Citrix Systems, Inc.

Citrix ADC VPX

Software Version: 12.1.55.180

FIPS 140-2 Non-Proprietary Security Policy

FIPS Security Level: 1
Document Version: 0.6

Prepared for:

Citrix Systems, Inc.
851 Cypress Creek Road
Fort Lauderdale, FL 33309
United States of America

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

Prepared by:

Corsec Security, Inc.
13921 Park Center Road, Suite 460
Herndon, VA 20171
United States of America

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

Table of Contents

List of Tables

List of Figures

1. Introduction

1.1 Purpose

This is a non-proprietary Cryptographic Module Security Policy for the Citrix ADC VPX from Citrix Systems, Inc. (hereafter referred to as Citrix). This Security Policy describes how the Citrix ADC VPX meets the security requirements of Federal Information Processing Standards (FIPS) Publication 140-2, which details the U.S.[1] and Canadian government requirements for cryptographic modules. More information about the FIPS 140-2 standard and validation program is available on the Cryptographic Module Validation Program (CMVP) website, which is maintained by the National Institute of Standards and Technology (NIST) and the Canadian Centre for Cyber Security (CCCS).

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 140-2 validation of the module. The Citrix ADC VPX is referred to in this document as "VPX" or "the module".

1.2 References

This document deals only with operations and capabilities of the module in the technical terms of a FIPS 140-2 cryptographic module security policy. More information is available on the module from the following sources:

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

1.3 Document Organization

The Security Policy document is organized into two (2) primary sections. Section 2 provides an overview of the validated module. This includes a general description of the capabilities and the use of cryptography, as well as a presentation of the validation level achieved in each applicable functional area of the FIPS standard. It also provides high-level descriptions of how the module meets FIPS requirements in each functional area. Section 0 documents the guidance needed for the secure use of the module, including initial setup instructions and management methods and policies.

This Security Policy and the other validation submission documentation were produced by Corsec Security, Inc. under contract to Citrix. With the exception of this Non-Proprietary Security Policy, the FIPS 140-2 Submission Package is proprietary to Citrix and is releasable only under appropriate non-disclosure agreements. For access to these documents, please contact Citrix.

[1] U.S. – United States

2. Citrix ADC VPX

2.1 Overview

The Citrix ADC VPX is a virtual application delivery controller (ADC) that accelerates application performance, enhances application availability with advanced L4-L7[2] 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.

VPX provides the web-based GUI[3], REST[4]ful Nitro API[5], and CLI[6] 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.

VPX 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[7] 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 VPX 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 VPX 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[8] or TCP[9] request, and on the basis of L4–L7 header information such as URL[10], 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 – VPX security and protection features protect web applications from Application Layer attacks. The VPX 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[11] injection attempts, cross-site scripting attacks, and more. In addition, the firewall provides identity theft protection by securing confidential corporate information and sensitive customer data.

[2] L4-L7 – Layer 4 through Layer 7
[3] GUI – Graphical User Interface
[4] REST – Representational State Transfer
[5] API – Application Programming Interface
[6] CLI – Command Line Interface
[7] IP – Internet Protocol
[8] HTTP – Hypertext Transfer Protocol
[9] TCP – Transmission Control Protocol
[10] URL – Universal Resource Locator
[11] SQL – Structured Query Language

- Optimization features – Optimization features offload resource-intensive operations, such as SSL[12] 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 VPX supports several transparent TCP optimizations, which mitigate problems caused by high latency and congested network links, accelerating the delivery of applications while requiring no configuration changes to clients or servers.

The VPX is validated at the FIPS 140-2 Section levels shown in Table 1.

Table 1 – Security Level per FIPS 140-2 Section

Section	*Section Title*	*Level*
1	Cryptographic Module Specification	1
2	Cryptographic Module Ports and Interfaces	1
3	Roles, Services, and Authentication	1
4	Finite State Model	1
5	Physical Security	N/A[13]
6	Operational Environment	1
7	Cryptographic Key Management	1
8	EMI/EMC[14]	1
9	Self-tests	1
10	Design Assurance	1
11	Mitigation of Other Attacks	1

2.2 Module Specification

The VPX is a software module with a multiple-chip standalone embodiment. The overall security level of the module is 1. It executes as a virtual appliance and was tested and found complaint on the platform(s) listed in Table 2.

Table 2 – Tested Platforms

Server	CPU[15]	Hypervisor	Guest OS
Dell PowerEdge R630	Intel Xeon E5-2680 v4 2.4 GHz[16]	VMware ESXi 6.5 U2[17]	FreeBSD v8.4

[12] SSL – Secure Sockets Layer
[13] N/A – Not Applicable
[14] EMI/EMC – Electromagnetic Interference / Electromagnetic Compatibility
[15] CPU – Central Processing Unit
[16] GHz – Gigahertz
[17] U2 – Update 2

The vendor affirms the module's continued validation compliance when operating on the platforms listed in Table 3.

Table 3 – Vendor-Affirmed Platforms

Server	CPU[18]	Hypervisor	Guest OS
Dell PowerEdge R630	Intel Xeon E5-2697 v4	Citrix XenServer 7.1 LTSR[19]	FreeBSD v8.4
Dell PowerEdge R630	Intel Xeon E5-2697 v4	Microsoft Hyper-V	FreeBSD v8.4
Dell PowerEdge R630	Intel Xeon E5-2697 v4	KVM w/ Ubuntu 16.04.05 LTS[20]	FreeBSD v8.4
Dell PowerEdge R640	Intel Xeon Platinum 8276	Microsoft Hyper-V	FreeBSD v8.4
Dell PowerEdge R640	Intel Xeon Platinum 8276	KVM w/ Ubuntu 16.04.05 LTS[21]	FreeBSD v8.4
Lenovo SR630	Intel Xeon Silver 4114	Citrix XenServer 8.2 LTSR	FreeBSD v8.4

Per FIPS 140-2 G.5, the cryptographic module maintains validation compliance when operating on any compatible GPC, provided that the GPC uses a single-user operating system/mode specified on the validation certificate, or another compatible single-user operating system. Note that such a GPC may be deployed in a supported cloud environment (Amazon Web Services, Google Cloud Platform, or Microsoft Azure).

The CMVP makes no statement as to the correct operation of the module or the security strengths of the generated keys when ported to an operational environment not listed on the validation certificate.

2.2.1 Physical Cryptographic Boundary

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 1 below 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.

[18] CPU – Central Processing Unit
[19] LTSR – Long Term Service Release
[20] LTS – Long Term Support
[21] LTS – Long Term Support

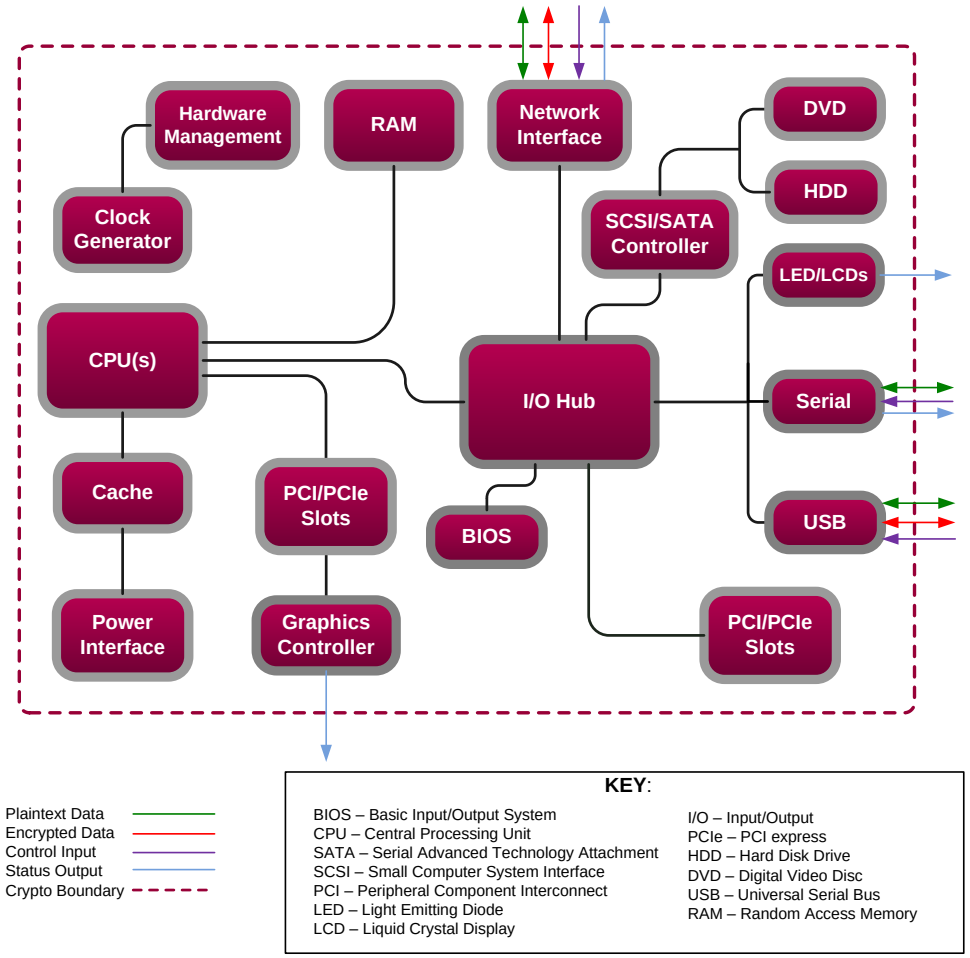

Figure 1 – Block Diagram of the Host Server

2.2.2 Logical Cryptographic Boundary

The logical cryptographic boundary of the module (shown by the red dotted line in Figure 2 below) consists of the VPX virtual appliance software and FreeBSD operating system acting as the guest OS.

