expected error status, and the error resolutions.

10.1 Pre-Operational Self-Tests

The table below lists the pre-operational self-tests performed by the module.

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
RSA SigVer (FIPS186-4) (A3943)	2048-bit, using SHA2-512	Software Integrity	SW/FW Integrity	"FIPS Post Failed" message in /var/log/ns.log	RSA 2048 digital signature verification with SHA-512

Table 23: Pre-Operational Self-Tests

10.2 Conditional Self-Tests

The table below lists the conditional self-tests performed by the module.

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-CBC (A7126)	128-bit	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	Encrypt	During module start-up after successful completion of the integrity test
AES-GCM (A7126)	128-bit	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	Encrypt	During module start-up after successful completion of the integrity test
Counter DRBG (A7126)		KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	Instantiate/Generate/Reseed	During module start-up after successful completion of the integrity test
KAS-FFC-SSC Sp800-56Ar3 (A7126)		KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	Primitive "Z" computation test	During module start-up after successful completion of the integrity test
KAS-ECC-SSC Sp800-56Ar3 (A7126)		KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	Primitive "Z" computation test	During module start-up after successful completion of the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
ECDSA SigGen (FIPS186-5) (A7126)	P-256	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	Sign	During module start-up after successful completion of the integrity test
ECDSA SigVer (FIPS186-5) (A7126)	P-256	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	Verify	During module start-up after successful completion of the integrity test
HMAC-SHA-1 (A7126)	SHA-1	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	Hashed message authentication	During module start-up after successful completion of the integrity test
HMAC-SHA2-256 (A7126)	SHA2-256	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	Hashed message authentication	During module start-up after successful completion of the integrity test
HMAC-SHA2-512 (A7126)	SHA2-512	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	Hashed message authentication	During module start-up after successful completion of the integrity test
PBKDF (A7126)	SHA-1	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log		During module start-up after successful completion of the integrity test
RSA SigGen (FIPS186-5) (A7126)	2048-bit, SHA2-256	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	Sign	During module start-up after successful completion of the integrity test
RSA SigVer (FIPS186-5) (A7126)	2048-bit, SHA2-256	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	Verify	Before integrity test
SHA-1 (A7126)		KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log		During module start-up after successful completion of the integrity test
SHA2-256 (A7126)		KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log		During module start-up after successful completion of the integrity test
SHA2-512 (A7126)		KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log		During module start-up after successful completion of the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
KDF IKEv1 (A7126)		KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	IKEv1 KDF Test	During module start-up after successful completion of the integrity test
KDF IKEv2 (A7126)		KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	IKEv2 KDF Test	During module start-up after successful completion of the integrity test
KDF SSH (A7126)		KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	SSH KDF Test	During module start-up after successful completion of the integrity test
KDF TLS (A7126)		KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	TLS Test	During module start-up after successful completion of the integrity test
TLS v1.2 KDF RFC7627 (A7126)		KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	TLS v1.2 Test	During module start-up after successful completion of the integrity test
ECDSA digital signature PCT (CP)		PCT	PCT	Message is displayed and logged if errored	Sign/verify	Upon generation of a key pair for ECDSA signature functions
DH key agreement PCT (CP)		PCT	PCT	Message is displayed and logged if errored	Key agreement	Upon generation of a key pair for ECDH key agreement functions
