Common Criteria Security Target

For

NetScaler Platinum Edition Load Balancer Version 10.5

Version 1-0   13 October 2015

Summary of Amendments

Version 1-0 13 October 2015

Initial document release.

0. Preface

0.1 Objectives of Document

This document presents the Common Criteria (CC) Security Target (ST) to express the security and evaluation requirements for the Citrix NetScaler Platinum Edition Load Balancer Version 10.5 product. The product claims conformance with version 1.1 of the protection profile 'Security Requirements for Network Devices' [NDPP].

The NetScaler product is designed and manufactured by Citrix Systems Inc. (http://www.citrix.com/). The Sponsor and Developer for the evaluation is Citrix Systems Inc.

0.2 Scope of Document

The scope of the Security Target within the development and evaluation process is described in the Common Criteria for Information Technology Security Evaluation [CC]. In particular, a Security Target defines the IT security requirements of an identified TOE and specifies the functional and assurance security measures offered by that TOE to meet stated requirements [CC1, Section C.1].

Security Functional Requirements (SFRs), as defined in [CC2], are the basis for the TOE IT security functional requirements expressed in this Security Target. These requirements describe the desired security behaviour expected of a TOE and are intended to meet the security objectives as stated in this Security Target. Security Functional Requirements express security requirements intended to counter threats in the assumed operating environment of the TOE, and cover any identified organisational security policies and assumptions.

0.3 Intended Readership

The target audience of this ST are consumers, developers, evaluators and certifiers of the TOE, additional information can be found in [CC1, Section 6.2].

0.4 Related Documents

Common Criteria[1]

[CC1] Common Criteria for Information Technology Security Evaluation,
Part 1: Introduction and General Model,
CCMB-2012-09-001, Version 3.1 Revision 4, September 2012.

[1] For details see http://www.commoncriteriaportal.org/

[CC2]	Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components, CCMB-2012-09-002, Version 3.1 Revision 4, September 2012.
[CC3]	Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Components, CCMB-2012-09-003, Version 3.1 Revision 4, September 2012.
[CEM]	Common Methodology for Information Technology Security Evaluation, Evaluation Methodology, CCMB-2012-09-004, Version 3.1, Revision 4, September 2012.

Other documentation

[CCECG]	Common Criteria Evaluated Configuration Guide for NetScaler 10 Platinum Edition, Document code: October 13, 2015 17:56:12 Version 3.6.
[FIPS-card]	CMVP FIPS 140-2 certificate #2316: NITROX XL 1600-NFBE HSM Family (Hardware Versions: P/Ns CN1610-NFBE1-3.0-FW-2.2-G, CN1620-NFBE1-3.0-FW-2.2-G, CN1620-NFBE3-3.0-FW-2.2-G, CN1610-NFBE1-2.0-FW-2.2-G, CN1620-NFBE1-2.0-FW-2.2-G and CN1620-NFBE3-2.0-FW-2.2-G; Firmware Version: CN16XX-NFBE-FW-2.2-130009)
[FIPS-ossl]	CMVP FIPS 140-2 certificate #1747: OpenSSL FIPS Object Module (Software Version: 2.0, 2.0.1, 2.0.2, 2.0.3, 2.0.4, 2.0.5, 2.0.6, 2.0.7, 2.0.8 or 2.0.9)
[NDPP]	Protection Profile for Network Devices, version 1.1, 8 June 2012
[NDPP-err3]	Security Requirements for Network Devices - Errata #3, dated 3 November 2014
[SecP-card]	FIPS 140-2 Level 3 Security Policy – NITROX XL 1600-NFBE HSM Family, version 2.6, 7/6/2015
[SecP-ossl]	OpenSSL FIPS 140-2 Security Policy, version 2.0.8, October 4, 2014
[SP800-56B]	NIST Special Publication 800-56B - Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography, August 2009
[UG-ossl]	User Guide for the OpenSSL FIPS Object Module v2.0 (including v2.0.1, v2.0.2, v2.0.3, v2.0.4, v2.0.5, v2.0.6, v2.0.7, v2.0.8, v2.09, 2.0.10), September 5 2015

0.5 Significant Assumptions

None.