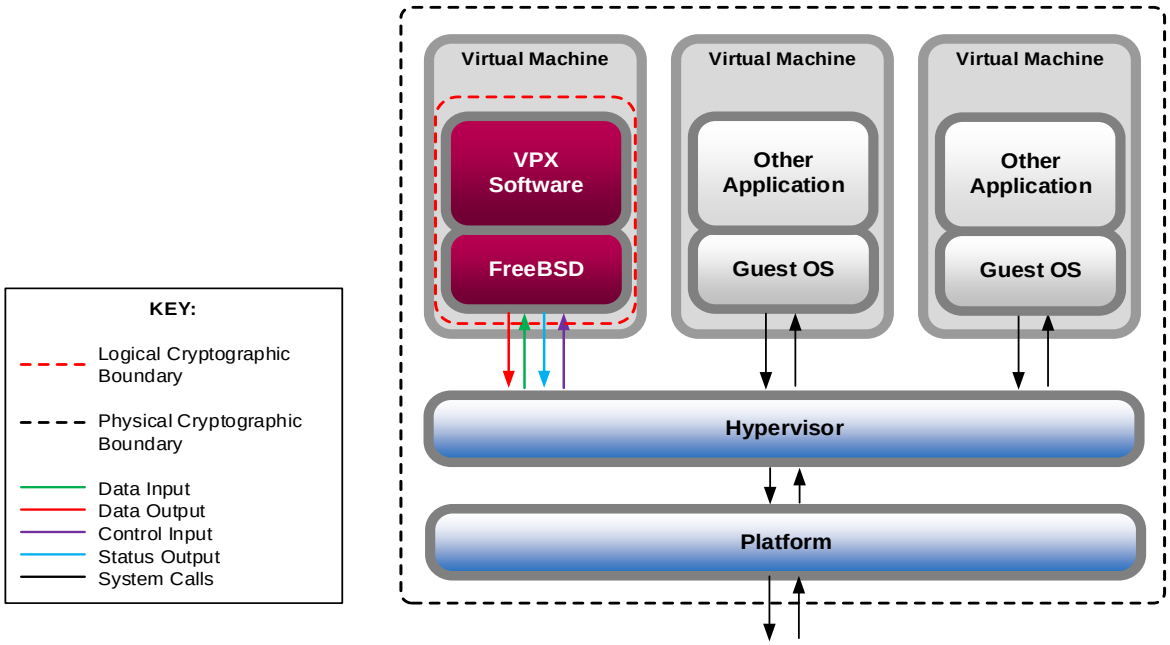

Figure 2 – VPX Logical Cryptographic Boundary

2.2.3 Algorithm Implementations

The module includes two software cryptographic libraries that provide basic cryptographic functionalities and support secure networking protocols. The software libraries for the module are:

- Citrix ADC CP[22] Cryptographic Library v3 (based on OpenSSL FOM[23])
- Citrix ADC DP[24] Cryptographic Library v3 (modified OpenSSL library developed by Citrix)

Table 4 below lists the FIPS-Approved algorithms implemented by the Citrix ADC CP Cryptographic Library v3.

Table 4 – Algorithm Certificate Numbers (Citrix ADC CP Cryptographic Library v3.0)

Certificate Number	Algorithm	Standard	Mode / Method	Key Lengths / Curves / Moduli	Use
C1918	AES[25]	FIPS PUB 197	CBC[26], CTR[27]	128, 192, 256	encryption/decryption
		NIST SP 800-38D	GCM[28]	128, 256	encryption/decryption
Vendor Affirmation	CKG[29]	NIST SP[30] 800-133	-	-	key generation

[22] CP – Control Plane
[23] FOM – FIPS Object Module
[24] DP – Data Plane
[25] AES – Advance Encryption Standard
[26] CBC – Cipher Block Chaining
[27] CTR – Counter
[28] GCM – Galois Counter Mode
[29] CKG – Cryptographic Key Generation
[30] SP – Special Publication

Certificate Number	Algorithm	Standard	Mode / Method	Key Lengths / Curves / Moduli	Use
C1918	CVL[31]	NIST SP 800-56Arev3	ECC CDH[33] Primitive	P-224, P-256, P-384, P-521	shared secret computation per NIST SP 800-56Arev3 and key derivation per NIST SP 800-135rev1 (Certs. #C855 and #C1917)
C1918	DRBG[34]	NIST SP 800-90Arev1	CTR-based	-	deterministic random bit generation
C1918	ECDSA[35]	FIPS PUB 186-4	KPG[36]	P-224, P-256, P-384, P-521	key pair generation
			SIG(gen), SIG(ver)	P-224, P-256, P-384, P-521	digital signature generation and verification
C1918	HMAC[37]	FIPS PUB 198-1	SHA[38]-1, SHA-256, SHA-384, SHA-512	160, 256, 384, 512	message authentication *The cryptographic library supports the truncation of HMAC SHA-1 to 96 bits according to NIST SP 800-107rev1.*
Vendor Affirmed	KAS-SSC[43]	NIST SP800-56Arev3	ECDH[44]	P-224, P-256, P-384, P-521	key agreement scheme - shared secret computation per NIST SP 800-56Arev3 and key derivation per NIST SP 800-135rev1 (Certs. #C855 and #C1917)
Vendor Affirmed	KAS-SSC	NIST SP800-56Arev3	DH (Groups 14, 15, 16, 17, and 18)	112 – 200-bits security strength (MODP-2048, MODP-3027, MODP-4096, MODP-6144, MODP-8192)	key agreement scheme - shared secret computation per NIST SP 800-56Arev3 and key derivation per NIST SP 800-135rev1 (Certs. #853, #C855, and #C1917)
			FFC[45] DH[46] Primitive		
Vendor Affirmed	PBKDF[54]	NIST SP 800-132	Option 1a with HMAC SHA-1, Option 1a with HMAC SHA-256	-	password-based key derivation
C1918	RSA[55]	FIPS PUB 186-4	KeyGen9.31	2048, 3072	key pair generation
			SigGenPKCS[56]1.5	2048, 3072	digital signature generation
			SigVerPKCS1.5	2048, 3072	digital signature verification

[31] CVL – Component Validation Listing
[33] ECC CDH – Elliptical Curve Cryptography Cofactor Diffie-Hellman
[34] DBRG – Deterministic Random Bit Generator
[35] ECDSA – Elliptic Curve Digital Signature Algorithm
[36] KPG – Key Pair Generation
[37] HMAC – (keyed-) Hashed Message Authentication Code
[38] SHA – Secure Hash Algorithm
[43] KAS-SSC – Key Agreement Scheme Shared Secret Computation
[44] ECDH – Elliptic Curve Diffie-Hellman
[45] FFC – Finite Field Cryptography
[46] DH – Diffie-Hellman
[54] PBKDF – Password-based Key Derivation Function 2
[55] RSA – Rivest Shamir Adleman
[56] PKCS – Public Key Cryptography Standard

Certificate Number	Algorithm	Standard	Mode / Method	Key Lengths / Curves / Moduli	Use
C1918	SHS[59]	FIPS PUB 180-4	SHA-1, SHA-256, SHA-384, SHA-512	-	message digest
C1918	Triple-DES[60]	NIST SP 800-67	CBC	Keying Option 1	encryption/decryption

The vendor affirms the following cryptographic security methods implemented by the Citrix ADC CP Cryptographic Library v3:

- Per *NIST SP 800-132*, the module uses PBKDF option 1a for KEK[62], PEM[63], and Kerberos Negotiate Key key establishment:
    - The PBKDF for KEK establishment takes an input salt that is 128 bits in length with a password/passphrase containing at least 8 characters and produces a random value of 256 bits. In addition, the function has an iteration count of 2,048. The underlying pseudorandom function used in this derivation is HMAC SHA-256.
    - The PBKDF for PEM key establishment takes an input salt that is 128 bits in length with a password/passphrase containing at least 8 characters and produces a random value of 256 bits for AES keys and 192-bits for Triple-DES keys. In addition, the function has an iteration count of 2,048. The underlying pseudorandom function used in this derivation is HMAC SHA-1.
    - The PBKDF for the Kerberos Negotiate Key key establishment takes an input salt that is 128 bits in length with a password/passphrase containing at least 8 characters and produces a random value of 256 bits for AES keys and 192-bits for Triple-DES keys. In addition, the function has an iteration count of 2,048. The underlying pseudorandom function used in this derivation is HMAC SHA-256.
    - The keys derived from these PBKDF functions are only used for storage applications.
- Per *NIST SP 800-133* section 4, the module uses the FIPS-Approved counter-based DRBG specified in *NIST SP 800-90Arev1* to generate both symmetric cryptographic keys and the random values used for asymmetric key generation. The resulting symmetric key or generated seed is an unmodified output from the DRBG.
- Key agreement scheme (shared secret computation) per *NIST SP 800-56Arev3*:
    - The module implements an FFC DH shared secret computation for its DH key agreement scheme. The shared secret computation is compliant with section 5.7.1.1 of *NIST SP 800-56Arev3*. This primitive is used by the dhHybrid1, dhEphem, dhHybridOneFlow, dhOneFlow and dhStatic schemes found in section 6 of that recommendation.

[59] SHS – Secure Hash Standard
[60] DES – Data Encryption Standard
[62] KEK – Key Encryption Key
[63] PEM – Privacy-Enhanced Mail

- The module also implements an ECC CDH shared secret computation for its ECDH key agreement scheme. The shared secret computation is compliant with section 5.7.1.2 of *NIST SP 800-56Arev3*. This primitive is used by the Full Unified Model, Ephemeral Unified Model, One-Pass Unified Model, One-Pass Diffie-Hellman, and Static Unified Model schemes found in section 6 of that recommendation.

Table 5 below lists the FIPS-Approved algorithms implemented by the Citrix ADC DP Cryptographic Library v3.