ECDH key agreement PCT (CP)		PCT	PCT	Message is displayed and logged if errored	Key agreement	Upon generation of a key pair for ECDH key agreement functions
RSA digital signature PCT (CP)		PCT	PCT	Message is displayed and logged if errored	Sign/verify	Upon generation of a key pair for RSA signature functions
RSA key transport PCT (CP)		PCT	PCT	Message is displayed and logged if errored	Encrypt/decrypt	Upon generation of a key pair for RSA key transport functions
Software Load Test	2048-bit RSA SigVer; SHA2-512	SW / FW Load	SW/FW Load	Message is displayed and logged if errored	RSA signature verification 2048-bit software load test	Upon performing a software update
AES-CBC (A3943)	128-bit	KAT	CAST	"FIPS POST Failed" message in /var/log/ns.log	Encrypt	During module start-up after successful completion of the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-GCM (A3943)	128-bit	KAT	CAST	"FIPS POST Failed" message in /var/log/ns.log	Encrypt	During module start-up after successful completion of the integrity test
KAS-ECC-SSC Sp800-56Ar3 (A3943)		KAT	CAST	"FIPS POST Failed" message in /var/log/ns.log	Primitive "Z" computation test	During module start-up after successful completion of the integrity test
ECDSA SigGen (FIPS186-5) (A3943)	P-256	KAT	CAST	"FIPS POST Failed" message in /var/log/ns.log	Sign	During module start-up after successful completion of the integrity test
ECDSA SigVer (FIPS186-5) (A3943)	P-256	KAT	CAST	"FIPS POST Failed" message in /var/log/ns.log	Verify	During module start-up after successful completion of the integrity test
Hash DRBG (A3943)	AES, 256-bit, with derivation function	KAT	CAST	"FIPS POST Failed" message in /var/log/ns.log	Instantiate/Generate/Reseed	During module start-up after successful completion of the integrity test
HMAC-SHA-1 (A3943)	128 bits	KAT	CAST	"FIPS POST Failed" message in /var/log/ns.log	Hashed message authentication	During module start-up after successful completion of the integrity test
HMAC-SHA2-256 (A3943)	256 bits	KAT	CAST	"FIPS POST Failed" message in /var/log/ns.log	Hashed message authentication	During module start-up after successful completion of the integrity test
HMAC-SHA2-512 (A3943)	256 bits	KAT	CAST	"FIPS POST Failed" message in /var/log/ns.log	Hashed message authentication	During module start-up after successful completion of the integrity test
RSA SigGen (FIPS186-5) (A3943)		KAT	CAST	"FIPS POST Failed" message in /var/log/ns.log	Sign	During module start-up after successful completion of the integrity test
RSA SigVer (FIPS186-5) (A3943)		KAT	CAST	"FIPS POST Failed" message in /var/log/ns.log	Verify	During module start-up after successful completion of the integrity test
SHA-1 (A3943)		KAT	CAST	"FIPS POST Failed" message in /var/log/ns.log		During module start-up after successful completion of the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
SHA2-256 (A3943)		KAT	CAST	"FIPS POST Failed" message in /var/log/ns.log		During module start-up after successful completion of the integrity test
SHA2-512 (A3943)		KAT	CAST	"FIPS POST Failed" message in /var/log/ns.log		During module start-up after successful completion of the integrity test
KDF TLS (A3943)		KAT	CAST	"FIPS POST Failed" message in /var/log/ns.log	TLS Test	During module start-up after successful completion of the integrity test
TLS v1.2 KDF RFC7627 (A3943)		KAT	CAST	"FIPS POST Failed" message in /var/log/ns.log	TLS v1.2 Test	During module start-up after successful completion of the integrity test
TLS v1.3 KDF (A3943)		KAT	CAST	"FIPS POST Failed" message in /var/log/ns.log	TLS v1.3 Test	During module start-up after successful completion of the integrity test
ECDSA digital signature PCT (DP)		PCT	PCT	Message is displayed and logged if errored	Sign/verify	Upon generation of a key pair for ECDSA signature functions
ECDH key agreement PCT (DP)		PCT	PCT	Message is displayed and logged if errored	Key agreement	Upon generation of a key pair for ECDH key agreement functions
RSA digital signature PCT (DP)		PCT	PCT	Message is displayed and logged if errored	Sign/verify	Upon generation of a key pair for RSA signature functions
RSA key transport PCT (DP)		PCT	PCT	Message is displayed and logged if errored	Encrypt/decrypt	Upon generation of a key pair for RSA key transport functions
SHA3-256 (NetScaler CPU Jitter Entropy Source)		KAT	CAST	"FIPS POST Failed" message in /var/log/ns.log		During module start-up after successful completion of the integrity test
Entropy Adaptive Proportion Test		APT	CAST	Message is displayed and logged if errored	Adaptive Proportion Test on entropy source	During module start-up after successful completion of the integrity test