0.6 Outstanding Issues

0.7 Glossary

Term	Meaning
Assurance	Grounds for confidence that a TOE meets the SFRs [CC1].
CC	Common Criteria
CLI	Command Line Interface
CM	Configuration Management
DNS	Domain Name System
EAL	Evaluation Assurance Level
FIPS	Federal Information Processing Standard
FTP	File Transfer Protocol
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
HTTPS	Secure Hypertext Transfer Protocol
IP	Internet Protocol
KAT	Known Answer Test
LCD	Liquid Crystal Display
LDAP	Lightweight Directory Access Protocol
NIC	Network Interface Controller
NTP	Network Time Protocol
NSIP	The IP address of the NetScaler appliance itself
Nsroot	A superuser account that provides complete access to all features of the NetScaler appliance. Access to the nsroot account is therefore required to be constrained as described in [CCECG].
PAM	Pluggable Authentication Module
PCT	Pairwise Consistency Test
PP	Protection Profile
RADIUS	Remote Authentication Dial-In User Service
SAR	Security Assurance Requirement
SCP	Secure Copy
SFP	Security Function Policy
SFR	Security Functional Requirement
SFTP	Secure Shell File Transfer Protocol
SSH	Secure Shell
SSL	Secure Sockets Layer
ST	Security Target

Term	Meaning
System User	A user who can log on to the NetScaler CLI and, subject to the command policies bound to the user account, then acts as an Authorized Administrator. (Note that this type of NetScaler user is distinct from what is known in NetScaler terminology as a ‘AAA User’ or a ‘VPN User’.)
Target of Evaluation	A set of software, firmware and/or hardware possibly accompanied by guidance. [CC1]
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TOE	Target of Evaluation
TOE Security Functionality	A set consisting of all hardware, software, and firmware of the TOE that must be relied upon for the correct enforcement of the SFRs. [CC1]
TSF	TOE Security Functionality
UDP	User Datagram Protocol
URL	Uniform Resource Locator
VPN	Virtual Private Network
WAN	Wide Area Network

See [CC1] for other Common Criteria abbreviations and terminology.

Contents

Figures / Tables

1. ST Introduction

1.1 ST and TOE Reference Identification

TOE Reference:	NetScaler Platinum Edition Load Balancer Version 10.5-53.22 (nCore)
ST Reference:	CN12-ST-0001
ST Version:	1-0
ST Date:	13 October 2015
Assurance Level:	Defined by [NDPP, 4.3] – see section 5.3 below
ST Author:	SiVenture

1.2 TOE Overview

1.2.1 NetScaler Product Overview

The TOE described in this Security Target is the Citrix NetScaler Platinum Edition Load Balancer Version 10.5, with software build 53.22 (nCore) (abbreviated in this document to "NetScaler"). The TOE comprises NetScaler running on the following hardware appliances (the difference between the appliances is the hardware performance):

- MPX 9700-FIPS
- MPX 10500-FIPS
- MPX 12500-FIPS
- MPX 15500-FIPS.

The TOE type is a 'network device' in the sense of [NDPP, 1.1]: "a device composed of hardware and software that is connected to the network and has an infrastructure role in the overall enterprise".

All appliances provide a serial port to connect a console directly to the appliance for management, and a Liquid Crystal Display (LCD) on the front of each appliance displays real-time statistics, diagnostics, and alerts. All platforms include a FIPS 140-2 Level 3 validated crypto card [FIPS-card], and FIPS 140-2 Level 1 validated version of OpenSSL [FIPS-ossl].

