Cloud Software Group

NetScaler Virtual Appliance

Software Version: 13.1.FIPS

FIPS 140-3 Non-Proprietary Security Policy

FIPS Security Level: 1
Document Version: 0.5

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

Abstract

This is a non-proprietary Cryptographic Module Security Policy for the NetScaler Virtual Appliance (version: 13.1.FIPS) from Cloud Software Group (Cloud Software Group). This Security Policy describes how the NetScaler Virtual Appliance 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 FIPS-Approved mode of operation. This policy was prepared as part of the Level 1 FIPS 140-3 validation of the module. The NetScaler Virtual Appliance is referred to in this document as NetScaler VA or the module.

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 Cloud Software Group website www.cloud.com contains information on the full line of services and solutions from Cloud Software Group.
- 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.

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.

Table of Contents

List of Tables

List of Figures

1. General

1.1 Overview

NetScaler Virtual Appliance is a virtual application delivery controller (ADC) that accelerates application performance, enhances application availability with advanced L4-L7[1] load balancing, provides an integrated application firewall, and lowers server expenses by offloading computationally intensive tasks. All these capabilities are combined into a single, integrated virtual appliance.
NetScaler Virtual Appliance is provides the web-based GUI[2], REST[3]ful Nitro API[4], and CLI[5] interfaces for configuring and managing the appliance. The GUI includes a configuration utility for configuring the appliance as well as a statistical utility called Dashboard.

NetScaler appliances are 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. In a typical installation, 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. Administrators enable appliance features and apply configured policies to incoming and outgoing traffic.

The NetScaler Virtual Appliance is 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 Virtual Appliance is 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[7] or TCP[8] request, and on the basis of L4–L7 header information such as URL[9], 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 Virtual Appliance is security and protection features protect web applications from Application Layer attacks. The NetScaler Virtual Appliance is 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[10] injection attempts, cross-site scripting attacks, and

[1] L4-L7 – Layer 4 through 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] HTTP – Hypertext Transfer Protocol
[8] TCP – Transmission Control Protocol
[9] URL – Universal Resource Locator
[10] SQL – Structured Query Language

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[11] 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 speeds up applications. The NetScaler Virtual Appliance is 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.

1.2 Security Levels

The NetScaler Virtual Appliance 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
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

The module has an overall security level of 1.

[11] SSL – Secure Sockets Layer

2. Cryptographic Module Specification

2.1 Description

2.1.1 Purpose and Use

NetScaler Virtual Appliance is a virtual application delivery controller (ADC) that accelerates application performance, enhances application availability with advanced L4-L7[12] load balancing, provides an integrated application firewall, and lowers server expenses by offloading computationally intensive tasks. All these capabilities are combined into a single, integrated virtual appliance.

NetScaler Virtual Appliance is provides the web-based GUI[13], REST[14]ful Nitro API[15], and CLI[16] interfaces for configuring and managing the appliance. The GUI includes a configuration utility for configuring the appliance as well as a statistical utility called Dashboard.

2.1.2 Module Type

The NetScaler Virtual Appliance 13.1.FIPS is a Software module.

2.1.3 Module Embodiment

The NetScaler Virtual Appliance has a MultiChipStand embodiment.

2.1.4 Module Characteristics

The module does not have any additional characteristics.

2.1.5 Cryptographic Boundary

The logical cryptographic boundary of the module (shown by the red dotted line in Figure 1) consists of the VPX virtual appliance software and FreeBSD operating system acting as the guest OS.

[12] L4-L7 – Layer 4 through Layer 7
[13] GUI – Graphical User Interface
[14] REST – Representational State Transfer
[15] API – Application Programming Interface
[16] CLI – Command Line Interface

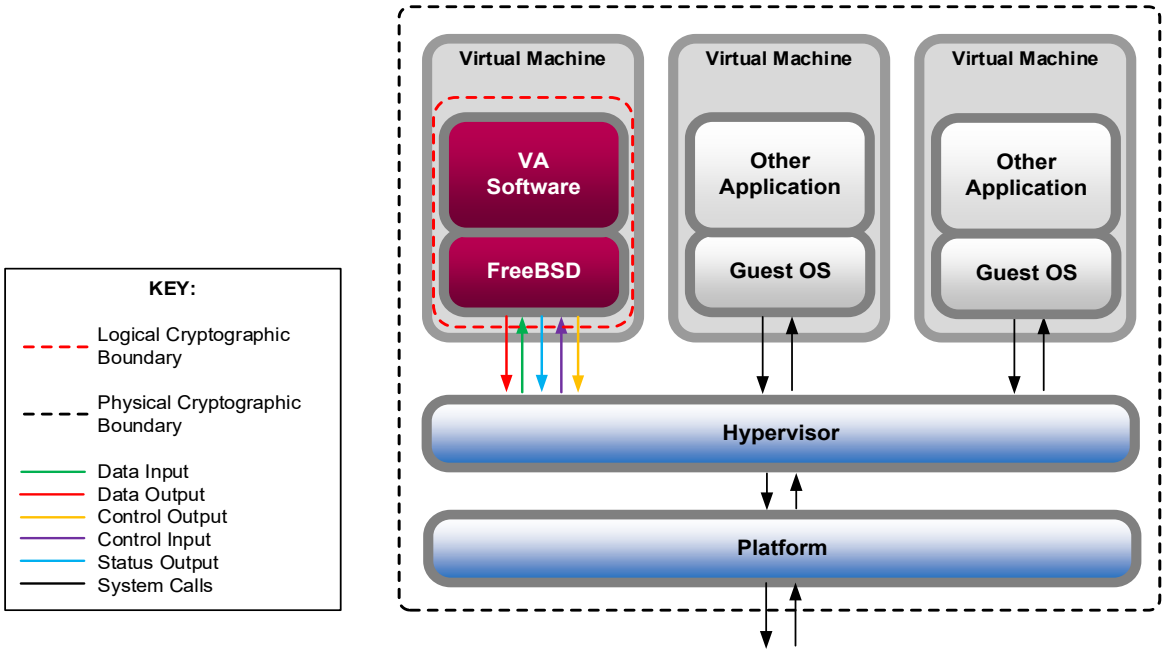

Figure 1. NetScaler Virtual Appliance Cryptographic Boundary

2.1.6 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 physical boundary of the cryptographic module is defined by the hard enclosure around the host server on which it runs. For this validation, the module will be 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 hardware case, a power supply, and interface ports.

Figure 2 displays the hardware components of the server used for testing (the dashed line surrounding the hardware components represents the module's physical cryptographic boundary, which is the outer case of the server), and identifies the hardware with which the processors interface.

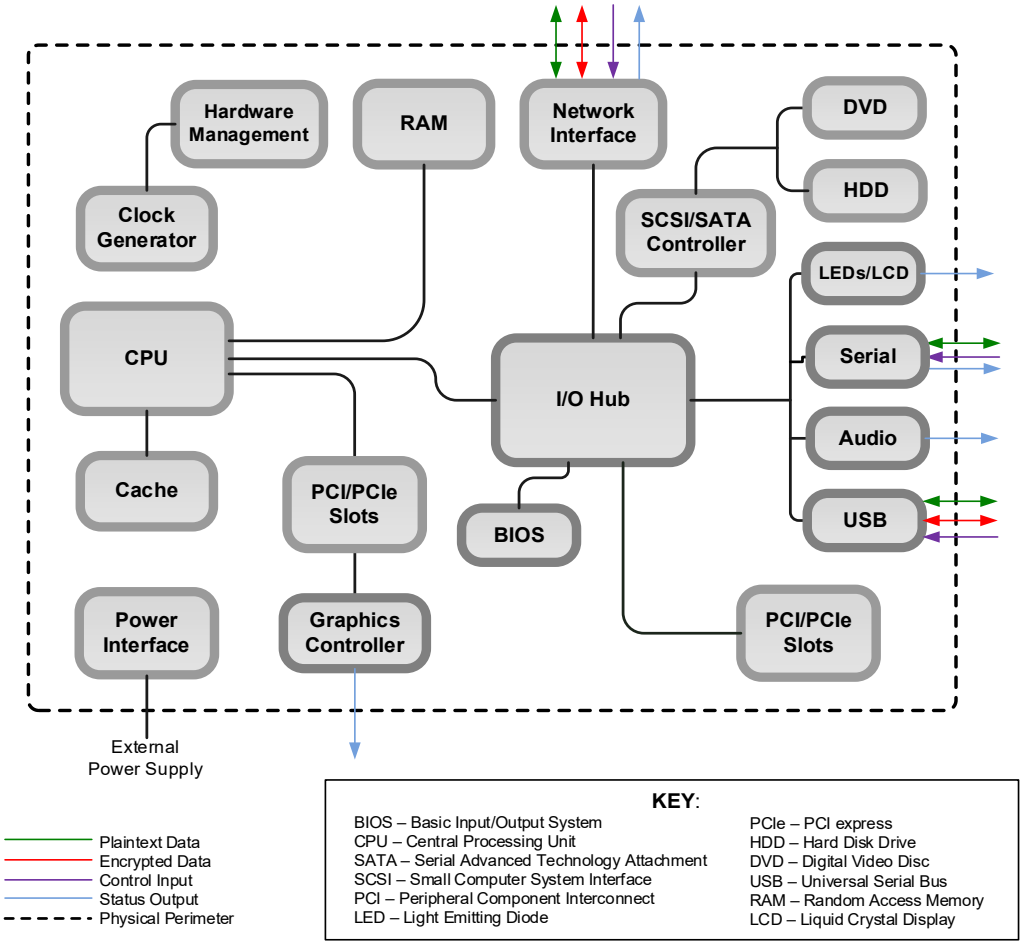

Figure 2 - GPC Block Diagram

Figure 3. Module Block Diagram (with Cryptographic Boundaries)

2.2 Tested and Vendor Affirmed Module Version and Identification

2.2.1 Tested Module Identification – Hardware

This section does not apply to this module.

N/A for this 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
ns-13.1-37.241.gz	13.1.FIPS	Contains FreeBSD OS and Netscaler Virtual Appliance software in an image.	2048-bit RSA with SHA2-512

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

2.2.3 Tested Module Identification – Hybrid Disjoint Hardware

The module does not have any hybrid disjoint hardware.

N/A for this module.

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 7.0U3	13.1.FIPS
FreeBSD 11.4	Dell PowerEdge R630	Intel Xeon E5-2680	No	VMWare ESXi 7.0U3	13.1.FIPS

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

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

There are no vendor-affirmed operational environments claimed.

N/A for this module.

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

When installed, configured, and operated according to this Security Policy, the module supports the Approved mode of operation only; non-Approved operations are not supported.