Table 5 – Algorithm Certificate Numbers (Citrix ADC DP Cryptographic Library v3.0)

Certificate Number	Algorithm	Standard	Mode / Method	Key Lengths / Curves / Moduli	Use
C1917	AES	FIPS PUB 197	CBC	128, 192, 256	encryption/decryption
		NIST SP 800-38D	GCM	128, 256	encryption/decryption
Vendor Affirmation	CKG	NIST SP 800-133	-	-	key generation
C1917	CVL	NIST SP 800-56Arev3	ECC CDH Primitive	P-224, P-256, P-384, P-521	shared secret computation per NIST SP 800-56Arev3 and key derivation per NIST SP 800-135rev1 (Cert. #C1919)
C1917	DRBG	NIST SP 800-90Arev1	Hash-based	-	deterministic random bit generation
C1917	ECDSA	FIPS PUB 186-4	SIG(gen), SIG(ver)	P-224, P-256, P-384, P-521	digital signature generation and verification
C1917	HMAC	FIPS PUB 198-1	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	160, 224, 256, 384, 512	message authentication
Vendor Affirmed	KAS-SSC	NIST SP 800-56Arev3	ECDH	P-224, P-256, P-384, P-521	key agreement scheme - shared secret computation per NIST SP 800-56Arev3 and key derivation per NIST SP 800-135rev1 (Cert. #C1919)
C1917	RSA	FIPS PUB 186-4	SigGenPKCS1.5	2048, 3072, 4096	digital signature generation
			SigVerPKCS1.5	1024, 2048, 3072	digital signature verification
A606	RSA	FIPS PUB 186-4	SigGenPKCS1.5	4096	digital signature generation
			SigVerPKCS1.5	4096	digital signature verification
C1917	SHS	FIPS PUB 180-4	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	-	message digest

The vendor affirms the following cryptographic security method implemented by the Citrix ADC DP Cryptographic Library v3:

- Per *NIST SP 800-133* section 4, the module uses the FIPS-Approved hash-based DRBG specified in *NIST SP 800-90A Revision 1* to generate both symmetric cryptographic keys and the random values used for asymmetric key generation. The resulting symmetric key or generated seed is an unmodified output from the DRBG.

- Key agreement scheme (shared secret computation) per *NIST SP 800-56Arev3*:

    - The module also implements an ECC CDH shared secret computation for its ECDH key agreement scheme. The shared secret computation is compliant with section 5.7.1.2 of *NIST SP 800-56Arev3*. This primitive is used by the Full Unified Model, Ephemeral Unified Model, One-Pass Unified Model, One-Pass Diffie-Hellman, and Static Unified Model schemes found in section 6 of that recommendation.

Per FIPS 140-2 IG 7.14, the module generates cryptographic keys whose strengths are modified by available entropy.

In addition, the module includes several protocol libraries that implement FIPS-Approved KDFs[64]. The IKE[65] KDFs are implemented by the Citrix ADC CP IKE KDF Library v1 (based on the Racoon2 protocol library), the SSH[66] KDF is implemented by the Citrix ADC CP SSH KDF Library v1 (based on the open source OpenSSH protocol library), and the SNMP[67]v3 KDF is implemented by the Citrix ADC CP SNMP KDF Library v1 (a modified version of the Net-SNMP protocol library). These libraries all link to the Citrix ADC CP Cryptographic Library v3 for their cryptographic operations.

There are two TLS[68] KDF implementations supported by the module. The first is implemented by the Citrix ADC CP TLS KDF Library v2 (based on the OpenSSL libssl protocol library) and uses the Citrix ADC CP Cryptographic Library v3 for its cryptographic operations. The second is implemented by the Citrix ADC DP Cryptographic Library v3.

The module implements the FIPS-Approved KDFs listed in Table 6 below.

Table 6 – CVL Certificate Numbers

Certificate Number	Algorithm	Specification	Mode / Method	Key Lengths / Curves / Moduli	Use	Library
C1917	TLS v1.0/1.1/1.2 KDF	NIST SP 800-135rev1	-	-	key derivation	Citrix ADC CP Cryptographic Library v3.0
C1919	TLS v1.0/1.1/1.2 KDF	NIST SP 800-135rev1	-	-	key derivation	Citrix ADC DP Cryptographic Library v3.0
C855	SSH KDF	NIST SP 800-135rev1	-	-	key derivation	Citrix ADC CP Cryptographic Library v3.0

[64] KDF – Key Derivation Function
[65] IKE – Internet Key Exchange
[66] SSH – Secure Shell
[67] SNMP – Simple Network Management Protocol
[68] TLS – Transport Layer Security

Certificate Number	Algorithm	Specification	Mode / Method	Key Lengths / Curves / Moduli	Use	Library
C853	IKEv1/v2 KDF	NIST SP 800-135rev1	-	-	key derivation	Citrix ADC CP Cryptographic Library v3.0
C854	SNMPv3 KDF	NIST SP 800-135rev1	-	-	key derivation	Citrix ADC CP Cryptographic Library v3.0

Note: *No parts of the IKE, SNMP, SSH, and TLS protocols, other than the KDFs, have been tested by the CAVP[69] or CMVP.*

The algorithm implementations shown in Table 7 below are allowed for use in a FIPS-Approved mode of operation.

Table 7 – Allowed Algorithm Implementations

Algorithm	Caveat	Use
RSA	key establishment methodology provides 112 or 128 bits of encryption strength	key transport (Citrix ADC CP Cryptographic Library v3)
	key establishment methodology provides 112 or 128-bits of encryption strength	key transport (Citrix ADC DP Cryptographic Library v3)
MD5[70]	-	hashing passwords; TLS 1.0/1.1 tunnel setup
NDRNG[71] (FreeBSD /dev/random)	-	seeding for the control plane DRBG
NDRNG (Intel RDRAND)	-	seeding for data plane hardware and firmware DRBG

2.3 Module Interfaces

The module's design separates the physical ports into four logically distinct and isolated categories. They are:

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

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

The mapping of the module's logical interfaces in the software to FIPS 140-2 logical interfaces is described in Table 3 below.

[69] CAVP – Cryptographic Algorithm Validation Program
[70] MD5 – Message Digest 5
[71] NDRNG – Non-Deterministic Random Number Generator

Table 8 – VPX Interface Mappings

Physical Port/Interface	Virtual Port/Interface	FIPS 140-2 Logical Interface
Host platform Network Interface (Ethernet 10/100/1000) ports	Virtual Network Interface	• Data Input • Data Output • Control Input • Status Output
Host platform Graphics Controller	Virtual Graphics Controller	• Status Output
Host platform LED/LCDs	Virtual LED/LCDs	• Status output
Host platform serial port	Virtual serial port (console)	• Data Input • Data output • Control Input • Status output
Host platform USB port	Virtual USB port	• Data input • Data output • Control input
Host platform power interface	Virtual power interface	• Power

2.4 Roles and Services

The sections below describe the module's roles and services and defines any authentication methods employed.

2.4.1 Authorized Roles

As required by FIPS 140-2, the module supports two roles that operators may assume:

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

- User – Users can view the current status of the module and employ the services of the module (including IPsec[72], TLS, SSH, and SNMPv3 services). The User role includes the privileges listed under the read-only VPX command policy.

For more information on the VPX command policies, refer to the Configuring users, user groups, and command policies webpage on Citrix's online product documentation portal.

[72] IPsec – Internet Protocol Security

2.4.2 Operator Services

Descriptions of the services available to the CO role and User role are provided in Table 9 below. Please note that the keys and Critical Security Parameters (CSPs) listed in the table indicate the type of access required using the following notation:

- R – Read: The CSP is read.
- W – Write: The CSP is established, generated, or modified.
- X – Execute: The CSP is used within an Approved or Allowed security function or authentication mechanism.
- Z – Zeroize: The CSP is deleted.

Table 9 – Mapping of Module Services to Roles, CSPs, and Type of Access[73]

Service	Operator		Description	Input	Output	CSP and Type of Access
	CO	User				
Perform initial network configuration	✓		Set up initial network configuration and VPX licenses	Command and parameters	Command response/ status output	None
Create KEK	✓		Create system master key	Command	Status output	KEK Passphrase – R/X KEK – W CTR DRBG Entropy – R/X CTR DRBG Seed – R/W/X CTR DRBG 'V' Value – R/W/X CTR DRBG 'Key' Value – R/W/X
View system information	✓		View system info and statistics; view/end system sessions	Command	Status output	None

[73] For commands and parameters related to the listed services, refer to the Citrix ADC 12.1 Product Documentation located at https://docs.citrix.com/en-us/citrix-adc/12-1.html.