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
Entropy Repetition Count Test		RCT	CAST	Message is displayed and logged if errored	Repetition Count Test on entropy source	During module start-up after successful completion of the integrity test
AES-CBC Decrypt (A7126)	128-bit	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	Decrypt	During module start-up after successful completion of the integrity test
AES-GCM Decrypt (A7126)	128-bit	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	Decrypt	During module start-up after successful completion of the integrity test
AES-CBC Decrypt (A3943)	128-bit	KAT	CAST	"FIPS POST Failed" message in /var/log/ns.log	Decrypt	During module start-up after successful completion of the integrity test
AES-GCM Decrypt (A3943)	128-bit	KAT	CAST	"FIPS POST Failed" message in /var/log/ns.log	Decrypt	During module start-up after successful completion of the integrity test
KDF SNMP (A7126)		KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	SSH KDF Test	During module start-up after successful completion of the integrity test

Table 24: Conditional Self-Tests

To ensure all conditional CASTs are performed prior to the first operational use of the associated algorithm, all CASTs are performed during the module's initial start-up sequence. CASTs for algorithms used in the pre-operational integrity test are performed prior to the integrity test itself; all other CASTs are executed immediately after the successful completion of the integrity test.

10.3 Periodic Self-Test Information

The module permits operators to initiate the pre-operational integrity test and conditional CASTs on demand for periodic testing of the module. The following tables list the applicable self-tests and the associated methods for test initiation.

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
RSA SigVer (FIPS186-4) (A3943)	Software Integrity	SW/FW Integrity	On Demand	Manually

Table 25: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-CBC (A7126)	KAT	CAST	On Demand	Manually
AES-GCM (A7126)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
Counter DRBG (A7126)	KAT	CAST	On Demand	Manually
KAS-FFC-SSC Sp800-56Ar3 (A7126)	KAT	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3 (A7126)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-5) (A7126)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-5) (A7126)	KAT	CAST	On Demand	Manually
HMAC-SHA-1 (A7126)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A7126)	KAT	CAST	On Demand	Manually
HMAC-SHA2-512 (A7126)	KAT	CAST	On Demand	Manually
PBKDF (A7126)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-5) (A7126)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-5) (A7126)	KAT	CAST	On Demand	Manually
SHA-1 (A7126)	KAT	CAST	On Demand	Manually
SHA2-256 (A7126)	KAT	CAST	On Demand	Manually
SHA2-512 (A7126)	KAT	CAST	On Demand	Manually
KDF IKEv1 (A7126)	KAT	CAST	On Demand	Manually
KDF IKEv2 (A7126)	KAT	CAST	On Demand	Manually
KDF SSH (A7126)	KAT	CAST	On Demand	Manually
KDF TLS (A7126)	KAT	CAST	On Demand	Manually
TLS v1.2 KDF RFC7627 (A7126)	KAT	CAST	On Demand	Manually
ECDSA digital signature PCT (CP)	PCT	PCT		
DH key agreement PCT (CP)	PCT	PCT		
ECDH key agreement PCT (CP)	PCT	PCT		
RSA digital signature PCT (CP)	PCT	PCT		
RSA key transport PCT (CP)	PCT	PCT		
Software Load Test	SW / FW Load	SW/FW Load		
AES-CBC (A3943)	KAT	CAST	On Demand	Manually
AES-GCM (A3943)	KAT	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3 (A3943)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-5) (A3943)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-5) (A3943)	KAT	CAST	On Demand	Manually
Hash DRBG (A3943)	KAT	CAST	On Demand	Manually
HMAC-SHA-1 (A3943)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A3943)	KAT	CAST	On Demand	Manually
HMAC-SHA2-512 (A3943)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-5) (A3943)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
RSA SigVer (FIPS186-5) (A3943)	KAT	CAST	On Demand	Manually
SHA-1 (A3943)	KAT	CAST	On Demand	Manually
SHA2-256 (A3943)	KAT	CAST	On Demand	Manually
SHA2-512 (A3943)	KAT	CAST	On Demand	Manually
KDF TLS (A3943)	KAT	CAST	On Demand	Manually
TLS v1.2 KDF RFC7627 (A3943)	KAT	CAST	On Demand	Manually
TLS v1.3 KDF (A3943)	KAT	CAST	On Demand	Manually
ECDSA digital signature PCT (DP)	PCT	PCT		
ECDH key agreement PCT (DP)	PCT	PCT		
RSA digital signature PCT (DP)	PCT	PCT		
RSA key transport PCT (DP)	PCT	PCT		
SHA3-256 (NetScaler CPU Jitter Entropy Source)	KAT	CAST	On Demand	Manually
Entropy Adaptive Proportion Test	APT	CAST	On Demand	Manually
Entropy Repetition Count Test	RCT	CAST	On Demand	Manually
AES-CBC Decrypt (A7126)	KAT	CAST	On Demand	Manually
AES-GCM Decrypt (A7126)	KAT	CAST	On Demand	Manually
AES-CBC Decrypt (A3943)	KAT	CAST	On Demand	Manually
AES-GCM Decrypt (A3943)	KAT	CAST	On Demand	Manually
KDF SNMP (A7126)	KAT	CAST	On Demand	Manually