Mode Name	Description	Type	Status Indicator
Approved	Only supported mode of operation of the module	Approved	Global Indicator

Table 4: 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 v1.0 (Cert. A3942)
- NetScaler Data Plane Cryptographic Library v1.0 (Cert. A3943)
- NetScaler CPU Jitter Entropy Source v3.4.0 (Cert. A3513)

Control Plane

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A3942	-	SP 800-38A
AES-CFB128	A3942	-	SP 800-38A
AES-CTR	A3942	-	SP 800-38A
AES-GCM	A3942	-	SP 800-38D
Counter DRBG	A3942	-	SP 800-90A Rev. 1
ECDSA KeyGen (FIPS186-4)	A3942	-	FIPS 186-4
ECDSA KeyVer (FIPS186-4)	A3942	-	FIPS 186-4
ECDSA SigGen (FIPS186-4)	A3942	-	FIPS 186-4
ECDSA SigVer (FIPS186-4)	A3942	-	FIPS 186-4
HMAC-SHA-1	A3942	-	FIPS 198-1
HMAC-SHA2-256	A3942	-	FIPS 198-1
HMAC-SHA2-384	A3942	-	FIPS 198-1
HMAC-SHA2-512	A3942	-	FIPS 198-1
KAS-ECC-SSC Sp800-56Ar3	A3942	-	SP 800-56A Rev. 3
KAS-FFC-SSC Sp800-56Ar3	A3942	-	SP 800-56A Rev. 3
KDF IKEv1 (CVL)	A3942	-	SP 800-135 Rev. 1
KDF IKEv2 (CVL)	A3942	-	SP 800-135 Rev. 1
KDF SNMP (CVL)	A3942	-	SP 800-135 Rev. 1
KDF SSH (CVL)	A3942	-	SP 800-135 Rev. 1
KDF TLS (CVL)	A3942	-	SP 800-135 Rev. 1
KTS-IFC	A3942	-	SP 800-56B Rev. 2
PBKDF	A3942	-	SP 800-132
RSA KeyGen (FIPS186-4)	A3942	-	FIPS 186-4
RSA SigGen (FIPS186-4)	A3942	-	FIPS 186-4
RSA SigVer (FIPS186-4)	A3942	-	FIPS 186-4
Safe Primes Key Generation	A3942	-	SP 800-56A Rev. 3
Safe Primes Key Verification	A3942	-	SP 800-56A Rev. 3
SHA-1	A3942	-	FIPS 180-4
SHA2-256	A3942	-	FIPS 180-4
SHA2-384	A3942	-	FIPS 180-4
SHA2-512	A3942	-	FIPS 180-4
TLS v1.2 KDF RFC7627 (CVL)	A3942	-	SP 800-135 Rev. 1

Table 5: Approved Algorithms - Control Plane

Data Plane

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A3943	-	SP 800-38A
AES-GCM	A3943	-	SP 800-38D
ECDSA KeyGen (FIPS186-4)	A3943	-	FIPS 186-4
ECDSA KeyVer (FIPS186-4)	A3943	-	FIPS 186-4
ECDSA SigGen (FIPS186-4)	A3943	-	FIPS 186-4
ECDSA SigVer (FIPS186-4)	A3943	-	FIPS 186-4
Hash DRBG	A3943	-	SP 800-90A Rev. 1
HMAC-SHA-1	A3943	-	FIPS 198-1
HMAC-SHA2-224	A3943	-	FIPS 198-1
HMAC-SHA2-256	A3943	-	FIPS 198-1
HMAC-SHA2-384	A3943	-	FIPS 198-1
HMAC-SHA2-512	A3943	-	FIPS 198-1
KAS-ECC-SSC Sp800-56Ar3	A3943	-	SP 800-56A Rev. 3
KDF SP800-108	A3943	-	SP 800-108 Rev. 1
KDF TLS (CVL)	A3943	-	SP 800-135 Rev. 1
KTS-IFC	A3943	-	SP 800-56B Rev. 2
RSA SigGen (FIPS186-4)	A3943	-	FIPS 186-4
RSA SigVer (FIPS186-2)	A3943	-	FIPS 186-4
RSA SigVer (FIPS186-4)	A3943	-	FIPS 186-4
SHA-1	A3943	-	FIPS 180-4
SHA2-224	A3943	-	FIPS 180-4
SHA2-256	A3943	-	FIPS 180-4
SHA2-384	A3943	-	FIPS 180-4
SHA2-512	A3943	-	FIPS 180-4
TLS v1.2 KDF RFC7627 (CVL)	A3943	-	SP 800-135 Rev. 1
TLS v1.3 KDF (CVL)	A3943	-	SP 800-135 Rev. 1

Table 6: Approved Algorithms - Data Plane

CPU Jitter Entropy Source

Algorithm	CAVP Cert	Properties	Reference
SHA3-256	A3513	-	FIPS 202

Table 7: Approved Algorithms - CPU Jitter Entropy Source

2.5.2 Vendor Affirmed Algorithms

The vendor affirms the following cryptographic security methods:

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

Table 8: 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	N/A	Message digest in TLS 1.0/1.1

Table 9: 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 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	AES for KEK, which is used for encrypting/decrypting passwords and passphrases	Publication:NIST SP 800-38A	AES-CBC: (A3942) Key Length: 256 Counter DRBG: (A3942)
AES for AES Key	BC-UnAuth	AES for the AES Key, which is used for encryption/decryption	Publication:NIST SP 800-38A	AES-CBC: (A3943) Hash DRBG: (A3943) SHA2-256: (A3943)
Key Derivation for TLS Extended Master Secret	KAS-135KDF	Key derivation for the TLS Extended Master Secret, which are used to derive the ticket encryption key and the ticket authentication key	Publication:NIST SP 800-135rev1	KDF TLS: (A3942, A3943) TLS v1.2 KDF RFC7627: (A3942, A3943) TLS v1.3 KDF: (A3943) SHA2-256: (A3942, A3943) SHA2-384: (A3942, A3943)
AES for TLS Ticket	BC-UnAuth	AES for the TLS Ticket Encryption Key, which is used for the encryption/decryption of TLS session tickets	Publication:NIST SP 800-38A	AES-CBC: (A3943) Key Length: 128 Hash DRBG: (A3943)
HMAC for TLS Ticket	MAC	HMAC for the TLS Ticket Authentication Key, which is used for the authentication of TLS session tickets	Publication:FIPS 198-1	HMAC-SHA2-256: (A3942, A3943) SHA2-256: (A3942, A3943)

Name	Type	Description	Properties	Algorithms
Key Generation for SSH	AsymKeyPair-KeyGen	Asymmetric key generation (RSA or ECDSA) for the SSH private and public keys	Publication:FIPS 186-4, FIPS 186-5	ECDSA KeyGen (FIPS186-4): (A3942) RSA KeyGen (FIPS186-4): (A3942) Counter DRBG: (A3942) CKG (Control Plane - VA): () CKG: Symmetric
Key Agreement for SSH (DH)	KAS-135KDF	Key agreement for SSH using DH	Publication:NIST SP 800-135rev1, NIST SP 800-56Arev3 Key Strength:Key establishment methodology provides between 112 and 176 bits of encryption strength Caveat:No part of the SSH protocol, other than the KDF, has been tested by the CAVP and CMVP	KDF SSH: (A3942) KAS-FFC-SSC Sp800-56Ar3: (A3942) Domain parameter generation methods: MODP-2048, MODP-4096, ffdhe2048, ffdhe4096 Counter DRBG: (A3942) Safe Primes Key Generation: (A3942) Safe prime groups: MODP-2048, MODP-4096, ffdhe2048, ffdhe4096 Safe Primes Key Verification: (A3942) Safe prime groups: MODP-2048, MODP-4096, ffdhe2048, ffdhe4096 SHA-1: (A3942) SHA2-256: (A3942) SHA2-384: (A3942) SHA2-512: (A3942) ECDSA SigGen (FIPS186-4): (A3942) ECDSA SigVer (FIPS186-4): (A3942) RSA SigGen (FIPS186-4): (A3942) RSA SigVer (FIPS186-4): (A3942)

Name	Type	Description	Properties	Algorithms
Key Agreement for SSH (ECDH)	KAS-135KDF	Key agreement for SSH utilizing ECDH	Publication:NIST SP 800-135rev1, NIST SP 800-56Arev3 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	KDF SSH: (A3942) KAS-ECC-SSC Sp800-56Ar3: (A3942, A3943) Counter DRBG: (A3942) ECDSA KeyGen (FIPS186-4): (A3942) ECDSA KeyVer (FIPS186-4): (A3942) SHA-1: (A3942) SHA2-256: (A3942) SHA2-384: (A3942) SHA2-512: (A3942) RSA SigGen (FIPS186-4): (A3942) RSA SigVer (FIPS186-4): (A3942) ECDSA SigGen (FIPS186-4): (A3943) ECDSA SigVer (FIPS186-4): (A3942) AES-CBC: (A3942) AES-CTR: (A3942)
AES for SSH	BC-UnAuth	AES (CTR or CBC modes) for the SSH Session Key, which is used for the encryption/decryption of SSH session packets	Publication:NIST SP 800-38A	AES-CBC: (A3942) AES-CTR: (A3942) Counter DRBG: (A3942)
HMAC for SSH	MAC	HMAC for the SSH Authentication Key, which is used for the authentication of SSH session packets	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: (A3942) SHA-1: (A3942) HMAC-SHA2-256: (A3942) SHA2-256: (A3942) HMAC-SHA2-384: (A3942) SHA2-384: (A3942) HMAC-SHA2-512: (A3942) SHA2-512: (A3942)

Name	Type	Description	Properties	Algorithms
Key Agreement for IKE/IPsec (DH)	KAS-135KDF	Key Agreement for IKE/IPsec using DH	Publication:NIST SP 800-135rev1, NIST SP 800-56Arev3 Key Strength:Key establishment methodology provides between 112 and 176 bits of encryption strength Caveat:No part of the IKE protocol, other than the KDF, has been tested by the CAVP and CMVP	KDF IKEv1: (A3942) KDF IKEv2: (A3942) KAS-FFC-SSC Sp800-56Ar3: (A3942) Domain parameter generation methods: MODP-2048, MODP-3072, MODP-6144 Counter DRBG: (A3942) Safe Primes Key Generation: (A3942) Safe prime groups: MODP-2048, MODP-3072, MODP-6144 Safe Primes Key Verification: (A3942) Safe prime groups: MODP-2048, MODP-3072, MODP-6144 SHA-1: (A3942) SHA2-256: (A3942) SHA2-384: (A3942) SHA2-512: (A3942)
AES for IKE/IPsec	BC-UnAuth	AES for the IKE/IPsecSession Key, which is used for the encryption/decryption of IKE/IPsec packets	Publication:NIST SP 800-38A	AES-CBC: (A3942)
HMAC for IKE/IPsec	MAC	HMAC for the IKE/IPsec Authentication Key, which is used for the authentication of IKE/IPsec session packets	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: (A3943) SHA-1: (A3943) HMAC-SHA2-224: (A3943) SHA2-224: (A3943) HMAC-SHA2-256: (A3943) SHA2-256: (A3943) HMAC-SHA2-384: (A3943) SHA2-384: (A3943) HMAC-SHA2-512: (A3943) SHA2-512: (A3943)
HMAC for HMAC key	MAC	HMAC for HMAC key used in message authentication	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: (A3943) SHA-1: (A3943) HMAC-SHA2-224: (A3943) SHA2-224: (A3943) HMAC-SHA2-256: (A3943) SHA2-256: (A3943) HMAC-SHA2-384: (A3943) SHA2-384: (A3943) HMAC-SHA2-512: (A3943) SHA2-512: (A3943)