Service	Operator		Description	Input	Output	CSP and Type of Access
	CO	User				
Reboot	✓		Reboot the module	Command	Status output	PEM Passphrase – Z PEM Key – Z AES GCM Key – Z AES GCM IV – Z DH/ECDH/RSA Private Key Component – Z DH/ECDH/RSA Public Key Component – Z SSH Shared Secret – Z SSH Session Key – Z SSH Authentication Key – Z IKE/IPsec Shared Secret – Z IKE/IPsec Session Key – Z IKE/IPsec Authentication Key – Z TLS Pre-Master Secret – Z TLS Master Secret – Z TLS Session Key – Z TLS Authentication Key – Z TLS Ticket Encryption Key – Z TLS Authentication Key – Z Hash DRBG Entropy – Z Hash DRBG Seed – Z Hash DRBG “V” Value” – Z Hash DRBG “C” Value – Z CTR DRBG Entropy – Z CTR DRBG Seed – Z CTR DRBG “V” Value – Z CTR DRBG “Key” Value – Z SNMPv3 Private Key – Z SNMPv3 Authentication Key - Z
Configure system settings	✓		Configure modes and features, system settings, and cloud parameters	Command and parameters	Command response/ status output	AES Key – W KEK – X Hash DRBG Entropy – R/X Hash DRBG Seed – R/W/X Hash DRBG 'V' Value – R/W/X Hash DRBG 'C' Value – R/W/X
Configure HA[74]	✓		Configure HA nodes, route monitors, failover interface set	Command and parameters	Status output	None
Manage NTP[75] servers	✓		Add, edit, delete NTP servers; configure NTP parameters and synchronization state	Command	Status output	None
Configure system profiles	✓		Add, edit, delete system profiles	Command and parameters	Command response/ status output	TLS Master Secret – R/W/X TLS Ticket Encryption Key – R/W TLS Ticket Authentication Key – R/W KEK – X CTR DRBG Entropy – R/X CTR DRBG Seed – R/W/X CTR DRBG 'V' Value – R/W/X CTR DRBG 'Key' Value – R/W/X

[74] HA – High Availability
[75] NTP – Network Time Protocol

Service	Operator		Description	Input	Output	CSP and Type of Access
	CO	User				
Manage users	✓		Add, edit delete users, groups, and command policies; view user/group partition bindings	Command	Status output	None
Configure system auditing	✓		Add, edit, delete syslog/nslog auditing policies and servers; bind classic/advanced global policies	Command and parameters	Command response/ status output	None
View audit logs	✓		View authentication, system, and event logs	Command	Status output	None
Configure network settings	✓		Configure network routing protocols	Command and parameters	Command response/ status output	ZebOS Router Password – R/W KEK – X
Exchange routing information	✓		Exchange routing update information using ZebOS, authenticate source of packets	Command	Status output	ZebOS Router Password – X KEK – X
Configure SSH	✓		Configure SSH authentication settings; generate SSH keys	Command and parameters	Command response/ status output	SSH Private Key – W/X SSH Public Key – W CTR DRBG Entropy – R/X CTR DRBG Seed – R/W/X CTR DRBG 'V' Value – R/W/X CTR DRBG 'Key' Value – R/W/X
Establish SSH sessions	✓	✓	Establish an SSH session	Command	Status output	SSH Public Key – R/X DH/ECDH Private Key Component – W/X DH/ECDH Public Key Component – R/X SSH Shared Secret – W/X SSH Session Key – W/X SSH Authentication Key – W/X CTR DRBG Entropy – R/X CTR DRBG Seed – R/W/X CTR DRBG 'V' Value – R/W/X CTR DRBG 'Key' Value – R/W/X
Configure CloudBridge	✓		Configure IPsec profile; configure CloudBridge Connector settings, network bridges, and IP tunnels; view IP tunnel details	Command and parameters	Command response/ status output	IKE/IPsec PSK[76] – R/W KEK – X
Configure clustering	✓		Configure an appliance to either be the cluster coordinator or a node in the cluster	Command and parameters	Command response/ status output	Cluster Password – R/W

[76] PSK – Pre-shared Key

Service	Operator		Description	Input	Output	CSP and Type of Access
	CO	User				
Establish IPsec session	✓	✓	Establish an IPsec Session	Command	Status output	DH Private Key Component – W/X DH Public Key Component – R/X IKE/IPsec Shared Secret – W/X IKE/IPsec PSK – X KEK – X IKE/IPsec Session Key – W/X IKE/IPsec Authentication Key – W/X CTR DRBG Entropy – R/X CTR DRBG Seed – R/W/X CTR DRBG 'V' Value – R/W/X CTR DRBG 'Key' Value – R/W/X
Backup and restore	✓		Backup/import system configuration files; download and delete backup files; restore	Command	Status output	None
Manage encryption keys	✓		Add, edit, delete encryption keys	Command	Status output	AES Key – R/W KEK – X Hash DRBG Entropy – R/X Hash DRBG Seed – R/W/X Hash DRBG 'V' Value – R/W/X Hash DRBG 'C' Value – R/W/X
Manage HMAC keys	✓		Add, edit, delete HMAC keys	Command	Status output	HMAC Key – R/W KEK – X Hash DRBG Entropy – R/X Hash DRBG Seed – R/W/X Hash DRBG 'V' Value – R/W/X Hash DRBG 'C' Value – R/W/X

Service	Operator		Description	Input	Output	CSP and Type of Access
	CO	User				
Configure traffic management	✓		Configure TLS; Configure load balancing, priority load balancing, content switching, and cache redirection settings, DNS[77], GSLB[78], Subscriber, service chaining, and user protocol settings	Command and parameters	Command response/ status output	CA[79] Public Key – R/W/X TLS Private Key – R/W/X TLS Public Key – R/W Private DNS KSK[80] – R/W/X Public DNS KSK – R/W/X Private DNS ZSK[81] – R/W/X Public DNS ZSK – R/W/X SSH Private Key – R/W/X SSH Public Key – R/W/X PEM Passphrase – R/W/X PEM Key – W/X KEK – X CTR DRBG Entropy – R/X CTR DRBG Seed – R/W/X CTR DRBG 'V' Value – R/W/X CTR DRBG 'Key' Value – R/W/X Hash DRBG Entropy – R/X Hash DRBG Seed – R/W/X Hash DRBG 'V' Value – R/W/X Hash DRBG 'C' Value – R/W/X
Establish TLS session	✓	✓	Establish a web session using TLS protocol	Command	Status output	TLS Public Key – R/X DH Private Key Component – W/X DH Public Key Component – R/X ECDH Private Key Component – W/X ECDH Public Key Component – R/X RSA Private Key Component – W/X RSA Public Key Component – R/X TLS Premaster Secret – R/W/X TLS Master Secret – W/X TLS Session Key – W/X TLS Authentication Key – W/X AES GCM IV[82] – W/X AES GCM Key – W/X PEM Passphrase – R/X PEM Key – W/X KEK – X CTR DRBG Entropy – R/X CTR DRBG Seed – R/W/X CTR DRBG 'V' Value – R/W/X CTR DRBG 'Key' Value – R/W/X Hash DRBG Entropy – R/X Hash DRBG Seed – R/W/X Hash DRBG 'V' Value – R/W/X Hash DRBG 'C' Value – R/W/X

[77] DNS – Domain Name System
[78] GSLB – Global Server Load Balancing
[79] CA – Certificate Authority
[80] KSK – Key Signing Key
[81] ZSK – Zone Signing Key
[82] IV – Initialization Vector

Service	Operator		Description	Input	Output	CSP and Type of Access
	CO	User				
Resume TLS session	✓	✓	Resume a web session using TLS protocol	Command	Status output	TLS Ticket Encryption Key – R/W/X TLS Ticket Authentication Key – R/W/X TLS Session Key – R/X TLS Authentication Key – R/X AES GCM IV – W/X AES GCM Key – W/X KEK – X Hash DRBG Entropy – R/X Hash DRBG Seed – R/W/X Hash DRBG 'V' Value – R/W/X Hash DRBG 'C' Value – R/W/X
Apply data policies	✓		Apply data policies to user data in transit (according to configuration)	Command	Status output	AES Key – X HMAC Key – X KEK – X
Configure security	✓		Configure DNS security profiles, application firewall profiles and policies, reputation settings, protection features, and content inspection policies	Command and parameters	Command response/ status output	None
Configure Citrix ADC Gateway	✓		Configure Gateway global settings, virtual servers, portal themes, AAA[83] groups and users, policies, and resources	Command and parameters	Command response/ status output	RDP[84] PSK – W KEK – X
Establish Citrix ADC Gateway	✓		Establish Gateway connection based on global settings	Command and parameters	Command response/ status output	RDP PSK – R/X KEK – X
Configure external servers for system, AAA, and Gateway authentication	✓		Configure LDAP[85], Oauth, OpenID, DFA[86], Kreberos, and SAML[87] servers to be used in system, AAA, or Gateway authentication	Command and parameters	Command response/ status output	LDAP Admin Password – R/W Oauth Client Secret – R/W DFA Shared Secret – R/W Kerberos CA Public Key – R/W Kerberos User Public Key – R/W Kerberos User Private Key – R/W Kerberos Negotiate Key – W KEK – X

[83] AAA – Authentication, Authorization, Accounting
[84] RDP – Remote Desktop Protocol
[85] LDAP – Lightweight Directory Access Protocol
[86] DFA – Delegated Form Authentication
[87] SAML – Security Assertion Markup Language

Service	Operator		Description	Input	Output	CSP and Type of Access
	CO	User				
Perform Kerberos functions	✓	✓	Establish Kerberos session; Access Kerberos service; Perform Kerberos negotiation	Command	Status Output	Kerberos CA Public Key – W/X Kerberos User Public Key – W Kerberos User Private Key – X Kerberos Server Public Key - W Kerberos DH Public Key – R/W/X Kerberos DH Private Key – R/W/X Kerberos Secret Key – W/X Kerberos Client/TGS[88] Session Key – R/X Kerberos Client/Server Session Key – W, X Kerberos Negotiate Key – R/X
Configure SNMPv3	✓		Configure SNMP communities, traps, managers, views, groups, users, alarms, and engine ID[89]; view SNMP OIDs[90]	Command and parameters	Command response/ status output	SNMPv3 Authentication Passphrase – R/W SNMPv3 Privacy Passphrase – R/W KEK – X
SNMPv3 traps	✓	✓	Provides system condition information	None	Status output	SNMPv3 Authentication Passphrase – X SNMPv3 Privacy Passphrase – X SNMPv3 Privacy Key – W/X SNMPv3 Authentication Key – W/X
Show status	✓	✓	Show the system status	Command	Status output	None
Zeroize KEK	✓		Zeroize KEK	Command	Status output	KEK – Z
Zeroize SSH private keys	✓		Zeroize SSH private keys	Command	Status output	SSH Private Key – Z

2.4.3 Additional Services

The module provides a limited number of services for which the operator is not required to assume an authorized role. Table 10 lists the services for which the operator is not required to assume an authorized role. None of these services disclose or substitute cryptographic keys and CSPs or otherwise affect the security of the module.