NetScaler is an application performance accelerator incorporating a Secure Sockets Layer (SSL/TLS) Virtual Private Network (VPN). The scope of this evaluation includes the TOE when operating as a dedicated self-contained appliance (i.e. running on dedicated hardware supplied as part of the TOE). Figure 1 below shows the details of the deployment configuration of the TOE:

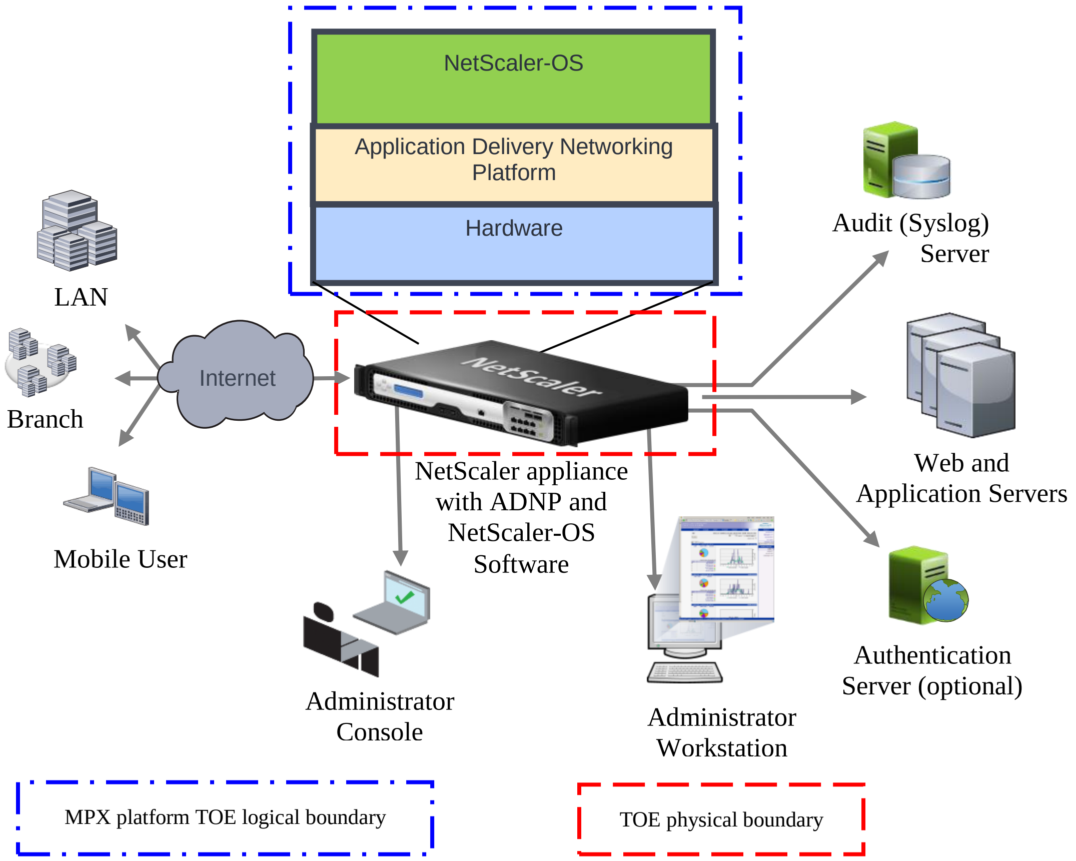
NetScaler-OS
Application Delivery Networking Platform
Hardware
LAN
Branch
Internet
Mobile User
Audit (Syslog) Server
Web and Application Servers
Authentication Server (optional)
NetScaler appliance with ADNP and NetScaler-OS Software
Administrator Console
Administrator Workstation
MPX platform TOE logical boundary
TOE physical boundary

Figure 1: Deployment Configuration of the Product and TOE Boundaries

The NetScaler appliance incorporates the NetScaler-OS that provides secure access to web-based applications from an external network. The NetScaler-OS runs on top of the Application Delivery Networking Platform on a dedicated MPX-FIPS hardware appliance (the specific appliances within the scope of this Security Target are listed at the start of section 1.2.1). These elements are described in more detail in the sections below.