Name	Type	Description	Properties	Algorithms
Key Agreement for TLS (DH)	KAS-135KDF	Key Agreement for TLS using DH	Publication:RFC 7627, NIST SP 800-135rev1, NIST SP 800-56Arev3 Key Strength:Key establishment methodology provides between 112 and 176 bits of encryption strength Caveats:No part of the TLS protocol, other than the KDF, has been tested by the CAVP and CMVP	KDF TLS: (A3942) TLS v1.2 KDF RFC7627: (A3942) KAS-FFC-SSC Sp800-56Ar3: (A3942) Domain parameter generation methods: fdhe2048, ffdhe3072, ffdhe4096, ffdhe6144 Counter DRBG: (A3942) Safe Primes Key Generation: (A3942) Safe prime groups: ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144 Safe Primes Key Verification: (A3942) Safe prime groups: ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144 SHA2-256: (A3942) SHA2-384: (A3942) ECDSA SigGen (FIPS186-4): (A3943) ECDSA SigVer (FIPS186-4): (A3943) RSA SigGen (FIPS186-4): (A3942) RSA SigVer (FIPS186-4): (A3942)
Key agreement for TLS (ECDH)	KAS-135KDF	Key agreement for TLS using ECDH	Publication:RFC 7627, NIST SP 800-135rev1, NIST SP 800-56Arev3 Key strength:Key establishment methodology provides between 112 and 256 bits of encryption strength Caveat:No part of the TLS protocol, other than the KDF, has been tested by the CAVP and CMVP	KDF TLS: (A3942, A3943) TLS v1.2 KDF RFC7627: (A3942, A3943) KAS-ECC-SSC Sp800-56Ar3: (A3942, A3943) Counter DRBG: (A3942) ECDSA KeyGen (FIPS186-4): (A3942, A3943) ECDSA KeyVer (FIPS186-4): (A3942, A3943) SHA2-256: (A3942, A3943) SHA2-384: (A3942, A3943) ECDSA SigGen (FIPS186-4): (A3942, A3943) ECDSA SigVer (FIPS186-4): (A3942, A3943) RSA SigGen (FIPS186-4): (A3942, A3943) RSA SigVer (FIPS186-4): (A3942) RSA SigVer (FIPS186-2): (A3943) Hash DRBG: (A3943)

Name	Type	Description	Properties	Algorithms
Key Transport for TLS	KTS-Encap	RSA key transport for TLS	Publication:FIPS 186-4, FIPS 186-5, NIST SP 800-56Brev2 Key Strength:Key establishment methodology provides 112 bits of encryption strength	KTS-IFC: (A3942, A3943) Counter DRBG: (A3942) Hash DRBG: (A3943) SHA-1: (A3942, A3943) RSA KeyGen (FIPS186-4): (A3942) Modulus: 2048 RSA SigVer (FIPS186-4): (A3942) RSA SigVer (FIPS186-2): (A3943)
Key Generation for TLS	AsymKeyPair-KeyGen	Key pair generation for TLS	Publication:FIPS 186-4, FIPS 186-5	RSA KeyGen (FIPS186-4): (A3942) ECDSA KeyGen (FIPS186-4): (A3942, A3943) Counter DRBG: (A3942) Hash DRBG: (A3943)
AES for TLS session	BC-UnAuth	AES for the TLS Session Key, which is used for the encryption/decryption of TLS session packets	Publication:NIST SP 800-38A	AES-CBC: (A3942, A3943) Key Length: 128, 256 Counter DRBG: (A3942) Hash DRBG: (A3943)
HMAC for TLS Session	MAC	HMAC for the TLS Authentication Key, which is used for the authentication of TLS session packets	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: (A3942, A3943) SHA-1: (A3942, A3943) HMAC-SHA2-256: (A3942, A3943) SHA2-256: (A3942, A3943) HMAC-SHA2-384: (A3942, A3943) SHA2-384: (A3942, A3943) HMAC-SHA2-224: (A3943) SHA2-224: (A3943)
AES GCM for TLS Session	BC-Auth	AES GCM for the TLS session key, which is used for the encryption/decryption of TLS session packets	Publication:NIST SP 800-38D IG:C.H	AES-GCM: (A3942, A3943) Key Length: 128, 256 Counter DRBG: (A3942) Hash DRBG: (A3943)
RSA SigGen for DNSsec	DigSig-SigGen	RSA digital signature generation for DNSsec	Publication:FIPS 186-4, FIPS 186-5	RSA SigGen (FIPS186-4): (A3942) Counter DRBG: (A3942) SHA-1: (A3942)
RSA SigVer for DNSsec	DigSig-SigVer	RSA digital signature verification for DNSSec	Publication:FIPS 186-4, FIPS 186-5	RSA SigVer (FIPS186-4): (A3942) Counter DRBG: (A3942)
AES (PEM Key) for Encrypting TLS Private Key	BC-UnAuth	AES for the PEM Key, which is used to encrypt the TLS Private Key	Publication:NIST SP 800-38A	AES-CBC: (A3942) Key Length: 256 HMAC-SHA-1: (A3942) SHA-1: (A3942)

Name	Type	Description	Properties	Algorithms
PBKDF	PBKDF	Password-based key derivation for PEM Key used for the encryption and decryption of asymmetric private keys	Publication:NIST SP 800-132	PBKDF: (A3942) SHA-1: (A3942)
AES for RDP Session	BC-Auth	AES-GCM for the RDP Session Key, which is used for the encryption /decryption of RDP user and target information	Publication:NIST SP 800-38D	AES-GCM: (A3942) Key Length: 256 Counter DRBG: (A3942) KDF SP800-108: (A3943) HMAC-SHA2-224: (A3943) SHA2-256: (A3943)
KBKDF for DFA Shared Secret	KBKDF	Key-based key derivation for DFA Shared Secret	Pubication:NIST SP 800-108rev1	KDF SP800-108: (A3943) HMAC-SHA2-256: (A3943) SHA2-256: (A3943)
AES for DFA Session Key	BC-UnAuth	AES for the DFA Session Key, which is used for DFA authentication to the module	Publication:NIST SP 800-38A	AES-CBC: (A3943) Key Length: 256 KDF SP800-108: (A3943) HMAC-SHA2-256: (A3943) SHA2-256: (A3943)
AES for SNMPv3	BC-UnAuth	AES (SNMPv3 Privacy Key) for the encryption and decryption of SNMPv3 packets	Publication:NIST SP 800-38A	AES-CFB128: (A3942) Key Length: 128 KDF SNMP: (A3942) Counter DRBG: (A3942)
HMAC for SNMPv3	MAC	HMAC (SNMPv3 Authentication Key) for the authentication of SNMPv3 packets	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: (A3942) KDF SNMP: (A3942) SHA-1: (A3942)
RSA SigVer for Software Load Integrity	DigSig-SigVer	RSA SigVer used to verify the new software load	Publication:FIPS 186-4, FIPS 186-5	RSA SigVer (FIPS186-2): (A3943) Signature Type: pkcs1v1.5 Modulus: 2048 SHA2-512: (A3943)
RSA SigVer for Web GUI	DigSig-SigVer	RSA SigVer for authentication via the Web GUI	Publication:FIPS 186-4, FIPS 186-5	RSA SigVer (FIPS186-4): (A3943) SHA2-256: (A3943) SHA2-384: (A3943) SHA2-512: (A3943)
Entropy Source	ENT-Cond	CPU Jitter Entropy Source with SHA3 conditioning component	Publication:NIST SP 800-90B	SHA3-256: (A3513)

Table 10: Security Function Implementations

2.7 Algorithm Specific Information

The following is algorithm specific information for the module:

- AES GCM: The AES-GCM algorithm is used in the TLS v1.2 and TLS v1.3 protocols.

- 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 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.

2.8 RNG and Entropy

The table below specifies the module's entropy certificates.

Cert Number	Vendor Name
E52	Cloud Software Group

Table 11: 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 on VMware ESXi 7 on Intel® Xeon® E5 v4 (Broadwell) Family without PAA	256 bits	Full entropy	SHA3-256 (A3513)

Table 12: Entropy Sources

2.9 Key Generation

When generating symmetric keys, the module uses the direct output of its approved DRBG to generate random numbers and seeding material, per the guidance in *NIST SP 800-133 Rev. 2*, Section 4.

When generating the Key Encryption Key (KEK), the module follows the method "Symmetric Keys Produced by Combining (Multiple) Keys and Other Data" described in *NIST SP 800-133 Rev. 2*, Section 6.3.

2.10 Key Establishment

2.10.1 Key Agreement Information

The module implements the following approved key agreement methods:

KAS-ECC-SSC – *NIST SP 800-56A Rev. 3* (*FIPS 140-3 IG* D.F, Scenario 2, Path (2))
KAS-FFC-SSC – *NIST SP 800-56A Rev. 3* (*FIPS 140-3 IG* D.F, Scenario 2, Path (2))

Key confirmation is not supported.

2.10.2 Key Transport Information

The module implements the following key transport method:

KTS-IFC – *NIST SP 800-56B Rev. 2* (*FIPS 140-3 IG* D.G, Key Encapsulation/Un-encapsulation)

2.11 Industry Protocols

The module uses the following industry protocols:

- IPsec with IKEv1
- IPsec with IKEv2
- SNMP
- SSH
- TLS 1.0/1.1
- TLS 1.2
- TLS 1.3

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, NetScaler VA has no physical characteristics. Its interfaces are logical; the hypervisor provides virtualized ports and interfaces for the module that maps 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	Data to be encrypted, decrypted, signed, verified, or hashed; Keys to be used in cryptographic services; Random seed material for the module's DRBG; Keying material to be used as input to key establishment services
N/A	Data Output	Data that has been encrypted, decrypted, or verified; Digital signatures; Hashes; Random values generated by the module's DRBG; Keys established using module's key establishment methods
N/A	Control Input	API commands invoking cryptographic services; Modes, key sizes, etc. used with cryptographic services
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	Includes status information regarding the module and status information regarding the invoked service/operation.

Table 13: Ports and Interfaces

4. Roles, Services, and Authentication

4.1 Authentication Methods

The module supports identity-based authentication; operators explicitly assume their role based on the authentication credentials used. Each role determines the functionality available to the operator within the module.

Operators authenticate to the module using either:

- a username and password. Password complexity policies can be configured by an operator with the Crypto Officer role and are enforced by the module. All operators are required to follow the password policies.
- Certificates associated with the selected protocol. The module supports RSA digital certificate authentication of users during Web GUI/HTTPS (TLS) access.

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 these objectives, the password policies shall be configured by the Crypto Officer such that all passwords shall require:

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

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	Password complexity policies must be configured by the Crypto Officer to include the following: a minimum of eight total characters, at least one lowercase letter, at least one uppercase letter, at least one digit, at least one special character (~, `, !, @, #, $, %, ^, &, *, -, _, =, +, {, }, [,], |, \, :, <, >, /, ., ,, " ").	Username / Password	1/11451713827320	1/4591650
Certificate	The module supports RSA digital certificate authentication of users during Web GUI/HTTPS (TLS) access.	Placeholder for SFI entry	1 per 2^122, 1 per 5.19 * 10^33	1/77 * 10^26

Table 14: Authentication Methods

4.2 Roles

The module supports a Crypto Officer (CO) that authorized operators can assume. 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.
The module also supports the following role(s):

- User – The User role can view the current status of the module and employ the services of the module (including IPsec[17], TLS, SSH, and SNMPv3 services). The User role includes the privileges listed under the read-only command policy.