[88] TGS – Ticket Granting Service
[89] ID – Identifier
[90] OID – Object Identifier

Table 10 – Additional Services[91]

Service	Description	Input	Output	CSP and Type of Access
Zeroize	Zeroize keys and CSPs	Power cycle	Status output	PEM Passphrase – Z PEM Key – Z AES GCM Key – Z AES GCM IV – Z DH/ECDH/RSA Private Key Component – Z DH/ECDH/RSA Public Key Component – Z SSH Shared Secret – Z SSH Session Key – Z SSH Authentication Key – Z IKE/IPsec Shared Secret – Z IKE/IPsec Session Key – Z IKE/IPsec Authentication Key – Z TLS Pre-Master Secret – Z TLS Master Secret – Z TLS Session Key – Z TLS Authentication Key – Z TLS Ticket Encryption Key – Z TLS Authentication Key – Z Hash DRBG Entropy – Z Hash DRBG Seed – Z Hash DRBG "V" Value" – Z Hash DRBG "C" Value – Z CTR DRBG Entropy – Z CTR DRBG Seed – Z CTR DRBG "V" Value – Z CTR DRBG "Key" Value – Z SNMPv3 Private Key – Z SNMPv3 Authentication Key - Z

[91] For commands and parameters related to the listed services, refer to the Citrix ADC 12.1 Product Documentation located at https://docs.citrix.com/en-us/citrix-adc/12-1.html.

Service	Description	Input	Output	CSP and Type of Access
Perform On-Demand Self-Tests	Perform self-tests on demand	Power cycle	Status output	PEM Passphrase – Z PEM Key – Z AES GCM Key – Z AES GCM IV – Z DH/ECDH/RSA Private Key Component – Z DH/ECDH/RSA Public Key Component – Z SSH Shared Secret – Z SSH Session Key – Z SSH Authentication Key – Z IKE/IPsec Shared Secret – Z IKE/IPsec Session Key – Z IKE/IPsec Authentication Key – Z TLS Pre-Master Secret – Z TLS Master Secret – Z TLS Session Key – Z TLS Authentication Key – Z TLS Ticket Encryption Key – Z TLS Authentication Key – Z Hash DRBG Entropy – Z Hash DRBG Seed – Z Hash DRBG “V” Value” – Z Hash DRBG “C” Value – Z CTR DRBG Entropy – Z CTR DRBG Seed – Z CTR DRBG “V” Value – Z CTR DRBG “Key” Value – Z SNMPv3 Private Key – Z SNMPv3 Authentication Key - Z
Authenticate operators	Used for operator logins to the module	Command	Status output	Operator Password - R LDAP Admin Password – R/X SSH Public Key – X Oauth Client Secret – X DFA Shared Secret – X TLS Public Key – X AES Key – X AES GCM Key – X AES GCM IV – X KEK – X

2.5 Physical Security

The VPX is a software module, which FIPS defines as a multiple-chip standalone cryptographic module. As such, it does not include physical security mechanisms. Therefore, as per Section G.3 of the Implementation Guidance for FIPS PUB 140-2 and the CMVP, requirements for physical security are not applicable.

2.6 Operational Environment

The operational environment of the module does not provide a general-purpose OS to module operators.

The VPX runs on the FreeBSD v8.4 OS, which acts as the guest OS on top of the virtualization layer. The virtualization layer is provided by VMware’s ESXi hypervisor v6.5. 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.

The module generates cryptographic keys whose strengths are modified by available entropy. The module receives 256-bits of full entropy from the module's entropy source, more than the minimum FIPS requirement of 112 bits of entropy.

2.7 Cryptographic Key Management

The module supports the CSPs listed below in Table 11.

Table 11 – Cryptographic Keys, Cryptographic Key Components, and CSPs

CSP	CSP Type	Generation / Input	Output	Storage	Zeroization	Use
KEK Passphrase	Alphanumeric string	Generated externally, input in plaintext form via local console or in encrypted form via SSH session	Never exits the module	Plaintext in volatile memory	N/A	Derivation of KEK
KEK	256-bit AES key	Generated internally via PBKDF	Never exits the module	Plaintext on disk	CLI command	Encryption and decryption of passwords and passphrases
PEM Passphrase	Alphanumeric string (8-31 characters)	Generated externally, input in plaintext form via local console or in encrypted form via TLS or SSH session	Exits the module in encrypted form as part of config backup file	Plaintext in volatile memory or encrypted on disk (via KEK)	[for plaintext] Reboot; remove power	Derivation of PEM Key
PEM Key	256-bit AES key 192-bit Triple-DES key	Generated internally via PBKDF	Never exits the module	Plaintext in volatile memory	Reboot; remove power	Encryption and decryption of asymmetric private keys
AES key	128/192/256-bit AES key	Generated internally via Approved DRBG OR Generated externally, imported in plaintext form via local console or in encrypted form via TLS or SSH session	Exits the module in encrypted form as part of config backup file	Encrypted on disk (via KEK)	N/A	Encryption and decryption
AES GCM key	256-bit AES GCM key	Generated internally via Approved DRBG	Never exits the module	Plaintext in volatile memory	Reboot; remove power	Encryption and decryption
AES GCM IV	96-bit IV	Internally generated deterministically in compliance with TLS 1.2 GCM Cipher Suites for TLS and Section 8.2.1 of NIST SP 800-38D	Never exits the module	Plaintext in volatile memory	Reboot; remove power	IV for AES GCM

CSP	CSP Type	Generation / Input	Output	Storage	Zeroization	Use
HMAC Key	160/224/256/384/512-bit HMAC key	Generated internally via Approved DRBG OR Generated externally, imported in plaintext form via local console or in encrypted form via TLS or SSH session	Exits the module in encrypted form as part of config backup file	Encrypted on disk (via KEK)	N/A	Message authentication with SHS
CA Public Key	1024/2048/3072/4096-bit RSA public key P-224/P-256/P-384/P-521 ECDSA public key	Generated externally, imported in plaintext form via local console or in encrypted form via TLS or SSH session	Exits the module in plaintext form	Plaintext on disk	N/A	TLS certificate authentication **1024-bit RSA public keys are used for signature verification only**
DH Private Key Component	[for SSH sessions] 2048, 4096, 8192-bit DH private key [for TLS sessions] 2048, 3072, 4096-bit DH private key [for IKE sessions] 2048-bit DH private key	Generated internally via Approved DRBG	Never exits the module	Plaintext in volatile memory	Reboot; remove power; session termination	Generation of SSH, TLS, and IKE shared secrets
DH Public Key Component	[for SSH sessions] 2048, 4096, 8192-bit DH public key [for TLS sessions] 2048, 3072, 4096-bit DH public key [for IKE sessions] 2048-bit DH public key	[for the module] Generated internally via Approved DRBG [for a peer] Input in plaintext form	[for the module] Exits the module in plaintext form [for a peer] Never exits the module	Plaintext in volatile memory	Reboot; remove power; session termination	Generation of SSH, TLS, and IKE shared secrets

CSP	CSP Type	Generation / Input	Output	Storage	Zeroization	Use
ECDH Private Key Component	Private key of ECDH protocol (P-224/P-256/P-384/P-521 curves)	Generated internally via Approved DRBG	Never exits the module	Plaintext in volatile memory	Reboot; remove power; session termination	Generation of SSH and TLS shared secrets
ECDH Public Key Component	Public key of ECDH protocol (P-224/P-256/P-384/P-521 curves)	[for the module] Generated internally via Approved DRBG [for a peer] Input in plaintext form	[for the module] Exits the module in plaintext form [for a peer] Never exits the module	Plaintext in volatile memory	Reboot; remove power; session termination	Generation of SSH and TLS shared secrets
RSA Private Key Component	2048, 3072-bit RSA Private Key	Generated internally via Approved DRBG	Never exits the module	Plaintext in volatile memory	Reboot; remove power; session termination	Generation of TLS shared secrets
RSA Public Key Component	2048, 3072-bit RSA Public Key	[for the module] Generated internally via Approved DRBG [for a peer] Input in plaintext form	[for the module] Exits the module in plaintext form [for a peer] Never exits the module	Plaintext in volatile memory	Reboot; remove power; session termination	Generation of TLS shared secrets
SSH Public Key	2048/3072-bit RSA public key P-224/P-256/P-384/P-521 ECDSA public key	[for the module] Generated internally via Approved DRBG OR Generated externally, imported in plaintext form [for a peer] Input in plaintext form	[for the module] Exits the module in plaintext form [for a peer] Never exits the module	[for the module] Plaintext on disk [for a peer] Plaintext in volatile memory	N/A	Authentication during SSH session negotiation; GSLB configuration sync

CSP	CSP Type	Generation / Input	Output	Storage	Zeroization	Use
SSH Private Key	2048/3072-bit RSA private key P-224/P-256/P-384/P-521 ECDSA public key	Generated internally via Approved DRBG	Exits the module in encrypted form as part of config backup file	Plaintext on disk	CLI command	Authentication during SSH session negotiation; RBA[92] Authentication for LDAP; GSLB configuration sync
SSH Shared Secret	Shared secret	Derived internally via DH/ECDH shared secret computation	Never exits the module	Plaintext in volatile memory	Reboot; remove power; session termination	Derivation of the SSH Session Key and SSH Authentication Key
SSH Session Key	128/192/256-bit AES key (CBC and CTR mode) 192-bit Triple-DES key	Derived internally via SSH KDF	Never exits the module	Plaintext in volatile memory	Reboot; remove power; session termination	Encryption and decryption of SSH session packets
SSH Authentication Key	160/256/512-bit HMAC key	Derived internally via SSH KDF	Never exits the module	Plaintext in volatile memory	Reboot; remove power; session termination	Authentication of SSH session packets
IKE/IPsec Shared Secret	Shared secret	Derived internally via DH shared secret computation	Never exits the module	Plaintext in volatile memory	Reboot; remove power; session termination	Derivation of the IKE/IPsec Session Keys and IKE/IPsec Authentication Keys
IKE/IPsec PSK	Pre-shared key	Generated externally, imported in plaintext form via local console or in encrypted form via TLS or SSH session	Exits the module in encrypted form as part of config backup file	Encrypted on disk (via KEK)	N/A	Authentication during IKE/IPsec session negotiation [IKEv1 Only] Derivation of the IKE/IPsec Session Keys and IKE/IPsec Authentication Keys
IKE/IPsec Session Key	128/192/256-bit AES key	Derived internally via IKE KDF	Never exits the module	Plaintext in volatile memory	Reboot; remove power; session termination	Encryption and decryption of IKE/IPsec session packets
IKE/IPsec Authentication Key	160/256/384/512-bit HMAC key	Derived internally via IKE KDF	Never exits the module	Plaintext in volatile memory	Reboot; remove power; session termination	Authentication of IKE/IPsec session packets