TOE Administrators can access the TOE locally through a direct serial connection to an Administrator Console, or by remote access from an Administrator Workstation through Secure Shell (SSH) (see section 1.2.2.2) over a TLS VPN (TLS) tunnel. Both types of connection give the administrator access to the Command Line Interface (CLI).

1.2.2 NetScaler-OS

The security functionality in this ST are implemented by two main parts of the NetScaler-OS: the Load Balancer and the NetScaler Gateway. These are described in the subsections below.

1.2.2.1 Load Balancer

The Load Balancer component of the NetScaler-OS manages the connections between clients and servers. Clients establish a connection with the NetScaler rather than directly with a server. When the NetScaler receives an application request from a client, it establishes a connection with the appropriate application server. This allows the Load Balancer to sort and prioritize application requests from multiple clients and requires only a single connection on the application server to handle requests from multiple clients.

The Load Balancer allows the definition of Load Balancing virtual servers (vserver). Each Load Balancing vserver consists of an IP address, port number, and protocol. A Load Balancing vserver accepts incoming traffic destined for its particular address-port-protocol combination and is mapped to one or more services running on physical servers in a server farm. Clients connect to the Load Balancing virtual server, which directs each request to a physical server. Load Balancing provides methods for each Load Balancing vserver to choose the physical server with the smallest load to direct traffic to.

1.2.2.2 NetScaler Gateway

The NetScaler Gateway component of the NetScaler-OS provides a TLS VPN connection that is used as a tunnel to enable remote administrator connections to be made.

As described in [CCECG, *Chapter 3, Initial Access and Configuration]***,** the IP address of the NS appliance itself (known as the 'NSIP') should not be made publicly routable, in order to protect against unauthorised attempts to access the appliance and its configuration. Therefore in a deployment configuration a remote administrator first opens a TLS VPN connection to the NetScaler appliance to a publicly accessible VPN vserver IP address (see [CCECG, *Chapter 3, Accessing the NetScaler Remotely.)*. When connecting to the vserver the administrator is authenticated as an authorised VPN user by means of the configured authentication mechanism. After successful authentication, a TLS tunnel is established through which the administrator can then address the NSIP and open an SSH connection to it in order to access the CLI for management operations. The TLS and SSH connections for this functionality use the cryptographic services that are subject to the requirements in section 5.2.2, and are therefore covered in the scope of the evaluation against this Security Target.

1.2.3 Application Delivery Networking Platform

The Application Delivery Networking Platform is a highly-specialized kernel and packet processing engine. It coordinates the operations of the other software and controls the network interfaces, memory management, and system timing. By interfacing closely with the network interface drivers, the Application Delivery Networking Platform is able to assure that critical applications receive the highest priority and are not pre-empted by lower-priority operations.

1.2.4 Guidance Documentation

The following guide is required reading and forms part of the TOE:

- [CCECG]

1.2.5 Required non-TOE hardware and software

No additional non-TOE hardware or software is required.

1.2.6 TOE Environment

The following hardware and software are part of the TOE environment:

- Administrator console and workstation for management of the TOE
- Application server(s)
- Audit (Syslog) server
- VPN client(s)
- Networks (including the Internet and the Enterprise Network)
- Authentication server (LDAP, RADIUS) (optional).

The TOE is intended to be deployed in a physically secure cabinet, room or data center with the appropriate level of physical access control and physical protection (e.g. fire control, locks, alarms, etc.). The TOE is intended to be managed by administrators operating under a consistent security policy.

The TOE is managed through a CLI[2] (and the CLI therefore forms part of the TOE) Administrators must access this interface from a trusted workstation that supports SSH or from a workstation on a direct serial connection.

1.3 TOE Boundaries

The TOE boundaries are shown in Figure 1 in section 1.2.1 above.