The table below lists the supported roles.

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

Table 15: Roles

4.3 Approved Services

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

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.

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Show status	Show the system status	N/A	Command	Status output	None	Crypto Officer
Perform self-tests on demand	Perform pre-operational self-tests	Log file	Command	Status output	None	Crypto Officer
Perform initial network configuration	Set up initial network configuration and licenses	Success from CLI	Command and parameters	Command response/status output	CKG (KEK)	Crypto Officer - KEK Fragment 1: R - KEK Fragment 2: R
Show versioning information	Show module name and version	Console Output	Command	Module name, version	None	Crypto Officer

[17] IPsec – Internet Protocol Security

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
View system information	View system info and statistics; view/end system sessions	Console Output	Command	Status output	None	Crypto Officer
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 AES Key	Crypto Officer - AES Key: W - KEK (AES Key): E - Hash DRBG Entropy: R,E - Hash DRBG Seed: R,W,E - Hash DRBG 'V' Value (Internal state value): R,W,E - Hash DRBG 'C' Value (Internal state value): R,W,E
Configure HA	Configure HA nodes, route monitors, failover interface set	Command Line Interface and Traffic	Command and parameters	Status output/control output	None	Crypto Officer
Manage NTP servers	Add, edit, delete NTP servers; configure NTP parameters and synchronization state	Command Line Interface	Command	Status output / control output	None	Crypto Officer

Zeroize	Procedural Zeroization: Operator initiated reboot/power-cycle of Module/TOEPP	N/A	Command	Status output	None	Crypto Officer - PEM Passphrase (Alphanumeric string): Z - PEM Key (AES Key): Z - AES Key: Z - AES GCM Key: Z - AES GCM IV (96 and 128-bit IV): Z - DH Public Key: Z - DH Private Key: Z - ECDH Public Key : Z - ECDH Private Key : Z - HMAC Key: Z - RDP PSK (Shared secret): Z - RSA Public Key: Z - RSA Private Key : Z - SSH Shared Secret: Z - SSH Session Key: Z - SSH Authentication Key: Z - IKE/IPsec Shared Secret: Z - IKE/IPsec Session Key (AES key): Z - IKE/IPsec Authentication Key (HMAC key): Z - TLS Pre-Master Secret: Z - TLS Extended Master Secret: Z - TLS Session Key: Z - TLS Authentication Key (HMAC Key): Z - TLS Ticket Encryption Key (AES key): Z - TLS Ticket Authentication Key (HMAC key): Z - Hash DRBG Entropy: Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- Hash DRBG Seed: Z - Hash DRBG 'V' Value (Internal state value): Z - Hash DRBG 'C' Value (Internal state value): Z - CTR DRBG Entropy: Z - CTR DRBG Seed: Z - CTR DRBG 'V' Value: Z - CTR DRBG 'Key' Value (AES key): Z - SNMPv3 Privacy Key (AES key): Z - SNMPv3 Authentication Key (HMAC key): Z
Configure TLS profiles	Add, edit, delete system profiles	Command Line Interface	Command and parameters	Command response / status output	AES for Disk Encryption Key Derivation for TLS Extended Master Secret AES for TLS Ticket	Crypto Officer - TLS Extended Master Secret: R,W,E - TLS Ticket Encryption Key (AES key): R,W - TLS Ticket Authentication Key (HMAC key): R,W - CTR DRBG Entropy: R,E - CTR DRBG Seed: R,W,E - CTR DRBG 'V' Value: R,W,E - CTR DRBG 'Key' Value (AES key): R,W,E - KEK (AES Key): E
Manage users	Add, edit delete users, groups, and command policies; view user/group partition bindings	Command Line Interface	Command	Status output	None	Crypto Officer
Configure system auditing	Add, edit, delete syslog/nslog auditing policies and servers; bind classic/advanced global policies	Command Line Interface	Command and parameters	Command response / status output / control output	None	Crypto Officer
View audit logs	View authentication, system, and event logs	N/A (Audit Logs)	Command	Status output	None	Crypto Officer

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Configure network settings	Configure network routing protocols	Command Line Interface	Command Line Interface	Command response / status output	AES for Disk Encryption	Crypto Officer - ZebOS Router Password (Alphanumeric string): R,W - KEK (AES Key): 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 (AES Key): E
Configure SSH	Configure SSH authentication settings; generate SSH keys	Command Line Interface	Command and parameters	Command response / status output	Key Generation for SSH	Crypto Officer - SSH Private Key: W,E - SSH Public Key: W - CTR DRBG Entropy: R,E - CTR DRBG Seed: R,W,E - CTR DRBG 'V' Value: R,W,E - CTR DRBG 'Key' Value (AES key): R,W,E
Establish SSH sessions	Establish an SSH session	Traffic	Command	Status output	Key Agreement for SSH (DH) Key Agreement for SSH (ECDH) AES for SSH HMAC for SSH Entropy Source CKG (Control Plane) CKG (Data Plane)	Crypto Officer - SSH Public Key: R,E - DH Private Key: W,E - DH Public Key: R,E - ECDH Private Key : W,E - ECDH Public Key : R,E - SSH Shared Secret: W,E - SSH Session Key: W,E - SSH Authentication Key: W,E - CTR DRBG Entropy: R,E - CTR DRBG Seed: R,W,E - CTR DRBG 'V' Value: R,W,E - CTR DRBG 'Key' Value (AES key): R,W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
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	Crypto Officer - IKE/IPsec Pre-shared key (PSK): R,W - KEK (AES Key): 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): R,W
Establish IPsec session	Establish an IPsec Session	Traffic	Command	Status output	AES for Disk Encryption Key Agreement for IKE/IPsec (DH) AES for IKE/IPsec HMAC for IKE/IPsec Entropy Source CKG (Control Plane) CKG (Data Plane)	Crypto Officer - DH Private Key: W,E - DH Public Key: R,E - IKE/IPsec Shared Secret: W,E - IKE/IPsec Pre-shared key (PSK): E - KEK (AES Key): E - IKE/IPsec Session Key (AES key): W,E - IKE/IPsec Authentication Key (HMAC key): W,E - CTR DRBG Entropy: R,E - CTR DRBG Seed: R,W,E - CTR DRBG 'V' Value: R,W,E - CTR DRBG 'Key' Value (AES key): R,W,E
Backup and restore	Backup/import system configuration files; download and delete backup files; restore	N/A	Command	Status output / control output	None	Crypto Officer

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Manage data policy encryption keys	Add, edit, delete encryption keys	Command Line Interface	Command	Status output	AES for Disk Encryption AES for AES Key	Crypto Officer - AES Key: R,W - KEK (AES Key): E - Hash DRBG Entropy: R,E - Hash DRBG Seed: R,W,E - Hash DRBG 'V' Value (Internal state value): R,W,E - Hash DRBG 'C' Value (Internal state value): R,W,E
Manage data policy HMAC keys	Add, edit, delete HMAC keys	Command Line Interface	Command	Status output	AES for Disk Encryption HMAC for HMAC key	Crypto Officer - AES Key: R,W - KEK (AES Key): E - Hash DRBG Entropy: R,E - Hash DRBG Seed: R,W,E - Hash DRBG 'V' Value (Internal state value): R,W,E - Hash DRBG 'C' Value (Internal state value): R,W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Configure traffic management	Configure TLS; Configure load balancing, priority load balancing, content switching	Command Line Interface	Command and parameters	Command response / status output	Key Transport for TLS Key Generation for TLS RSA SigGen for DNSsec RSA SigVer for DNSsec AES (PEM Key) for Encrypting TLS Private Key PBKDF	Crypto Officer - CA Public Key: R,W,E - TLS Private Key: R,W,E - TLS Public Key: R,W - Private DNS KSK (RSA private key): R,W,E - Public DNS KSK (RSA public key): R,W - Private DNS ZSK (RSA private key): R,W,E - Public DNS ZSK (RSA public key): R,W - SSH Private Key: R,W,E - SSH Public Key: R,W,E - PEM Passphrase (Alphanumeric string): R,W,E - PEM Key (AES Key): W,E - CTR DRBG Entropy: R,E - CTR DRBG Seed: R,W,E - CTR DRBG 'V' Value: R,W,E - CTR DRBG 'Key' Value (AES key): R,W,E - Hash DRBG Entropy: R,E - Hash DRBG Seed: R,W,E - Hash DRBG 'V' Value (Internal state value): R,W,E - Hash DRBG 'C' Value (Internal state value): R,W,E - KEK (AES Key): E

Establish TLS session	Establish a web session using TLS protocol	Traffic	Command	Status output	Key Agreement for TLS (DH) Key agreement for TLS (ECDH) AES for TLS session HMAC for TLS Session AES GCM for TLS Session AES (PEM Key) for Encrypting TLS Private Key Entropy Source CKG (Control Plane) CKG (Data Plane)	Crypto Officer - TLS Public Key: R,E - DH Private Key: W,E - DH Public Key: R,E - ECDH Private Key : W,E - ECDH Public Key : R,E - RSA Private Key : W,E - RSA Public Key: R,E - TLS Pre-Master Secret: R,W,E - TLS Extended Master Secret: W,E - TLS Session Key: W,E - TLS Authentication Key (HMAC Key): W,E - AES GCM IV (96 and 128-bit IV): W,E - AES GCM Key: W,E - PEM Passphrase (Alphanumeric string): R,E - PEM Key (AES Key): W,E - KEK (AES Key): E - CTR DRBG Entropy: R,E - CTR DRBG Seed: R,W,E - CTR DRBG 'V' Value: R,W,E - CTR DRBG 'Key' Value (AES key): R,W,E - Hash DRBG Entropy: R,E - Hash DRBG Seed: R,W,E - Hash DRBG 'V' Value (Internal state value): R,W,E - Hash DRBG 'C' Value (Internal state value): R,W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Resume TLS session	Resume a web session using TLS protocol	Traffic	Command	Status output	AES for Disk Encryption AES for TLS Ticket HMAC for TLS Ticket AES for TLS session HMAC for TLS Session AES GCM for TLS Session	Crypto Officer - TLS Ticket Encryption Key (AES key): R,W,E - TLS Ticket Authentication Key (HMAC key): R,W,E - TLS Session Key: R,E - TLS Authentication Key (HMAC Key): R,E - AES GCM IV (96 and 128-bit IV): W,E - AES GCM Key: W,E - KEK (AES Key): E - Hash DRBG Entropy: R,E - Hash DRBG Seed: R,W,E - Hash DRBG 'V' Value (Internal state value): R,W,E - Hash DRBG 'C' Value (Internal state value): R,W,E
Apply data policies	Apply data policies to user data in transit (according to configuration)	Traffic	Command	Status output	AES for Disk Encryption AES for AES Key HMAC for HMAC key	Crypto Officer - AES Key: E - HMAC Key: E - KEK (AES Key): E
Configure security	Configure DNS security profiles, application firewall profiles and policies, reputation settings, protection features, and content inspection policies	Command Line Interface	Command and parameters	Command response / status output	None	Crypto Officer
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	Crypto Officer - RDP PSK (Shared secret): W - KEK (AES Key): E
Establish Gateway connection	Establish Gateway connection based on global settings	Traffic	Command and parameters	Command response / status output / control output	AES for RDP Session	Crypto Officer - RDP PSK (Shared secret): R,E - RDP Session Key: R,E - KEK (AES Key): E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Configure external servers for system, AAA, and Gateway authentication	Configure LDAP , Oauth, OpenID, DFA , and SAML servers to be used in system, AAA, or Gateway authentication	Command Line Interface	Command and parameters	Command response / status output	KBKDF for DFA Shared Secret AES for DFA Session Key	Crypto Officer - LDAP Admin Password (Alphanumeric string): R,W - Oauth Client Secret (Shared secret): R,W - DFA Shared Secret: R,W - KEK (AES Key): E - DFA Session Key: R,W,E