[92] RBA – Role-based Authentication

CSP	CSP Type	Generation / Input	Output	Storage	Zeroization	Use
TLS Public Key	1024/2048/3072/4096-bit RSA public key P-224/P-256/P-384/P-521 ECDSA public key	[for the module] Generated internally via Approved DRBG (1024/2048/3072-bit) OR Generated externally, imported in plaintext form via local console or in encrypted form via TLS or SSH session (1024/2048/3072/4096-bit) [for a peer] Input in plaintext form as part of TLS session negotiation 1024/2048/3072/4096-bit	[for the module] Exits the module in plaintext form [for a peer] Never exits the module	[for the module] Plaintext on disk [for a peer] Plaintext in volatile memory	N/A	TLS authentication; SAML authentication (RSA only); OpenID authentication (RSA only) **1024-bit RSA public keys are used for signature verification only**
TLS Private Key	2048/3072/4096-bit RSA private key P-224/P-256/P-384/P-521 ECDSA public key	Generated internally via Approved DRBG (2048/3072-bit) OR Generated externally, imported in plaintext form via local console or in encrypted form via TLS or SSH session (2048/3072/4096-bit)	Exits the module in encrypted form as part of config backup file	Encrypted on disk (via PEM key)	N/A	TLS authentication; SAML authentication (RSA only); OpenID authentication (RSA only)
TLS Pre-Master Secret	[for RSA cipher suites] 384-bit random value [for DH/ECDH cipher suites] DH/ECDH shared secret	[for RSA cipher suites and module acting as client] Generated internally via FIPS-Approved DRBG [for RSA cipher suites and module acting as server] Generated externally, imported in encrypted form via RSA key transport [for DH/ECDH cipher suites] Derived internally via DH/ECDH shared secret computation	[for RSA cipher suites and module acting as client] Exits the module in encrypted form via RSA key transport [for RSA cipher suites and module acting as server] Never exits the module [for DH/ECDH cipher suites] Never exits the module	Plaintext in volatile memory	Reboot; remove power; completion of TLS Session Key and TLS Authentication Key derivation	Derivation of the TLS Master Secret

CSP	CSP Type	Generation / Input	Output	Storage	Zeroization	Use
TLS Master Secret	384-bit shared secret	Derived internally using the TLS Pre-Master Secret via TLS KDF	Never exits the module	Plaintext in volatile memory	Reboot; remove power; session termination	Derivation of the TLS Session Key and TLS Authentication Key
TLS Session Key	128/256-bit AES key 128/256-bit AES GCM key	Derived internally using the TLS Master Secret via TLS KDF	Never exits the module	Plaintext in volatile memory	Reboot; remove power; session termination	Encryption and decryption of TLS session packets
TLS Authentication Key	160/256/384-bit HMAC key	Derived internally using the TLS Master Secret via TLS KDF	Never exits the module	Plaintext in volatile memory	Reboot; remove power; session termination	Authentication of TLS session packets
TLS Ticket Encryption Key	128-bit AES key	Generated internally via Approved DRBG OR Generated externally, imported in plaintext form via local console or in encrypted form via TLS or SSH session	Never exits the module	[for internally generated keys] Plaintext in volatile memory [for imported keys] Encrypted on disk (via KEK)	[for internally generated keys] Reboot; remove power	Encryption and decryption of TLS session tickets
TLS Ticket Authentication Key	256-bit HMAC key	Generated internally via Approved DRBG OR Generated externally, imported in plaintext form via local console or in encrypted form via TLS or SSH session	Never exits the module	[for internally generated keys] Plaintext in volatile memory [for imported keys] Encrypted on disk (via KEK)	[for internally generated keys] Reboot; remove power	Computes the digest of TLS session tickets
Hash DRBG Entropy	256-bit value	Generated externally	Never exits the module	Plaintext in volatile memory	Reboot; remove power	Entropy input for Hash DRBG
Hash DRBG Seed	440-bit value	Generated internally	Never exits the module	Plaintext in volatile memory	Reboot; remove power	Seed material for Hash DRBG
Hash DRBG ‘V’ Value	Internal state value	Generated internally	Never exits the module	Plaintext in volatile memory	Reboot; remove power	Internal state value used with Hash DRBG
Hash DRBG ‘C’ Value	Internal state value	Generated internally	Never exits the module	Plaintext in volatile memory	Reboot; remove power	Internal state value used with Hash DRBG

CSP	CSP Type	Generation / Input	Output	Storage	Zeroization	Use
CTR DRBG Entropy	256-bit value	Generated externally	Never exits the module	Plaintext in volatile memory	Reboot; remove power	Entropy input for CTR DRBG
CTR DRBG Seed	384-bit value	Generated internally	Never exits the module	Plaintext in volatile memory	Reboot; remove power	Seed material for CTR DRBG
CTR DRBG 'V' Value	128-bit value	Generated internally	Never exits the module	Plaintext in volatile memory	Reboot; remove power	Internal state value used with CTR DRBG
CTR DRBG 'Key' Value	256-bit AES key	Generated internally	Never exits the module	Plaintext in volatile memory	Reboot; remove power	Internal state value used with CTR DRBG
SNMPv3 Privacy Passphrase	Alphanumeric string	Input in plaintext form via local console or in encrypted form via TLS or SSH session	Exits the module in encrypted form as part of config backup file	Encrypted on disk (via KEK)	N/A	Derivation of the SNMPv3 Privacy Key
SNMPv3 Authentication Passphrase	Alphanumeric string	Input in plaintext form via local console or in encrypted form via TLS or SSH session	Exits the module in encrypted form as part of config backup file	Encrypted on disk (via KEK)	N/A	Derivation of the SNMPv3 Authentication Key
SNMPv3 Privacy Key	128-bit AES key	Derived internally via the SNMP KDF	Never exits the module	Plaintext in volatile memory	Reboot; remove power	Encryption and decryption of SNMPv3 packets
SNMPv3 Authentication Key	160-bit HMAC key	Derived internally via the SNMP KDF	Never exits the module	Plaintext in volatile memory	Reboot; remove power	Authentication of SNMPv3 packets
LDAP Admin Password	Alphanumeric string	Input in plaintext form via local console or in encrypted form via TLS or SSH session	Exits the module in encrypted form as part of config backup file	Encrypted on disk (via KEK)	N/A	Used to bind to the LDAP server
RDP PSK	Shared secret	Input in plaintext form via local console or in encrypted form via TLS or SSH session	Exits the module in encrypted form as part of config backup file	Encrypted on disk (via KEK)	N/A	Encryption and decryption of RDP user and target information
Oauth Client Secret	Shared secret	Input in plaintext form via local console or in encrypted form via TLS or SSH session	Exits the module in encrypted form as part of config backup file	Encrypted on disk (via KEK)	N/A	Oauth and Oauth IDP[93] authentication to the module
DFA Shared Secret	Shared secret	Input in plaintext form via local console or in encrypted form via TLS or SSH session	Exits the module in encrypted form as part of config backup file	Encrypted on disk (via KEK)	N/A	DFA authentication to the module

[93] IDP – Identity Provider

CSP	CSP Type	Generation / Input	Output	Storage	Zeroization	Use
ZebOS Router Password	Alphanumeric string	Input in plaintext form via local console or in encrypted form via TLS or SSH session	Exits the module in encrypted form as part of config backup file	Encrypted on disk (via KEK)	N/A	Router authentication
Public DNS KSK	2048/3072/4096-bit RSA public key	Generated internally (2048/3072-bit) Generated externally, imported in plaintext form via local console or in encrypted form via TLS or SSH session (2048/3072/4096-bit)	Exits the module in plaintext form as part of config backup file	Plaintext on disk	N/A	Public DNS ZSK authentication
Private DNS KSK	2048/3072/4096-bit RSA private key	Generated internally (2048/3072-bit) Generated externally, imported in plaintext form via local console or in encrypted form via TLS or SSH session (2048/3072/4096-bit)	Exits the module in encrypted form as part of config backup file	Encrypted on disk (via PEM key)	N/A	Public DNS ZSK signature generation
Public DNS ZSK	2048/3072/4096-bit RSA public key	Generated internally (2048/3072-bit) Generated externally, imported in plaintext form via local console or in encrypted form via TLS or SSH session (2048/3072/4096-bit)	Exits the module in plaintext form as part of config backup file	Plaintext on disk	N/A	DNS zone authentication
Private DNS ZSK	2048/3072/4096-bit RSA private key	Generated internally (2048/3072-bit Generated externally, imported in plaintext form via local console or in encrypted form via TLS or SSH session (2048/3072/4096-bit)	Exits the module in encrypted form as part of config backup file	Encrypted on disk (via PEM key)	N/A	DNS zone signature generation
Kerberos CA Public Key	2048-bit RSA public key	Generated externally, imported into the module [for a peer] Input in plaintext form as part of the Kerberos authentication	Exits the module in plaintext [for a peer] Never exits the module	Plaintext on disk [for a peer] Plaintext in volatile memory	Reboot; remove power	Used in Kerberos authentication