1.3.1 Physical Scope

Figure 1 shows the physical scope and the physical boundary of the TOE within the context of a deployment of the NetScaler product: the physical boundary is defined by the appliance itself. The appliance hardware includes a Cavium crypto card that is separately validated against FIPS140-2 (see [FIPS-card]).

The hardware platforms covered by this Security Target are listed in section 1.2.1.

1.3.2 Logical Scope

The logical boundary of the TOE is shown in Figure 1.

The TOE is composed of the NetScaler-OS, the Application Delivery Networking Platform, and the hardware appliance (see list of platforms in section 1.2.1 above). These logical

[2] Note that management via the GUI, Dashboard, Command Center application and NetScaler Nitro API are not included in the scope of the TOE. The evaluated configuration disables the use of these features as described in [CCECG].

components run directly on the NetScaler appliance hardware. The NetScaler-OS incorporates a version of OpenSSL that is separately validated against FIPS140-2 (see [FIPS-ossl]).

The SFRs implemented by the TOE are grouped under the following Security Function Classes:

FAU	Security Audit
FCS	Cryptographic Support
FDP	User Data Protection
FIA	Identification and Authentication
FMT	Security Management
FPT	Protection of the TSF
FTA	TOE Access
FTP	Trusted Path/Channels.

These functions are described in detail in sections 5.2 and 6.1.

1.3.3 Out-of-Scope Features and Functionality

Hardware and software located in the TOE environment (see section 1.2.6) are not included in the scope of evaluations against this Security Target.

Only security functionality specified in the SFRs in section 5.2 (and the corresponding security functions in section 6.1) is covered by the scope of evaluation against this Security Target. Other product features or functionality are not included in the TOE. In particular the following NetScaler features are excluded from the TOE and the scope of this Security Target:

- IPv6
- Web Logging
- Application Firewall
- Global Server Load Balancing (GSLB)
- AAA-TM Authentication
- External authentication methods: Kerberos, TACACS+, Radius, SAML[3]
- Responder
- Rewrite (URL Transformation)
- DNS
- Edgesight

[3] The TOE may be configured to use external authentication servers, but the protocols and authentication servers are outside the scope of this Security Target.

- Layer 3 Routing
- NetScaler GUI, Dashboard, Command Center application and NetScaler Nitro API[4]
- Vpath
- RISE
- High Availability
- CloudBridge
- CallHome
- Integrated Disk Caching
- General TLS VPN functionality[5]
- Clientless VPN functionality
- Use of superuser privileges except as described in [CCECG][6].

The evaluated configuration does not include connection of the TOE to an NTP server.

[4] These are alternative methods of managing NetScaler. However, only the CLI method of management is included in the evaluated configuration.

[5] The use of TLS VPN functionality (and its underlying cryptographic function) is included in scope as part of creating a remote management connection as described in section 1.2.2.2. However, this Security Target does not cover more general use of the TLS VPN functionality.

[6] The TOE uses the SysAdmin role and, other subsets of the privileges of this role, to enable day-to-day administrative operations as described in [CCECG, Chapter 4, *SysAdmin Capabilities and Entitlements*].

2. CC Conformance

As defined by the references [CC1], [CC2] and [CC3], this TOE conforms to the requirements of Common Criteria v3.1, Revision 4. The methodology applied for the evaluation is defined in [NDPP], [NDPP-err3] and [CEM].

The TOE is Part 2 extended, Part 3 conformant.