Table 26: Conditional Periodic Information

10.4 Error States

The table below describes the module's error states and error status indicators.

Name	Description	Conditions	Recovery Method	Indicator
Critical Error (from pre-operational self tests)	Terminal state where the boot sequence and entire system is halted.	Module fails pre-operational integrity test.	Manual restart must be initiated (clearing the error state) and the module must pass the pre-operational integrity test.	NetScaler Control Plane Cryptographic Library: Failure is indicated by a system halt and "POST Failed" in /var/log/FIPS-post.log. Successful completion of the self-tests is indicated by "POST Success" in /var/log/FIPS-post.log. NetScaler Data Plane Cryptographic Library: Failure is indicated by a system halt and "FIPS Post Failed" in /var/log/ns.log. Successful completion of the self-tests is indicated by "FIPS POST Successful" in /var/log/ns.log.

Name	Description	Conditions	Recovery Method	Indicator
Critical Error (from Conditional CASTs)	Terminal state where cryptographic operations are halted and the module inhibits all data output from the module.	Module fails a conditional CAST.	The module logs the error and automatically reboots to an unconfigured state (clearing the error state).	NetScaler Control Plane Cryptographic Library: Failure is indicated by a system halt and "POST Failed" in /var/log/FIPS-post.log. Successful completion of the self-tests is indicated by "POST Success" in /var/log/FIPS-post.log. NetScaler Data Plane Cryptographic Library: Failure is indicated by a system halt and "FIPS Post Failed" in /var/log/ns.log. Successful completion of the self-tests is indicated by "FIPS POST Successful" in /var/log/ns.log.
Soft Error	Temporary state resulting from failure of a conditional PCT or other non-CAST conditional self-test.	Module fails a conditional PCT or other non-CAST conditional self-test.	The module logs the error and displays an error message (clearing the error state). The module is immediately recovered.	Failure is indicated by the display of the following message: "Internal failure in SSL cert/key generation tool". Successful completion of the conditional self-test is indicated by the absence of an error message.

Table 27: Error States

If the module experiences repeated errors or continuously goes to a halted state, the module is considered to be malfunctioning or compromised, and CSG Customer Support must be contacted.

10.5 Operator Initiation of Self-Tests

Module operators can initiate the pre-operational integrity test and conditional CASTs on demand by rebooting/power-cycling the module.

11. Life-Cycle Assurance

The sections below describe how to ensure the module is operating in its validated configuration, including the following:

- Procedures for secure installation, initialization, startup, and operation of the module
- Maintenance requirements
- Administrator and non-Administrator guidance

Operating the module without following the guidance herein (including the use of undocumented services) will result in non-compliant behavior and is outside the scope of this Security Policy.

11.1 Installation, Initialization, and Startup Procedures

The module is available as a software package that includes both the application software and the operating system. After purchasing NetScaler VPX, the installation files can be downloaded from NetScaler Downloads using valid credentials provided by Cloud Software Group. License entitlement(s) are sent by Cloud Software Group via email after purchase or can be accessed via the NetScaler Support portal using valid credentials.