CSP	CSP Type	Generation / Input	Output	Storage	Zeroization	Use
Kerberos User Public Key	2048-bit RSA public key	Generated internally via Approved DRBG OR Generated externally, imported in plaintext form via local console or in encrypted form via TLS or SSH session	Exits the module in plaintext	Plaintext on disk	Reboot; remove power	Used in Kerberos authentication
Kerberos User Private Key	2048-bit RSA private key	Generated internally via Approved DRBG OR Generated externally, imported in plaintext form via local console or in encrypted form via TLS or SSH session	Never exits the module	Encrypted on disk (via PEM key)	Reboot; remove power	Used to sign authentication request
Kerberos Server Public Key	2048-bit RSA public key	Generated externally, imported into the module in plaintext	Never exits the module	Plaintext in volatile memory	Reboot; remove power	Used in Kerberos authentication
Kerberos DH Public Key	2048-bit DH public key	{For the module} Generated internally [For a peer] Generated externally, imported into the module in plaintext	Exits the module in plaintext form	Plaintext in volatile memory	Reboot; remove power	Used to generate the Kerberos secret key
Kerberos DH Private Key	2048-bit DH private key	Generated internally	Never exits the module	Plaintext in volatile memory	Reboot; remove power	Used to generate the Kerberos secret key
Kerberos Secret Key	256-bit AES key	Generated internally using DH components	Never exits the module	Plaintext in volatile memory	Reboot; remove power	Used to encrypt the Kerberos Client/TGS Session Key
Kerberos Client/TGS Session Key	256-bit AES key	Generated externally, imported electronically in encrypted form	Never exits the module	Plaintext in volatile memory	Reboot; remove power	Used in the Kerberos Client Authentication. Used to decrypt the Kerberos Client/Server Session Key.
Kerberos Client/Server Session Key	256-bit AES key	Generated externally, imported electronically in encrypted form	Never exits the module	Plaintext in volatile memory	Reboot; remove power	Used to authenticate to the Kerberos Service Server

CSP	CSP Type	Generation / Input	Output	Storage	Zeroization	Use
Kerberos Negotiate Key	256-bit AES key	Generated internally via PBKDF	Never exits the module	Plaintext in volatile memory	Reboot; remove power	Used to decrypt and verify packets received from the Kerberos server
Cluster Password	Alphanumeric string	Input in plaintext form via local console or in encrypted form via TLS or SSH session	Exits the module in encrypted form	Encrypted on disk (via KEK)	N/A	Used to connect nodes to the cluster coordinator
Operator Password	Alphanumeric string	Input in plaintext form via TLS or SSH session	Exists the module in encrypted form	Plaintext in volatile memory	Reboot; remove power	Authenticate the operator to the module via an external authentication service

Note: *Keys derived from the PBKDF function shall only be used for storage applications.*

All RSA and ECDSA keys at 2048 and 3072-bit modulus size are generated internally by the Citrix ADC CP Cryptographic Library v3. All RSA and ECDSA keys at the 4096-bit modulus size are generated outside of the module and input either in plaintext form via local console or encrypted form via a TLS or SSH session.

The AES-GCM IV is used in the TLS protocol. The TLS AES-GCM IV is generated in compliance with TLS v1.2 GCM cipher suites as specified in RFC[94] 5288 and section 3.3.1 of NIST SP 800-52rev1. Per RFC 5246, when the nonce_explicit part of the IV exhausts the maximum number of possible values for a given session key, the module will trigger a handshake to establish a new encryption key. The AES-GCM IV is a random 96-bit value generated with available entropy provided by the available entropy source.

[94] RFC – Request For Comment

2.8 EMI / EMC

The module's host servers were tested and found conformant to the EMI/EMC requirements specified by 47 Code of Federal Regulations, Part 15, Subpart B, Unintentional Radiators, Digital Devices, Class A (business use).

2.9 Self-Tests

Cryptographic self-tests are performed automatically by the module when the module is first powered up and loaded into memory as well as conditionally. The following sections list the self-tests performed by the module, their expected error status, and the error resolutions.

2.9.1 Power-Up Self-Tests

The VPX performs the following self-tests at power-up:

- Software integrity test (using RSA 2048 with SHA-512)
- Citrix ADC CP Cryptographic Library v3 self-tests
  - AES CBC encrypt KAT[95] (256-bit length)
  - AES CBC decrypt KAT (256-bit length)
  - AES GCM encrypt KAT
  - AES GCM decrypt KAT
  - Triple-DES CBC encrypt KAT
  - Triple-DES CBC decrypt KAT
  - SHA-1, SHA-256, SHA-512 KAT
  - HMAC KAT with SHA-1, SHA-256, and SHA-512
  - CTR DRBG KAT
  - RSA sign/verify (SHA-256 w/ 2048-bit key) KAT
  - ECDSA PCT[96] (P-256)
  - DH Primitive "Z" computation test
  - ECDH Primitive "Z" computation test
- Citrix ADC DP Cryptographic Library v3 self-tests
  - AES CBC encrypt KAT (256-bit length)
  - AES CBC decrypt KAT (256-bit length)
  - AES GCM encrypt KAT
  - AES GCM decrypt KAT
  - SHA-1, SHA-256, SHA-512 KAT
  - HMAC KAT with SHA-1, SHA-256, and SHA-512
  - Hash DRBG KAT
  - RSA sign/verify (SHA-256 w/ 2048-bit key) KAT
  - ECDSA PCT (P-256)
  - ECDH Primitive "Z" computation test

[95] KAT – Known Answer Test
[96] PCT – Pairwise Consistency Test

2.9.2 Conditional Self-Tests

The VPX performs the following conditional self-tests:

- Citrix ADC CP Cryptographic Library v3 conditional self-tests
  - CRNGT[97] for NDRNG
  - RSA PCT for sign/verify
  - RSA PCT for encrypt/decrypt
  - ECDSA PCT for sign/verify
- Citrix ADC DP Cryptographic Library v3 conditional self-tests
  - CRNGT for NDRNG

2.9.3 Critical Functions Self-Tests

VPX implements the *NIST SP 800-90A* Hash DRBG and CTR DRBG as its random number generators. The *NIST SP 800-90A* specification requires that certain critical functions be tested to ensure the security of the DRBGs. Therefore, the following power-up critical function tests are implemented by the cryptographic module for the Hash and CTR DRBGs:

- Instantiate Critical Function Test
- Generate Critical Function Test
- Reseed Critical Function Test

2.9.4 Self-Test Failures

If any of the power-up self-tests fail, the module enters a critical error state, and an error message is logged. In this state, cryptographic operations are halted, and the module inhibits all data output from the module.

If the module enters the critical error state due to a failure of the integrity test, 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 integrity test and then all subsequent power-up self-tests. If the module continues to return to a halted state, the module is considered to be malfunctioning or compromised, and Citrix Customer Support must be contacted.

If the module enters the critical error state due to a failure of any of the remaining power-up self-tests, the module will automatically reboot to clear the error state and an error message will be logged. The CO must contact Citrix Customer Support if this error occurs.

The successful completion or failure of the power-up self-tests can be verified by checking the log files. Successful completion of the Citrix ADC DP Cryptographic Library v3 self-tests is indicated by “FIPS POST Successful” in /var/log/ns.log and successful completion of the Citrix ADC CP Cryptographic Library v3 self-tests in indicated by “POST Success” in /var/log/FIPS-post.log. Failure of the Citrix ADC DP Cryptographic Library v3 self-tests is indicated by “FIPS Post Failed” in /var/log/ns.log and failure of the Citrix ADC CP Cryptographic Library v3 self-tests is indicated by “POST Failed” in /var/log/FIPS-post.log (both messages indicate a critical error state).