RADIUS Over TLS	Establish a TLS session with radius server	Traffic	Command	Status output	Key Agreement for TLS (DH) Key agreement for TLS (ECDH) AES for TLS session HMAC for TLS Session AES GCM for TLS Session AES (PEM Key) for Encrypting TLS Private Key Entropy Source CKG (Control Plane) CKG (Data Plane)	Crypto Officer - TLS Public Key: R,E - DH Private Key: W,E - DH Public Key: R,E - ECDH Private Key : W,E - ECDH Public Key : R,E - RSA Private Key : W,E - RSA Public Key: R,E - TLS Pre-Master Secret: R,W,E - TLS Extended Master Secret: W,E - TLS Session Key: W,E - TLS Authentication Key (HMAC Key): W,E - AES GCM IV (96 and 128-bit IV): W,E - AES GCM Key: W,E - PEM Passphrase (Alphanumeric string): R,E - PEM Key (AES Key): W,E - KEK (AES Key): E - CTR DRBG Entropy: R,E - CTR DRBG Seed: R,W,E - CTR DRBG 'V' Value: R,W,E - CTR DRBG 'Key' Value (AES key): R,W,E - Hash DRBG Entropy: R,E - Hash DRBG Seed: R,W,E - Hash DRBG 'V' Value (Internal state value): R,W,E - Hash DRBG 'C' Value (Internal state value): R,W,E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
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	Crypto Officer - SNMPv3 Authentication Passphrase (Alphanumeric string): R,W - SNMPv3 Privacy Passphrase (Alphanumeric string): R,W - KEK (AES Key): 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 (Alphanumeric string): E - SNMPv3 Privacy Passphrase (Alphanumeric string): E - SNMPv3 Privacy Key (AES key): W,E - SNMPv3 Authentication Key (HMAC key): W,E
Zeroize KEK	Zeroize KEK	API return value	Command	Status output	None	Crypto Officer - KEK (AES Key): W
Zeroize SSH private keys	Zeroize SSH private keys	API return value	Command	Status output	None	Crypto Officer - SSH Private Key: W
Authenticate operators	Used for operator logins to the module	Traffic	Command	Status output	RSA SigVer for Web GUI	Crypto Officer - Operator Password (Alphanumeric string): E User - Operator Password (Alphanumeric string): E
Software load	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 (RSA public key): R

Table 16: Approved Services

4.4 Non-Approved Services

The module does not provide any non-Approved services.

N/A for this module.

4.5 External Software/Firmware Loaded

The module does not support the loading of external software or firmware.

5. Software/Firmware Security

5.1 Integrity Techniques

All software within the cryptographic boundary is verified using an approved integrity technique implemented within the cryptographic module itself. The module implements a 2048-bit RSA digital signature verification with a SHA-512 hash to ensure the integrity of its software components.

The module's pre-operational integrity check is performed automatically at module power-up.

5.2 Initiate on Demand

This integrity check can also be performed on demand by the module operator by performing a reboot.

6. Operational Environment

6.1 Operational Environment Type and Requirements

The NetScaler Virtual Appliance comprises a software cryptographic library that executes in a Modifiable operational environment.

6.2 Configuration Settings and Restrictions

NetScaler Virtual Appliance runs on the FreeBSD v11.4 OS, which acts as the guest OS son top of the virtualization layer. The virtualization layer is provided by VMware's ESXi hypervisor v7.0. The VMware hypervisor runs directly on the 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 and specifies the appropriate storage area in the “Storage” column if applicable to a specific SSP.

Storage Area Name	Description	Persistence Type
On Disk	SSPs are stored on disk	Static
Non-volatile Memory	SSPs are stored in non-volatile memory	Static
Volatile Memory	SSPs are stored in volatile memory	Dynamic

Table 17: 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 input and output methods listed and specifies the appropriate method in the “Inputs/Outputs” column if applicable to a specific SSP.

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
Exported in encrypted form via part of config backup file	On Disk	External	Encrypted	Automated	Electronic	
Exported in plaintext	placeholder	placeholder	Plaintext	Automated	Electronic	
Imported in encrypted form via TLS or SSH session	External	On Disk	Encrypted	Automated	Electronic	
Imported in plaintext via local console	External	On Disk	Encrypted	Automated	Electronic	
Imported in encrypted form via RSA key transport	External	Volatile Memory	Encrypted	Automated	Electronic	Key Transport for TLS

Table 18: 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 zeroization methods listed and specifies the appropriate method in the “Zeroization” column if applicable to a specific SSP.

Zeroization Method	Description	Rationale	Operator Initiation
CLI command	The zeroization method is conducted via CLI command	The CLI command overwrites the storage location keys with 0's, making them irretrievable	Crypto Officer by command

Zeroization Method	Description	Rationale	Operator Initiation
Completion of TLS Session Key and TLS Authentication Key derivation	Zeroization upon the completion of a TLS Session Key and TLS Authentication Key derivation	Keys are automatically zeroized upon completion of TLS key derivation and are irretrievable	Crypto Officer or User by TLS session termination
Reboot	Zeroization when module is rebooted	Keys are procedurally zeroized by rebooting the host platform, which is acceptable at Software level 1	Crypto Officer by rebooting the host system
Remove power	Zeroization when power is removed from the module	Keys are procedurally zeroized by removing the power of the host platform, which is acceptable at Software level 1	Crypto Officer by removing power
Session termination	Zeroization when the session is terminated	Keys are automatically zeroized upon session termination and are irretrievable	Crypto Officer or User by session termination
NetScaler VA Detection	SSPs are not zeroized. NetScaler VA detects any modification.	NetScaler VA detects modification of Public Security Parameters for listed entry.	N/A
Zeroisation of KEK	The KEK is zeroised	Zeroization of the KEK renders SSP permanently unrecoverable	Crypto Officer reboots or removes the power

Table 19: SSP Zeroization Methods

9.4 SSPs

The module supports the keys and other SSPs listed in the table below. Note that all SSP imports and exports are electronic and performed within the Tested OE's Physical Parameter (TOEPP).

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
KEK Fragment 1	Hashed in combination with KEK Fragment 2 to derive KEK	N/A - N/A	Keying material - Neither	CKG (Control Plane)		AES for Disk Encryption
KEK Fragment 2	Hashed in combination with KEK Fragment 1 to derive KEK	N/A - N/A	Keying material - Neither	CKG (Control Plane)		AES for Disk Encryption
KEK (AES Key)	Encryption and decryption of passwords and passphrases	256 bits - 256 bits	Symmetric Key - CSP	CKG (KEK)		AES for Disk Encryption
PEM Key (AES Key)	Encryption and decryption of asymmetric private keys	256 bits - 256 bits	Symmetric Key - CSP		PBKDF	
AES Key	Encryption and decryption	Between 128 and 256 bits - Between 128 and 256 bits	Symmetric Key - CSP	CKG (Data Plane)		AES for AES Key
AES GCM Key	Encryption and decryption	256 bits - 256 bits	Symmetric Key - CSP	CKG (Data Plane)		
HMAC Key	Message authentication with SHS	Between 160 and 512 bits - Between 128 and 256 bits	Authentication - CSP	CKG (Data Plane)		HMAC for HMAC key

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
CA Public Key	TLS certificate authentication	RSA: Between 2048 and 3072 bits, ECDSA: Between 224 and 512 bits - RSA: Between 112 and 128 bits, ECDSA: Between 112 and 256 bits	Public/Private - PSP			
DH Private Key	Generation of SSH, TLS, and IKE shared secrets	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	CKG (Data Plane)		Key Agreement for SSH (DH) Key Agreement for IKE/IPsec (DH) Key Agreement for TLS (DH)
DH Public Key	Generation of SSH, TLS, and IKE shared secrets	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	CKG (Data Plane)		Key Agreement for SSH (DH) Key Agreement for IKE/IPsec (DH) Key Agreement for TLS (DH)
ECDH Private Key	Generation of SSH and TLS shared secrets	Between 224 and 512 bits - Between 112 and 256 bits	Public/Private - CSP	CKG (Control Plane) CKG (Data Plane)		Key Agreement for SSH (ECDH) Key agreement for TLS (ECDH)
ECDH Public Key	Generation of SSH and TLS shared secrets	Between 224 and 512 bits - Between 112 and 256 bits	Public/Private - PSP	CKG (Control Plane) CKG (Data Plane)		Key Agreement for SSH (ECDH) Key agreement for TLS (ECDH)
RSA Private Key	Generation of TLS shared secrets	2048 or 3072 bits - 112 or 128 bits	Public/Private - CSP	CKG (Control Plane)		Key Agreement for TLS (DH) Key agreement for TLS (ECDH)

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
RSA Public Key	Generation of TLS shared secrets	2048 or 3072 bits - 112 or 128 bits	Public/Private - PSP	CKG (Control Plane)		Key Agreement for TLS (DH) Key agreement for TLS (ECDH)
SSH Private Key	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	CKG (Control Plane)		
SSH Public Key	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	CKG (Control Plane)		
SSH Session Key	Encryption and decryption of SSH session packets	Between 128 and 256 bits - Between 128 and 256 bits	Symmetric Key - CSP			AES for SSH
SSH Authentication Key	Authentication of SSH session packets	Between 160 and 512 bits - Between 128 and 256 bits	Authentication - PSP			HMAC for SSH
IKE/IPsec Pre-shared key (PSK)	Authentication during IKE/IPsec session negotiation; Derivation of the IKE/IPsec Session Keys and IKE/IPsec Authentication Keys for IKEv1	- - -	Authentication - CSP			Key Agreement for IKE/IPsec (DH)
IKE/IPsec Session Key (AES key)	Encryption and decryption of IKE/IPsec session packets	Between 128 and 256 bits - Between 128 and 256 bits	Symmetric Key - CSP			AES for IKE/IPsec
IKE/IPsec Authentication Key (HMAC key)	Authentication of IKE/IPsec session packets	Between 160 and 512 bits - Between 128 and 256 bits	Authentication - CSP			HMAC for IKE/IPsec
RDP Session Key	Encryption and decryption of RDP user and target information	256 bits - 256 bits	Symmetric Key - Neither			AES for RDP Session
DFA Session Key	DFA authentication to the module	256 bits - 256 bits	Symmetric Key - Neither			AES for DFA Session Key
TLS Private Key	TLS authentication; SAML authentication (RSA only); OpenID authentication (RSA only)	RSA: Between 2048 and 4096 bits, ECDA: between 224 and 512 bits - RSA: Between 112 and 150 bits, ECDA: between 112 and 256 bits	Public/Private - CSP	CKG (Control Plane)		Key Agreement for TLS (DH) Key agreement for TLS (ECDH)