The CO is responsible for all initial setup activities, including configuring the virtual machine and installing/configuring the NetScaler VPX. Prior to the installation, the CO should become familiar with the document entries within the NetScaler 14.1 section on CSG's NetScaler online product documentation portal.

The following sections provide references to step-by-step instructions for the installation of the module, as well as the steps necessary to configure the module for operation in the Approved mode.

11.1.1 Installation

For detailed guidance regarding the installation of the module, please refer to the following entries on the About NetScaler VPX webpage:

- NetScaler VPX support matrix
- Install a NetScaler VPX instance on VMware ESX

The above document entries include the NetScaler VPX support matrix and usage guidelines, prerequisites for installing the NetScaler VPX, hardware requirements for the host platforms, and NetScaler VPX installation instructions. To install the required license files, the CO shall follow the instructions on the Licensing overview webpage on the online product documentation portal.

11.1.2 Initialization

After installation is complete, the CO is responsible for initializing the module. The CO can use the Web GUI or CLI to perform the configuration steps to prepare the module for operation. General configuration steps must be complete before performing configuration steps necessary to place the module in the Approved mode of

operation. Guidance for secure deployments is provided in the Best practices for NetScaler MPX, VPX, and SDX security page on the online product documentation portal.

Once general configuration steps have been completed, the CO shall perform these additional security-relevant initialization steps to ensure that the module operates in its Approved mode:

- Configure the password/passphrase policies
- Replace the default TLS certificate
- Disable HTTP access to the Web GUI
- Enable external authentication
- Disable local authentication
- Disable non-compliant TLS 1.2 KDF

To accomplish these tasks, the CO shall follow the procedures detailed in the sections below.

11.1.2.1 Configure the Password/Passphrase Requirements

Passphrases are used to derive keys using PBKDF. The CO shall configure strong password/passphrase policies. This is accomplished with the following steps from the Web GUI:

1. In the Configuration navigation pane, go to **System** and click the **Settings** node.
2. In the **Settings** section, click the **Change Global System Settings** link.
3. In the **Strong Password** field, select **Enable All**.
4. In the **Min Password Length** field, type “8”.
5. Click **OK**.

11.1.2.2 Replace the Default TLS Certificate

By default, the module includes a factory-provisioned RSA certificate for TLS connections (`ns-server.cert` and `ns-server.key`). This certificate is not intended for use in production deployments and must be replaced. The CO shall replace the default certificate with a newly generated certificate after the initial installation.

To replace the default TLS certificate, the CO shall follow these steps:

1. Run the following CLI command to set the hostname of the module: `set ns hostName [hostname]`
2. From the Web GUI, complete the following procedure to create a Certificate Signing Request (CSR):
    a. In the Configuration navigation pane, go to **Traffic Management** and click the **SSL** node.
    b. In the **SSL Certificates** section, click the **Create Certificate Request** link.
    c. Make sure to provide values for all the required fields marked with an “*” and then click **Create**. Note that the **Common Name** field will contain the value of `hostname` created in step 1 above.
3. Submit the CSR file to a trusted CA. The CSR file is available in the `/nsconfig/ssl` directory.
4. After receiving the certificate from the trusted CA, copy the file to the `/nsconfig/ssl` directory.
5. From the Web GUI, navigate to **Traffic Management** > **SSL** and choose **ns-server-certificate**.
6. Click **Update**.
7. In the **Certificate File Name** field, choose the certificate file that was received from the CA. Use the **Browse** option to choose the file that you have received from CA after signing**.** Choose the **Browse > Local** option if the file is saved on your workstation/local drive.
8. In the **Private Key File Name** field, specify the default private key file name (`ns-server.key`).
9. Select the **No Domain Check** option.

10. Click **OK**.

For more information, please refer to the Citrix Support Knowledge Center article CTX122521) on the online product documentation portal.