[97] CRNGT – Continuous Random Number Generator Test

If any of the 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
```

Once the message is displayed (and the error is logged), the module returns to an operational state. 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.

2.10 Mitigation of Other Attacks

The module's software includes several features that provide defenses against a wide range of application and web server DoS attacks. These features (including packet inspection, priority queuing, bypassing the cache, rate limiting, and packet rejection) prevent the allocation of server resources for specific connections. Additionally, the module mitigates SYN flood attacks by utilizing SYN cookies rather than maintaining half-open connections on the system memory stack. DNS DoS attacks are mitigated using parameters that protect the DNS cache memory.

The module's built-in Web App Firewall provides configurable security checks to detect and mitigate Web attacks (including attacks on operating system and web server software vulnerabilities, SQL database vulnerabilities, errors in the design and coding of web sites and web devices, and failures to secure sites that host or can access sensitive information). Web requests or responses that violate security checks are blocked or transformed (making the attack harmless). Specific attacks mitigated by the Web App Firewall include:

- HTML[98]/XML[99] Cross-Site Scripting (XSS) attacks
- HTML/XML SQL injection attacks
- HTML Cross-Site Request Forgery (CSRF) attacks
- HTML Form/hidden field and parameter manipulation
- XML DoS attacks
- Cookie or session poisoning
- Forceful browsing
- Buffer overflow attacks
- XML-based attacks using invalid or poorly formed XML requests, content injection, or inconsistencies in XML interoperability

The IP Reputation feature of the module protects against password cracking attacks (via botnets), Windows exploit attacks, and phishing proxy attacks by identifying IP addresses that are sending unwanted request and rejecting requests received from an IP with a bad reputation.

The module's DNS Security Options feature and configurable DNS parameters are used to mitigate DNS-based attacks. These attacks include random subdomain/NXDOMAIN/NODATA attacks, root referral amplification attacks, cache poisoning, and Slowloris attacks. The attacks are mitigated by preventing the insertion of corrupt data into the DNS cache, restricting access to root referrals for unrelated domains that are not configured or cached, forcing DNS transactions to use TCP instead of UDP[100] when clients send a flood of queries but cannot handle responses, and dropping DNS queries that exceed a specified length or are split into multiple packets.

[98] HTML – Hypertext Markup Language
[99] XML – Extensible Markup Language
[100] UDP – User Datagram Protocol

The module software includes defenses against TCP spoofing. TCP spoofing is mitigated by enabling configurable parameters to respond to invalid sequence numbers with a corrective acknowledgement, and/or to drop invalid SYN packets.

3. Secure Operation

The sections below describe how to place and keep the module in the FIPS-approved mode of operation. **Any operation of the module without following the guidance provided below will result in non-compliant use and is outside the scope of this Security Policy.**

3.1 Installation and Setup

The module is available as a software package that includes both the application software and the operating system. After purchasing the VPX, the installation files can be downloaded from Citrix ADC Downloads using valid credentials provided by Citrix. License entitlement(s) are sent by Citrix via email after purchase or can be accessed via the Citrix Support Portal using valid credentials. The module was tested with a Platinum edition license.

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

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

3.1.1 Installation

For detailed guidance regarding the installation of VPX, 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 VPX support matrix and usage guidelines, prerequisites for installing the VPX virtual appliance, hardware requirements for the host platforms, and VPX installation instructions. To install the required license files, the CO must follow the instructions on the Citrix ADC licensing overview webpage on Citrix's online product documentation portal.

3.1.2 General Configuration

After the VPX has been installed on a VMware ESXi 6.5 U2 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 Citrix's online product documentation portal.

3.1.3 FIPS-Approved Mode Configuration and Status

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

- Enforce strong passphrase requirements
- Replace the default TLS certificate
- Disable HTTP access to the Web GUI
- Create the KEK master key
- Disable local authentication after initial configuration

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 Secure Deployment Guide for NetScaler MPX, VPX, and SDX Appliances).

3.1.3.1 Enforce strong passphrase requirements

Passphrases are used to derive keys using PBKDF. The CO must enable strong passphrase requirements. This is accomplished with the following steps from the VPX 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**.

3.1.3.2 Replace the default TLS certificate

By default, the VPX 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 VPX:

   ```
   set ns hostName [hostname]
   ```

2. From the VPX GUI, complete the following procedure to create a Certificate Signing Request (CSR):
   - In the Configuration navigation pane, go to **Traffic Management** and click the **SSL** node.
   - In the **SSL Certificates** section, click the **Create Certificate Request** link.
   - 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 VPX 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.

3.1.3.3 Disable HTTP access to the Web GUI

To protect traffic to the administrative interface and Web GUI, the VPX must be configured to use HTTPS[101]. Once the VPX has been configured to use new TLS and SSH certificates (see section 3.1.3.1 above), the CO must disable HTTP access to the GUI management interface with the following CLI command:

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

3.1.3.4 Create the KEK Master Key

The KEK master key is used to encrypt passphrases and other sensitive information. To prevent the default KEK from being used, the CO must create a new KEK. To create the KEK, the CO must run the following CLI command:

```
create system kek
```

When prompted, the CO shall enter a strong passphrase (the KEK will be derived from this).

3.1.3.5 Disable local authentication

The `nsroot` account is a default account with root CLI access (superuser) privileges that is required for initial configuration. After initial configuration, local system authentication must be disabled 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 to disable local authentication:

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

3.1.3.6 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 Configuring external user authentication webpage found on the Citrix online product documentation portal to configure external system authentication. The CO must ensure the following before enabling external authentication:

- Ensure a secure connection is established with the external authentication service.
- Ensure shell access is disabled for all profiles on the external authentication service.

[101] HTTPS – Hypertext Transfer Protocol Secure

3.2 Crypto Officer Guidance

The CO is responsible for ensuring that the module is operating in the FIPS-Approved mode of operation. When configured and operated according to the guidance in this Security Policy (including the previous instructions in section 3.1.3), the module only runs in the FIPS-Approved mode of operation.

3.2.1 Management

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 sections 3.1.3, 3.2, and 3.4 for guidance that the Crypto Officer must follow to ensure that the module is operating in a FIPS-Approved manner.

3.2.2 On-Demand Self-Tests

Although power-up 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, using the reset button on the platform (if applicable), the `reboot` CLI command, the `reboot` API method, or via the Web GUI by navigating to **Configuration > System > System Information** and clicking the **Reboot** button.

3.2.3 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. All ephemeral keys are zeroized on module reboot, power removal, or session termination.

The KEK is stored as plaintext in non-volatile memory. The zeroization of the KEK renders all passphrases and passwords stored in the non-volatile memory unrecoverable, effectively zeroizing them. The KEK is 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
```

3.2.4 Monitoring Status

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 logged in via the CLI can view the operational status 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[102] and serial number.

The RESTful Nitro API can be used with the GET method to view the operational status by using the following URLs:

- `https://<netscaler-ip-address>/nitro/v3/config/nshardware`
- `https://<netscaler-ip-address>/nitro/v3/config/nsversion`

An operator logged in via the Web GUI can also view the operational status 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.

3.3 User Guidance

The User role does not have the ability to configure sensitive information on the module. The User must be diligent to select strong passwords and must not reveal their password to anyone. Additionally, User role operators should be careful to protect any secret or private keys in their possession.

3.4 Additional Guidance and Usage Policies

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

- All private keys (except for SSH private keys) must be stored as PEM files in encrypted format using a FIPS-Approved encryption algorithm listed in Table 2 or Table 3.

- Upon successful bootup of the module, the VPX is configured by default to use only *NIST SP 800-52rev2* 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 Rev2* as guidance.

- The VPX 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.

[102] ID – Identifier

- The VPX supports Kerberos traffic management. This feature must be configured to use PKINIT[103]. Once configured the module restricts Kerberos to using only FIPS-Approved ciphersuites. For details on configuring the protocol to use PKINIT refer to Citrix ADC 12.1 – An Overview of Citrix ADC Kerberos SSO.

- The VPX supports client-side Kerberos. The CO must configure a password or keytab file to use this feature. VPX will only accept packages using FIPS 140-2 encryption. For details on configuring client-side Kerberos refer to Citrix ADC 12.1 – Configuring Kerberos Authentication on the Citrix ADC Appliance.

- The VPX supports clustering. A VPX may either be 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 12.1 – Clustering.

- The CO must ensure that communication between the module and the external authentication service is secure.

- The CO must ensure that shell access is disabled for all profiles on the external authentication service.

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

- Module operators shall ensure that only those algorithms and key sizes mentioned in Section 2.7 (Cryptographic Key Management) of this document are in use to remain in the FIPS-Approved mode of operation.

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

- The module operator shall ensure that the number of encryptions performed by the TDES key is performed no more than 2^{16} times by periodically rebooting the module.

- The VPX 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 because they are not being used for production purposes or for true protection of data,. The CO must ensure that all certificates are signed using a trusted CA and not by a self-signed certificate.

3.5 Non-FIPS-Approved Mode

When initialized, configured, and operated according to the guidance in this Security Policy, the module does not support a non-FIPS-Approved mode of operation.

[103] PKINIT – Public Key Cryptography for Initial Authentication in Kerberos – details for PKINIT in the Kerberos protocol are in RFC #4556

4. Acronyms

Table 12 provides definitions for the acronyms used in this document.

Table 12 – Acronyms

Acronym	Definition
AAA	Authentication, Authorization, Accounting
ADC	Application Delivery Controller
AES	Advanced Encryption Standard
API	Application Programming Interface
AWS	Amazon Web Services
CA	Certificate Authority
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CCCS	Canadian Centre for Cyber Security
CKG	Cryptographic Key Generation
CLI	Command Line Interface
CMVP	Cryptographic Module Validation Program
CO	Crypto Officer
CPU	Central Processing Unit
CRNGT	Continuous Random Number Generator Test
CSP	Critical Security Parameter
CSR	Certificate Signing Request
CSRF	Cross-Site Request Forgery
CTR	Counter
CVL	Component Validation List
DES	Data Encryption Standard
DFA	Delegated Form Authentication
DH	Diffie-Hellman
DNS	Domain Name System
DoS	Denial-of-Service
DRBG	Deterministic Random Bit Generator
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

Acronym	Definition
FIPS	Federal Information Processing Standard
FOM	FIPS Object Module
GCM	Galois/Counter Mode
GCP	Google Cloud Platform
GHz	Gigahertz
GSLB	Global Server Load Balancing
GUI	Graphical User Interface
HA	High Availability
HMAC	(keyed-) Hash Message Authentication Code
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ID	Identifier
IDP	Identity Provider
IKE	Internet Key Exchange
IP	Internet Protocol
IPsec	Internet Protocol Security
IV	Initialization Vector
KAT	Known Answer Test
KDF	Key Derivation Function
KEK	Key Encryption Key
KPG	Key Pair Generation
KSK	Key Signing Key
KVM	Kernel-based Virtual Machine
L4-L7	Layer 4 through Layer 7
LDAP	Lightweight Directory Access Protocol
LTS	Long Term Support
LTSR	Long Term Service Release
MD5	Message Digest 5
N/A	Not Applicable
NDRNG	Non-Deterministic Random Number Generator
NIST	National Institute of Standards and Technology
NTP	Network Time Protocol
OID	Object Identifier
OS	Operating System
PBKDF	Password-based Key Derivation Function

Acronym	Definition
PCT	Pairwise Consistency Test
PEM	Privacy-Enhanced Mail
PKCS	Public Key Cryptography Standard
PSK	Pre-shared Key
RAM	Random Access Memory
RBA	Role-Based Authentication
RDP	Remote Desktop Protocol
REST	Representational State Transfer
RFC	Request For Comment
RSA	Rivest Shamir Adleman
SAML	Security Assertion Markup Language
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SNMP	Simple Network Management Protocol
SP	Special Publication
SQL	Structured Query Language
SSH	Secure Shell
SSL	Secure Socket Layer
TCP	Transmission Control Protocol
TLS	Transport Layer Security
U2	Update 2
UDP	User Datagram Protocol
URL	Uniform Resource Locator
U.S.	United States
XML	Extensible Markup Language
XSS	Cross-Site Scripting
ZSK	Zone Signing Key

Prepared by:
Corsec Security, Inc.

13921 Park Center Road, Suite 460
Herndon, VA 20171
United States of America

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