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
TLS Public Key	TLS authentication;	RSA: Between 2048 and 4096 bits, ECDA: between 224 and 512 bits - RSA: Between 112 and 150 bits, ECDA: between 112 and 256 bits	Public/Private - PSP		Key Agreement for TLS (DH) Key agreement for TLS (ECDH)	
TLS Session Key	Encryption and decryption of TLS session packets	128 or 256 bits - 128 or 256 bits	Symmetric Key - Neither			AES for TLS session AES GCM for TLS Session
TLS Authentication Key (HMAC Key)	Authentication of TLS session packets	Between 160 and 384 bits - Between 128 and 256 bits	Authentication - CSP			HMAC for TLS Session
TLS Ticket Encryption Key (AES key)	Encryption and decryption of TLS session tickets	128 bits - 128 bits	Symmetric Key - CSP	CKG (Data Plane)		AES for TLS session
TLS Ticket Authentication Key (HMAC key)	Computes the digest of TLS session tickets	256 bits - 256 bits	Authentication - CSP	CKG (Control Plane) CKG (Data Plane)		HMAC for TLS Ticket
SNMPv3 Privacy Key (AES key)	Encryption and decryption of SNMPv3 packets	128 bits - 128 bits	Symmetric Key - CSP			AES for SNMPv3
SNMPv3 Authentication Key (HMAC key)	Authentication of SNMPv3 packets	160 bits - 128 bits	Authentication - CSP			HMAC for SNMPv3
Public DNS KSK (RSA public key)	Public DNS ZSK authentication	Between 2048 and 4096 bits - Between 112 and 150 bits	Public/Private - PSP			
Private DNS KSK (RSA private key)	Public DNS ZSK authentication	Between 2048 and 4096 bits - Between 112 and 150 bits	Public/Private - CSP			RSA SigGen for DNSsec
Public DNS ZSK (RSA public key)	DNS zone authentication	Between 2048 and 4096 bits - Between 112 and 150 bits	Public/Private - PSP			RSA SigVer for DNSsec
Private DNS ZSK (RSA private key)	DNS zone signature generation	Between 2048 and 4096 bits - Between 112 and 150 bits	Public/Private - CSP			RSA SigGen for DNSsec
PEM Passphrase (Alphanumeric string)	Derivation of PEM Key	- - -	Alphanumeric String - CSP			AES (PEM Key) for Encrypting TLS Private Key
AES GCM IV (96 and 128-bit IV)	IV for AES GCM	- - -	Initialization Vector - CSP			
SSH Shared Secret	Derivation of the SSH Session Key and SSH Authentication Key	- - -	Shared Secret - CSP		Key Agreement for SSH (DH) Key Agreement for SSH (ECDH)	

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
IKE/IPsec Shared Secret	Derivation of the IKE/IPsec Session Keys and IKE/IPsec Authentication Keys	- - -	Shared Secret - CSP		Key Agreement for IKE/IPsec (DH)	
TLS Pre-Master Secret	Derivation of the TLS Extended Master Secret	- - -	Pre-Master Secret - CSP	CKG (Control Plane) CKG (Data Plane)	Key Agreement for TLS (DH) Key agreement for TLS (ECDH)	
TLS Extended Master Secret	Derivation of the TLS Session Key and TLS Authentication Key	- - -	Extended Master Secret - CSP		Key Agreement for TLS (DH) Key agreement for TLS (ECDH)	
Hash DRBG Entropy	Entropy input for Hash DRBG	- - -	Entropy - CSP			
Hash DRBG Seed	Seed material for Hash DRBG	- - -	DRBG Seed - CSP			
Hash DRBG 'V' Value (Internal state value)	Internal state value used with Hash DRBG	- - -	Internal State Value - CSP			
Hash DRBG 'C' Value (Internal state value)	Internal state value used with Hash DRBG	- - -	Internal State Value - CSP			
CTR DRBG Entropy	Entropy input for CTR DRBG	- - -	Entropy - CSP			
CTR DRBG Seed	Seed material for CTR DRBG	- - -	DRBG Seed - CSP			
CTR DRBG 'V' Value	Internal state value used with CTR DRBG	- - -	Internal State Value - CSP			
CTR DRBG 'Key' Value (AES key)	Internal state value used with CTR DRBG	- - -	Internal State Value - CSP			
SNMPv3 Privacy Passphrase (Alphanumeric string)	Derivation of the SNMPv3 Privacy Key	- - -	Alphanumeric String - CSP			AES for SNMPv3
SNMPv3 Authentication Passphrase (Alphanumeric string)	Derivation of the SNMPv3 Authentication Key	- - -	Alphanumeric String - CSP			HMAC for SNMPv3
LDAP Admin Password (Alphanumeric string)	Used to bind to the LDAP server	- - -	Alphanumeric String - CSP			
RDP PSK (Shared secret)	Used as input to derive RDP Session Key	- - -	Shared Secret - CSP			AES for RDP Session
Oauth Client Secret (Shared secret)	Oauth and Oauth IDP authentication to the module	- - -	Shared Secret - CSP			
DFA Shared Secret	Used as input to derive DFA Session Key	- - -	Shared Secret - CSP			AES for DFA Session Key

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
ZebOS Router Password (Alphanumeric string)	Router authentication	- - -	Alphanumeric String - CSP			
Cluster Password (Alphanumeric string)	Used to connect nodes to the cluster coordinator	- - -	Alphanumeric String - CSP			
Operator Password (Alphanumeric string)	Authenticate the operator to the module via an external authentication service	- - -	Alphanumeric String - CSP			
Software Load Integrity Key (RSA public key)	(Not an SSP), Used to verify the new software load	2048 bits - 112 bits	Public/Private - Neither			RSA SigVer for Software Load Integrity

Table 20: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
KEK Fragment 1		Non-volatile Memory:Plaintext		CLI command	KEK Fragment 2:Used With
KEK Fragment 2		Non-volatile Memory:Plaintext		CLI command	KEK Fragment 1:Used With
KEK (AES Key)		Volatile Memory:Plaintext		Reboot Remove power	KEK Fragment 1:Derived From KEK Fragment 2:Derived From
PEM Key (AES Key)		On Disk:Encrypted		CLI command	KEK (AES Key):Derived From
AES Key	Exported in encrypted form via part of config backup file Imported in encrypted form via TLS or SSH session Imported in plaintext via local console	On Disk:Encrypted		Reboot Remove power	KEK (AES Key):Encrypts
AES GCM Key		Volatile Memory:Plaintext	Until module reboot or power off	Reboot Remove power	
HMAC Key	Exported in encrypted form via part of config backup file Imported in encrypted form via TLS or SSH session Imported in plaintext via local console	On Disk:Encrypted		Zeroisation of KEK	KEK (AES Key):Encrypts

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
CA Public Key	Exported in plaintext Imported in encrypted form via TLS or SSH session Imported in plaintext via local console	On Disk:Plaintext		NetScaler VA Detection	
DH Private Key		Volatile Memory:Plaintext	Until module reboot, power off, or session termination	Reboot Remove power Session termination	DH Public Key:Paired With
DH Public Key		Volatile Memory:Plaintext	Until module reboot, power off, or session termination	Reboot Remove power Session termination	DH Private Key:Paired With
ECDH Private Key		Volatile Memory:Plaintext	Until module reboot, power off, or session termination	Reboot Remove power Session termination	ECDH Public Key :Paired With
ECDH Public Key	Exported in plaintext Imported in plaintext via local console	Volatile Memory:Plaintext	Until module reboot, power off, or session termination	Reboot Remove power Session termination	ECDH Private Key :Paired With
RSA Private Key	Exported in plaintext Imported in plaintext via local console	Volatile Memory:Plaintext		Zeroisation of KEK	RSA Public Key:Paired With KEK (AES Key):Encrypts
RSA Public Key	Exported in plaintext Imported in plaintext via local console	Volatile Memory:Plaintext		NetScaler VA Detection	RSA Private Key :Paired With
SSH Private Key	Exported in encrypted form via part of config backup file	On Disk:Plaintext		CLI command	
SSH Public Key	Exported in encrypted form via part of config backup file	Volatile Memory:Plaintext		NetScaler VA Detection	
SSH Session Key		Volatile Memory:Plaintext	Until module reboot, power off, or session termination	Reboot Remove power Session termination	
SSH Authentication Key		Volatile Memory:Plaintext	Until module reboot, power off, or session termination	Reboot Remove power Session termination	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
IKE/IPsec Pre-shared key (PSK)	Exported in encrypted form via part of config backup file Imported in plaintext via local console	Volatile Memory:Plaintext	Until module reboot, power off, or session termination	Reboot Remove power Session termination	
IKE/IPsec Session Key (AES key)		Volatile Memory:Plaintext	Until module reboot, power off, or session termination	Reboot Remove power Session termination	
IKE/IPsec Authentication Key (HMAC key)		Volatile Memory:Plaintext	Until module reboot, power off, or session termination	Reboot Remove power Session termination	
RDP Session Key		Volatile Memory:Plaintext	Until module reboot, power off, or session termination	Reboot Remove power Session termination	
DFA Session Key		Volatile Memory:Plaintext		Reboot Remove power Session termination	
TLS Private Key	Exported in encrypted form via part of config backup file Imported in encrypted form via TLS or SSH session Imported in plaintext via local console	On Disk:Encrypted		Zeroisation of KEK	TLS Public Key:Paired With
TLS Public Key		Volatile Memory:Plaintext		NetScaler VA Detection	TLS Private Key:Paired With
TLS Session Key		Volatile Memory:Plaintext	Until module reboot, power off, or session termination	Reboot Remove power Session termination	
TLS Authentication Key (HMAC Key)		Volatile Memory:Plaintext	Until module reboot, power off, or session termination	Reboot Remove power Session termination	
TLS Ticket Encryption Key (AES key)	Imported in encrypted form via TLS or SSH session	Volatile Memory:Plaintext	Until module reboot, power off, or session termination	Reboot Remove power Session termination	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
TLS Ticket Authentication Key (HMAC key)	Imported in encrypted form via TLS or SSH session	Volatile Memory:Plaintext	Until module reboot, power off, or session termination	Reboot Remove power Session termination	
SNMPv3 Privacy Key (AES key)		Volatile Memory:Plaintext	Until module reboot, power off, or session termination	Reboot Remove power Session termination	
SNMPv3 Authentication Key (HMAC key)		Volatile Memory:Plaintext	Until module reboot or power off	Reboot Remove power	
Public DNS KSK (RSA public key)	Exported in plaintext Imported in encrypted form via TLS or SSH session	On Disk:Plaintext		Zeroisation of KEK	Private DNS KSK (RSA private key):Paired With
Private DNS KSK (RSA private key)	Exported in encrypted form via part of config backup file Imported in encrypted form via TLS or SSH session	On Disk:Encrypted		NetScaler VA Detection	PEM Key (AES Key):Encrypts Public DNS KSK (RSA public key):Paired With
Public DNS ZSK (RSA public key)	Exported in plaintext Imported in encrypted form via TLS or SSH session	On Disk:Plaintext		NetScaler VA Detection	Private DNS ZSK (RSA private key):Paired With
Private DNS ZSK (RSA private key)	Exported in encrypted form via part of config backup file Imported in encrypted form via TLS or SSH session	On Disk:Encrypted		Zeroisation of KEK	Public DNS ZSK (RSA public key):Paired With
PEM Passphrase (Alphanumeric string)	Exported in encrypted form via part of config backup file Imported in plaintext via local console	Non-volatile Memory:Plaintext On Disk:Encrypted	Until module reboot or power off	Reboot Remove power	KEK (AES Key):Encrypts
AES GCM IV (96 and 128-bit IV)		Volatile Memory:Plaintext	Until module reboot or power off	Reboot Remove power	
SSH Shared Secret		Volatile Memory:Plaintext	Until module reboot, power off, or session termination	Reboot Remove power Session termination	
IKE/IPsec Shared Secret		Volatile Memory:Plaintext	Until module reboot, power off, or session termination	Reboot Remove power Session termination	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
TLS Pre-Master Secret	Imported in encrypted form via RSA key transport	Volatile Memory:Plaintext	Until module reboot, power off, or session termination	Reboot Remove power Session termination	
TLS Extended Master Secret		Volatile Memory:Plaintext	Until module reboot, power off, or session termination	Reboot Remove power Session termination	TLS Pre-Master Secret:Derived From
Hash DRBG Entropy		Volatile Memory:Plaintext	Until module reboot or power off	Reboot Remove power	
Hash DRBG Seed		Volatile Memory:Plaintext	Until module reboot or power off	Reboot Remove power	
Hash DRBG 'V' Value (Internal state value)		Volatile Memory:Plaintext	Until module reboot or power off	Reboot Remove power	
Hash DRBG 'C' Value (Internal state value)		Volatile Memory:Plaintext	Until module reboot or power off	Reboot Remove power	
CTR DRBG Entropy		Volatile Memory:Plaintext	Until module reboot or power off	Reboot Remove power	
CTR DRBG Seed		Volatile Memory:Plaintext	Until module reboot or power off	Reboot Remove power	
CTR DRBG 'V' Value		Volatile Memory:Plaintext	Until module reboot or power off	Reboot Remove power	
CTR DRBG 'Key' Value (AES key)		Volatile Memory:Plaintext	Until module reboot or power off	Reboot Remove power	
SNMPv3 Privacy Passphrase (Alphanumeric string)	Exported in encrypted form via part of config backup file Imported in encrypted form via TLS or SSH session Imported in plaintext via local console	On Disk:Encrypted		Zeroisation of KEK	KEK (AES Key):Encrypts
SNMPv3 Authentication Passphrase (Alphanumeric string)	Exported in encrypted form via part of config backup file Imported in encrypted form via TLS or SSH session Imported in plaintext via local console	On Disk:Encrypted		Zeroisation of KEK	KEK (AES Key):Encrypts