11.1.2.3 Disable HTTP Access to the Web GUI

Traffic to the administrative interface and Web GUI is protected by configuring the module to use HTTPS[16]. Once the module has been configured to use new TLS and SSH certificates, the CO shall disable HTTP access to the GUI management interface with the following CLI command:

```
set ns ip <NSIP> -gui SECUREONLY
```

11.1.2.4 Enable External Authentication

Once the module is configured in Approved mode and the `nsroot` account is disabled, then external authentication must be configured. The CO shall follow the instructions on the External user authentication webpage found on the online product documentation portal to configure external system authentication.

The CO shall ensure the following before enabling external authentication:

- A secure connection is established with the external authentication server.
- Shell access is disabled for all profiles on the external authentication server.

11.1.2.5 Disable Local Authentication

The `nsroot` account is a default account with root CLI access (superuser) privileges that is required for initial configuration. During initial configuration, the CO shall disable local system authentication to block access to all local accounts (including the `nsroot` account), and the CO shall ensure that superuser privileges are not assigned to any user account. To disable local system authentication and enable external system authentication, the CO shall run the following CLI command:

```
set system parameter -localauth disabled
```

11.1.2.6 Disable Non-Compliant TLS 1.2 KDF

The module includes implementations of the TLS 1.2 KDFs that are compliant with RFC 7627 (with the extended master secret) and with RFC 5246 (with the non-EMS master secret). To disable support for the non-EMS KDF, the CO shall do the following:

- For the control plane, add the following line to the `/etc/ssl/openssl.conf` file (under `fips_sect`):

```
disable-no-ems-kdf =1
```

- For the data plane, configure the following CLI under SSL profile:

```
set ssl profile <profile name> -allowLegacyKDF NO
set ssl profile <profile name> -allowextendedMasterSecret YES
```

[16] HTTPS – Hypertext Transfer Protocol Secure

11.1.3 Startup

No additional startup steps are required to be performed by end-users.

11.2 Administrator Guidance

Once installed and configured, the Crypto Officer is responsible for maintaining and monitoring the status of the module to ensure that it is running in its Approved mode. Please refer to this section for guidance that the Crypto Officer must follow to ensure that the module is operating in the Approved mode.

11.2.1 Status Information

The CO shall be responsible for regularly monitoring the module's status for the Approved mode of operation. When configured according to the CO's guidance, the module only operates in the Approved mode. Thus, the current status of the module when operational is always in the Approved mode.

11.2.2 Versioning Information

An operator can view the versioning information using any of the following methods:

- Using the CLI:
  - `show ns info` - shows details about the software, including software version, enabled and disabled features, and configured network information
  - `show ns version` – shows version and build number of the appliance
  - `show ns hardware` – shows details of the appliance hardware and information such as the host ID[17] and serial number
- Using the RESTful Nitro API with the GET method:

```
https://module-ip-address>/nitro/v5/config/nshardware
https://module-ip-address>/nitro/v5/config/nsversion
```

- Using the Web GUI by navigating to **Configuration > System > System Information**

These methods will display general system and hardware information about the device, including the platform version, CPU information, and appliance serial number. Additionally, the Web GUI's dashboard includes a system overview section with information such as system HA state, system master state, and system uptime.

The versioning output "NetScaler NS14.1, version NS14.1-FIPS" denotes the module name **NetScaler VPX** and module version **14.1.FIPS**, which can be correlated with the module's validation record.

[17] ID – Identifier

11.2.3 Additional Administrator Policies and Guidance

This section notes additional policies below that must be followed by the CO:

- All private keys (except for SSH private keys) must be stored as PEM files in encrypted format using one of the Approved encryption algorithms listed in section 2.5.1.
- Upon successful bootup of the module, the module is configured by default to use only *NIST SP 800-52rev2* recommended cipher suites for TLS connections. If modified, the CO shall ensure that only Approved cipher suites are configured while in the Approved mode. It is recommended to use the list of approved TLS cipher suites in section 3.3 of NIST SP 800-52*rev2* as guidance.
- The module must be configured to use PSK[18]-based authentication for IPsec connections. The CO shall provide a PSK value when configuring IPsec profiles via the GUI, CLI, or API. Configuring digital certificate-based authentication for IPsec connections is **prohibited** while in the Approved mode of operation.
- Kerberos traffic management/SSO shall not be configured or used in the Approved mode of operation.
- The CO must ensure that the “Key” and “AutoKey” authentication parameters are not set when adding NTP servers via the web GUI, CLI, or API.
- The module has built-in CA tools used to create self-signed certificates for testing purposes. While the feature does include the generation of keys, those keys are not considered CSPs (as they are not being used for production purposes). The CO shall ensure that all certificates are signed using a trusted CA and not by a self-signed certificate.
- The operator shall not enable the LOM port via ADC configuration.
- The TLS Ticket Encryption Key and TLS Ticket Authentication Key can be entered manually via the CLI or generated by the module using the data plane DRBG. Operators shall not enter these keys manually.
- If any irregular activity is noticed or the module is consistently reporting errors, then CSG Customer Support should be contacted.

11.3 Non-Administrator Guidance

Operators with the User role do not have the ability to configure sensitive information on the module. They must be diligent in selecting strong passwords and must not reveal their password to anyone. Additionally, they must be careful to protect any secret or private keys in their possession.

11.4 Design and Rules

By design, the module follows or enforces the following rules of operation:

- The module provides two distinct operator roles: User and Cryptographic Officer.

[18] PSK – Pre-Shared Key

- An operator does not have access to any cryptographic services prior to assuming an authorized role.
- The module performs all self-tests without any operator action required.
- The module inhibits data output during key generation, self-tests, zeroization, and error states.
- Status information output by the module does not contain CSPs or sensitive data that if misused could lead to a compromise of the module.
- The module does not support a maintenance interface or role.
- The module does not support manual SSP establishment methods.
- The module does not have any proprietary external input/output devices used for entry/output of data.
- The module does not output intermediate key values.
- The module does not provide bypass services or ports/interfaces.

11.5 End of Life

As the module does not persistently store SSPs, secure sanitization of the module consists of performing the zeroization methods described in section 9.3 above. This will ensure that any SSPs in volatile memory are zeroized.

12. Mitigation of Other Attacks

The module does not claim to mitigate any attacks beyond the FIPS 140-3 Level 1 requirements for this validation. Therefore, per ISO/IEC 19790:2012 section 7.12, requirements for this section are not applicable.

Appendix A. Acronyms and Abbreviations

Table 28 provides definitions for the acronyms and abbreviations used in this document.

Table 28. Acronyms and Abbreviations

Acronym	Definition
AES	Advanced Encryption Standard
API	Application Programming Interface
CBC	Cipher Block Chaining
CCCS	Canadian Centre for Cyber Security
CMVP	Cryptographic Module Validation Program
CO	Cryptographic Officer
CPU	Central Processing Unit
CSP	Critical Security Parameter
CTR	Counter
CVL	Component Validation List
DEP	Default Entry Point
DES	Data Encryption Standard
DH	Diffie-Hellman
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book
ECC CDH	Elliptic Curve Cryptography Cofactor Diffie-Hellman
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
EMI/EMC	Electromagnetic Interference /Electromagnetic Compatibility
FIPS	Federal Information Processing Standard
GCM	Galois/Counter Mode
GMAC	Galois Message Authentication Code
GPC	General-Purpose Computer
HMAC	(keyed-) Hash Message Authentication Code
KAS	Key Agreement Scheme
KAT	Known Answer Test
KTS	Key Transport Scheme
KW	Key Wrap
KWP	Key Wrap with Padding
NIST	National Institute of Standards and Technology

Acronym	Definition
OS	Operating System
PCT	Pairwise Consistency Test
PKCS	Public Key Cryptography Standard
PSK	Pre-Shared Key
PSS	Probabilistic Signature Scheme
RNG	Random Number Generator
RSA	Rivest, Shamir, and Adleman
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SP	Special Publication
TDES	Triple Data Encryption Standard

Prepared by:

Corsec Security, Inc.

12600 Fair Lakes Circle, Suite 210
Fairfax, VA 22033
United States of America

Phone: +1 703 267 6050
Email: info@corsec.com
Web: www.corsec.com