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
LDAP Admin Password (Alphanumeric string)	Exported in encrypted form via part of config backup file Imported in encrypted form via TLS or SSH session Imported in plaintext via local console	On Disk:Encrypted		Zeroisation of KEK	KEK (AES Key):Encrypts
RDP PSK (Shared secret)	Exported in encrypted form via part of config backup file Imported in encrypted form via TLS or SSH session Imported in plaintext via local console	On Disk:Encrypted		Zeroisation of KEK	KEK (AES Key):Encrypts
Oauth Client Secret (Shared secret)	Exported in encrypted form via part of config backup file Imported in encrypted form via TLS or SSH session Imported in plaintext via local console	On Disk:Encrypted		Zeroisation of KEK	KEK (AES Key):Encrypts
DFA Shared Secret	Exported in encrypted form via part of config backup file Imported in encrypted form via TLS or SSH session Imported in plaintext via local console	On Disk:Encrypted		Zeroisation of KEK	KEK (AES Key):Encrypts
ZebOS Router Password (Alphanumeric string)	Exported in encrypted form via part of config backup file Imported in encrypted form via TLS or SSH session Imported in plaintext via local console	On Disk:Encrypted		Zeroisation of KEK	KEK (AES Key):Encrypts

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
Cluster Password (Alphanumeric string)	Exported in encrypted form via part of config backup file Imported in encrypted form via TLS or SSH session Imported in plaintext via local console	On Disk:Encrypted		Zeroisation of KEK	KEK (AES Key):Encrypts
Operator Password (Alphanumeric string)	Exported in encrypted form via part of config backup file Imported in encrypted form via TLS or SSH session Imported in plaintext via local console	On Disk:Encrypted		Zeroisation of KEK	KEK (AES Key):Encrypts
Software Load Integrity Key (RSA public key)	Imported in plaintext via local console	Non-volatile Memory:Plaintext	Until module reboot or power off	Reboot Remove power	

Table 21: SSP Table 2

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 module performs the following pre-operational self-test(s):

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 22: Pre-Operational Self-Tests

10.2 Conditional Self-Tests

The module performs the following conditional self-tests:

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-CBC (A3942)	128-bit	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	Encrypt, Decrypt	After successful completion of software integrity test
AES-GCM (A3942)	256-bit	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	Encrypt, Decrypt	After successful completion of software integrity test.
Counter DRBG (A3942)	-	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	Instantiate/Generate/Reseed	After successful completion of software integrity test.
KAS-FFC-SSC Sp800-56Ar3 (A3942)	-	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	Primitive "Z" computation test	After successful completion of software integrity test.
KAS-ECC-SSC Sp800-56Ar3 (A3942)	-	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	Primitive "Z" computation test	After successful completion of software integrity test.
ECDSA SigGen (FIPS186-4) (A3942)	P-256	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	Sign	After successful completion of software integrity test.

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
ECDSA SigVer (FIPS186-4) (A3942)	P-256	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	Verify	After successful completion of software integrity test.
HMAC-SHA-1 (A3942)	SHA-1	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	N/A	After successful completion of software integrity test.
HMAC-SHA2-256 (A3942)	SHA2-256	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	N/A	After successful completion of software integrity test.
HMAC-SHA2-512 (A3942)	SHA2-512	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	N/A	After successful completion of software integrity test.
PBKDF (A3942)	SHA-1	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	N/A	After successful completion of software integrity test.
RSA SigGen (FIPS186-4) (A3942)	2048-bit, SHA2-256	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	Sign	Before software integrity test
RSA SigVer (FIPS186-4) (A3942)	2048-bit, SHA2-256	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	Verify	Before software integrity test
SHA-1 (A3942)	-	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	N/A	After successful completion of software integrity test.
SHA2-256 (A3942)	-	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	N/A	After successful completion of software integrity test.
SHA2-512 (A3942)	-	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	N/A	After successful completion of software integrity test.
KDF IKEv1 (A3942)	-	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	N/A	After successful completion of software integrity test.
KDF IKEv2 (A3942)	-	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	N/A	After successful completion of software integrity test.
KDF SSH (A3942)	-	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	N/A	After successful completion of software integrity test.
KDF TLS (A3942)	-	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	N/A	After successful completion of software integrity test.

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
TLS v1.2 KDF RFC7627 (A3942)	-	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	N/A	After successful completion of software integrity test.
AES-CBC (A3943)	128-bit	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	Encrypt, Decrypt	After successful completion of software integrity test.
AES-GCM (A3943)	256-bit	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	Encrypt, Decrypt	After successful completion of software integrity test.
KAS-ECC-SSC Sp800-56Ar3 (A3943)	-	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	N/A	After successful completion of software integrity test.
ECDSA SigGen (FIPS186-4) (A3943)	P-256	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	Sign	After successful completion of software integrity test.
ECDSA SigVer (FIPS186-4) (A3943)	P-256	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	Verify	After successful completion of software integrity test.
Hash DRBG (A3943)	AES, 256-bit, with derivation function	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	Instantiate/Generate/Reseed	After successful completion of software integrity test.
HMAC-SHA-1 (A3943)	-	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	N/A	After successful completion of software integrity test.
HMAC-SHA2-256 (A3943)	-	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	N/A	After successful completion of software integrity test.
HMAC-SHA2-512 (A3943)	-	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	N/A	After successful completion of software integrity test.
KDF SP800-108 (A3943)	-	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	N/A	After successful completion of software integrity test.
RSA SigGen (FIPS186-4) (A3943)	2048-bit, SHA2-256	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	N/A	After successful completion of software integrity test.
RSA SigVer (FIPS186-4) (A3943)	2048-bit, SHA2-256	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	N/A	After successful completion of software integrity test.
SHA-1 (A3943)	-	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	N/A	After successful completion of software integrity test.

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
SHA2-256 (A3943)	-	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	N/A	After successful completion of software integrity test.
SHA2-512 (A3943)	-	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	N/A	After successful completion of software integrity test.
KDF TLS (A3943)	-	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	N/A	After successful completion of software integrity test.
TLS v1.2 KDF RFC7627 (A3943)	-	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	N/A	After successful completion of software integrity test.
TLS v1.3 KDF (A3943)	-	KAT	CAST	"POST FAILED" message in /var/log/FIPS-post.log	N/A	After successful completion of software integrity test.

Table 23: Conditional Self-Tests

10.3 Periodic Self-Test Information

The module has conditions that may interrupt module operations during the time to repeat the periodic self-tests. The table below specifies the period and the policy for these conditions.

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

Table 24: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-CBC (A3942)	KAT	CAST	On Demand	Manually
AES-GCM (A3942)	KAT	CAST	On Demand	Manually
Counter DRBG (A3942)	KAT	CAST	On Demand	Manually
KAS-FFC-SSC Sp800-56Ar3 (A3942)	KAT	CAST	On Demand	Manually
KAS-ECC-SSC Sp800-56Ar3 (A3942)	KAT	CAST	On Demand	Manually
ECDSA SigGen (FIPS186-4) (A3942)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-4) (A3942)	KAT	CAST	On Demand	Manually
HMAC-SHA-1 (A3942)	KAT	CAST	On Demand	Manually
HMAC-SHA2-256 (A3942)	KAT	CAST	On Demand	Manually
HMAC-SHA2-512 (A3942)	KAT	CAST	On Demand	Manually
PBKDF (A3942)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-4) (A3942)	KAT	CAST	On Demand	Manually

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
RSA SigVer (FIPS186-4) (A3942)	KAT	CAST	On Demand	Manually
SHA-1 (A3942)	KAT	CAST	On Demand	Manually
SHA2-256 (A3942)	KAT	CAST	On Demand	Manually
SHA2-512 (A3942)	KAT	CAST	On Demand	Manually
KDF IKEv1 (A3942)	KAT	CAST	On Demand	Manually
KDF IKEv2 (A3942)	KAT	CAST	On Demand	Manually
KDF SSH (A3942)	KAT	CAST	On Demand	Manually
KDF TLS (A3942)	KAT	CAST	On Demand	Manually
TLS v1.2 KDF RFC7627 (A3942)	KAT	CAST	On Demand	Manually
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 (FIPS186-4) (A3943)	KAT	CAST	On Demand	Manually
ECDSA SigVer (FIPS186-4) (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
KDF SP800-108 (A3943)	KAT	CAST	On Demand	Manually
RSA SigGen (FIPS186-4) (A3943)	KAT	CAST	On Demand	Manually
RSA SigVer (FIPS186-4) (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

Table 25: Conditional Periodic Information

10.4 Error States

If the module enters the critical error state due to a failure of the pre-operational integrity test, the module enters a critical error state and logs an error message. In this state, the boot sequence and entire system is halted. The only action available from this state is to reboot the module to trigger the re-execution of the integrity test. The error condition is considered to have been cleared if the module successfully passes the pre-operational integrity test. If the module continues to return to a halted state, the module is considered to be malfunctioning or compromised, and Cloud Software Group Customer Support must be contacted.

If the module enters the critical error state due to a failure of any of the conditional CASTs, cryptographic operations are halted, and the module inhibits all data output from the module. The module logs an error message and automatically reboots to clear the error state. The CO must contact Cloud Software Group if this error occurs.

The successful completion or failure of the pre-operational self-tests and conditional CASTs can be verified by checking the log files.

- NetScaler Control Plane Cryptographic Library – Successful completion of the self-tests is indicated by "POST Success" in `/var/log/FIPS-post.log`. Failure is indicated by "POST Failed" in `/var/log/FIPS-post.log` (both messages indicate a critical error state).

- NetScaler Data Plane Cryptographic Library – Successful completion of the self-tests is indicated by "FIPS POST Successful" in `/var/log/ns.log`. Failure is indicated by "FIPS Post Failed" in `/var/log/ns.log` (both messages indicate a critical error state).

If any of the remaining conditional self-tests fail, the module goes through a soft error state and the following message is displayed:

```
"Internal failure in SSL cert/key generation tool"
```

For these failures, the module returns to an operational state once the message is displayed (and the error is logged). The user may retry the service (which calls the conditional self-test again) or move to other operations. Successful completion of the conditional self-test is indicated by the absence of an error message.

The table below describes the error states the status indicators of the module.

Name	Description	Conditions	Recovery Method	Indicator
Critical Error (pre-operational self tests)	The booth sequence and entire system is halted. The only action available from this state is to reboot the module to trigger the re-execution of the integrity test.	Module fails pre-operational integrity test.	The module successfully passes the pre-operational integrity test. If the module continues to return to a halted state, the module is considered to be malfunctioning or compromised, and Cloud Software Group Customer Support must be contacted.	Logs "POST Failed" error message in /var/log/FIPS-post.log for Netscaler Control Plane Cryptographic Library. Logs "FIPS Post Failed" in /var/log/ns.log for Netscaler Data Plane Cryptographic Library.
Critical Error (Conditional CASTs)	Cryptographic operations are halted, and the module inhibits all data output from the module.	Module fails any conditional CASTs.	The module automatically reboots after logging an error message to clear the error state. The CO must contact Cloud Software Group Support if this error occurs.	Logs "POST Failed" error message in /var/log/FIPS-post.log for Netscaler Control Plane Cryptographic Library. Logs "FIPS Post Failed" in /var/log/ns.log for Netscaler Data Plane Cryptographic Library.
Soft Error	Error that may occur when invoking service that calls the conditional self-test.	Module fails any of the remaining conditional self-tests.	The module returns to an operational state once the message is displayed, and the error is logged. The user may retry the service or move to other operations.	The following message is displayed: "Internal failure in SSL cert/key generation tool".

Table 26: Error States

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 Virtual Appliance, the installation files can be downloaded from Citrix ADC 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 Citrix Support Portal using valid credentials.

The CO is responsible for all initial setup activities, including configuring the virtual machine and installing/configuring the NetScaler Virtual Appliance. Prior to the installation, the CO should read the document entries within the Getting Started with Citrix ADC webpage on Cloud Software Group's online product documentation portal.

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

11.1.1 Installation

For detailed guidance regarding the installation of NetScaler VA, please see the Deploy a Citrix ADC VPX instance webpage on Citrix's online product documentation portal and refer to the following document entries:

- Support matrix and usage guidelines
- Install a Citrix ADC VPX instance on VMware ESX

The above document entries include the NetScaler Virtual Appliance support matrix and usage guidelines, prerequisites for installing the NetScaler Virtual Appliance, hardware requirements for the host platforms, and NetScaler Virtual Appliance installation instructions. To install the required license files, the CO must follow the instructions on the Citrix ADC licensing overview webpage on Cloud Software Group's online product documentation portal.

11.1.2 Initialization

After the appliance has been setup, the CO is responsible for the general configuration of the module. The Web GUI or CLI can be used for the general configuration of the module. All general configuration must be complete before performing configuration necessary to place the module in a FIPS-Approved mode of operation.

The general configuration requirements and instructions are described in the "Quick Start Installation and Configuration" section of the Citrix ADC Deployment Guide found on Citrix's online product documentation portal.

11.1.3 General Configuration

After the NetScaler Virtual Appliance has been installed on a VMware ESXi 7.0 hypervisor, the CO is responsible for the general configuration of the module. The Web GUI (configuration utility) or CLI can be used for the general configuration of the module. All general configuration steps must be complete before performing configuration necessary to place the module in a FIPS-Approved mode of operation.

The general configuration requirements and instructions are described in the "Quick Start Installation and Configuration" section of the Citrix NetScaler Deployment Guide found on Cloud Software Group's online product documentation portal.

11.1.4 FIPS-Approved Mode Configuration and Status

The CO is responsible for the security-relevant configuration of the module. To initialize the module for the approved mode of operation, the CO must:

- Configure the passphrase requirements
- Replace the default TLS certificate
- Disable HTTP access to the Web GUI
- Enable external authentication
- Disable local authentication

To accomplish these tasks, the CO must follow the procedures detailed in the sections below (for more information, please see the "Configuration Guidelines" section of the document entry Citrix ADC Deployment Guide.

Once the CO has completed all configuration steps, the CO shall review all saved settings to ensure they match the required settings as documented in the configuration guidance below. If properly set, the module is considered to be operating in the FIPS-Approved mode.

11.1.4.1 Configure the Passphrase Requirements

Passphrases are used to derive keys using PBKDF. The CO must configure strong passphrase requirements. 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**.

11.1.4.2 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 must replace the default certificate with a newly-generated certificate after the initial installation.

To replace the default TLS certificate, the CO must 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 Citrix's online product documentation portal.

11.1.4.3 Disable HTTP Access to the Web GUI

The CO protects traffic to the administrative interface and Web GUI, by configuring the module to use HTTPS[18]. Once the module has been configured to use new TLS and SSH certificates, disable HTTP access to the GUI management interface with the following CLI command: `set ns ip <NSIP> -gui SECUREONLY`

11.1.4.4 Enable External Authentication

Once the module is configured in FIPS-Approved mode and the `nsroot` account is disabled, then external authentication must be configured. Follow the instructions on the Citrix ADC 13.1 – Configuring external user authentication webpage found on the Cloud Software Group online product documentation portal to configure external system authentication.

The CO must ensure the following before enabling external authentication:

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

[18] HTTPS – Hypertext Transfer Protocol Secure

11.1.4.5 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 must ensure that superuser privileges are not assigned to any user account. To disable local system authentication and enable external system authentication, the CO must run the following CLI command:

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

11.1.5 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 FIPS-Approved mode. Please refer to this section for guidance that the Crypto Officer must follow to ensure that the module is operating in a FIPS-Approved manner.

11.2.1 On-Demand Self-Tests

Although pre-operational self-tests are performed automatically during module power up, they can also be manually launched on demand. Self-tests can be executed by:

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

11.2.2 Zeroization

There are many CSPs within the module's cryptographic boundary including symmetric keys, private keys, public keys, and passphrases. CSPs reside in multiple storage media including the RAM and system memory of the host platform. All ephemeral keys are zeroized on module reboot, power removal, or session termination.

KEK fragments 1 and 2 are stored as plaintext in non-volatile memory. Zeroizing the KEK fragments render all passphrases and passwords stored in the non-volatile memory unrecoverable, effectively zeroizing them. The KEK fragments are zeroized via the following CLI command:

```
rm system csps -type KEK
```

SSH private keys are stored as plaintext in non-volatile memory. SSH private keys are zeroized via the following CLI command:

```
rm system csps -type SSH_HOST_KEYS
```

The output (indicator) of both zeroization commands above is successful return from the command line without any error showing on the console. If the commands fails, an error will show on the console before returning control to the user.

After the module's integrity test is complete the software clears out all values when the signature verification operation is complete (zeroizes temporary values used in the integrity test).

11.2.3 Status and Versioning Information

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

An operator can obtain the module's operational status by reviewing the configuration settings. If set per the guidance documented in section 11.1.4 above, this indicates that the module is operating in the FIPS-Approved mode.

An operator can view the versioning information by:

- using the following CLI commands:

`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[19] 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**

This 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.

If any irregular activity is noticed or the module is consistently reporting errors, then Cloud Software Group Customer Support should be contacted.

[19] ID – Identifier

11.2.4 Additional Administrator Policies and Guidance

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

- All private keys (except for SSH private keys) must be stored as PEM files in encrypted format using one of the FIPS-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 800-52 Rev. 2* recommended cipher suites for TLS connections. If modified, the CO must ensure that only FIPS-Approved cipher suites are configured while in the FIPS-Approved mode. It is recommended to use the list of approved TLS cipher suites in section 3.3 of *NIST SP 800-52 Rev. 2* as guidance.

- The module must be configured to use PSK-based authentication for IPsec connections. The CO must 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 FIPS-Approved mode of operation.

- Kerberos traffic management/SSO shall not be configured or used in the FIPS-Approved mode of operation.

- The module supports clustering, and it may act as either the cluster coordinator or the cluster node. Once appliances are clustered together, all configuration is done on the cluster coordinator and pushed to nodes within the cluster. For details on configuring clusters, refer to Citrix ADC 13.1 – Clustering.

- The CO must ensure that the “Key” and “AutoKey” authentication parameters are not set when adding NTP servers via the GUI, CLI, or API.

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

- 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 must ensure that all certificates are signed using a trusted CA and not by a self-signed certificate.

- When performing a software update the following steps must be performed:

    - Load software load integrity key
    - Load software package
    - Zeroize SSH keys using `rm system csps -type SSH_HOST_KEYS`
    - Run the installation script

- The operator shall not enable the LOM port via ADC configuration.

- The operator shall perform the zeroization commands as specified in section 11.2.2 prior to invoking the “Software Load” service.

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 to select 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.

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

Table 27. 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
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