This ST claims conformance to version 1.1 of ‘Protection Profile for Network Devices’ [NDPP], including errata#3 [NDPP-err3]. This includes a claim of ‘Exact Compliance’ as defined in [NDPP-err3, 2.1]. NetScaler comprises software running on a hardware appliance and is a network device in the sense of [NDPP, 1.1]: “a device composed of hardware and software that is connected to the network and has an infrastructure role in the overall enterprise” (cf. section 1.2.1). The Security Problem Definition, Security Objectives and Security Requirements in this ST are adopted directly from [NDPP] and [NDPP-err3], The Security Functional Requirements in section 5.2 include FCS_TLS_EXT.1, FCS_SSH_EXT.1, and FCS_HTTPS_EXT.1 from the ‘Additional Requirements in [NDPP] (with revisions to FCS_TLS_EXT.1 and FCS_SSH_EXT.1 as in [NDPP-err3]).

3. Security Problem Definition

The Security Problem Definition in this Security Target is adopted from [NDPP, 2].

3.1 Threats

This section identifies the threats to the IT assets against which protection is required by the TOE or by the security environment. Threats that the TOE addresses are adopted from [NDPP, 2.1-2.6], and are summarised below using a copy of table 4 from [NDPP, Annex A].

Threat Name	Threat Definition
T.ADMIN_ERROR	An administrator may unintentionally install or configure the TOE incorrectly, resulting in ineffective security mechanisms.
T.TSF_FAILURE	Security mechanisms of the TOE may fail, leading to a compromise of the TSF.
T.UNDETECTED_ACTIONS	Malicious remote users or external IT entities may take actions that adversely affect the security of the TOE. These actions may remain undetected and thus their effects cannot be effectively mitigated.
T.UNAUTHORIZED_ACCESS	A user may gain unauthorized access to the TOE data and TOE executable code. A malicious user, process, or external IT entity may masquerade as an authorized entity in order to gain unauthorized access to data or TOE resources. A malicious user, process, or external IT entity may misrepresent itself as the TOE to obtain identification and authentication data.
T.UNAUTHORIZED_UPDATE	A malicious party attempts to supply the end user with an update to the product that may compromise the security features of the TOE.
T.USER_DATA_REUSE	User data may be inadvertently sent to a destination not intended by the original sender.

Table 1: Threats addressed by the TOE

3.2 Organisational Security Policies

Organisational Security Policies for the TOE are adopted from [NDPP, Annex A], and are summarised below using a copy of table 5 from [NDPP, Annex A].

Policy Name	Policy Definition
P.ACCESS_BANNER	The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the TOE.

Table 2: Organisational Security Policy for the TOE

3.3 Assumptions

The Assumptions that apply to the TOE are adopted from [NDPP, Annex A], and are summarised below using a copy of table 3 from [NDPP, Annex A]. These Assumptions represent conditions that are assumed to exist in the TOE's Operational Environment.

Assumption Name	Assumption Definition
A.NO_GENERAL_PURPOSE	It is assumed that there are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.
A.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the environment.
A.TRUSTED_ADMIN	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.

Table 3: Assumptions applicable to the TOE

4. Security Objectives

4.1 Security Objectives for the TOE

The Security Objectives for NetScaler are adopted from [NDPP, 3.1-3.6], and are summarised below using a copy of table 6 from [NDPP, Annex A].

TOE Security Objective	TOE Security Objective Definition
O.PROTECTED_COMMUNICATIONS	The TOE will provide protected communication channels for administrators, other parts of a distributed TOE, and authorized IT entities.
O.VERIFIABLE_UPDATES	The TOE will provide the capability to help ensure that any updates to the TOE can be verified by the administrator to be unaltered and (optionally) from a trusted source.
O.SYSTEM_MONITORING	The TOE will provide the capability to generate audit data and send those data to an external IT entity.
O.DISPLAY_BANNER	The TOE will display an advisory warning regarding use of the TOE.
O.TOE_ADMINISTRATION	The TOE will provide mechanisms to ensure that only administrators are able to log in and configure the TOE, and provide protections for logged-in administrators.
O.RESIDUAL_INFORMATION_CLEARING	The TOE will ensure that any data contained in a protected resource is not available when the resource is reallocated.
O.SESSION_LOCK	The TOE shall provide mechanisms that mitigate the risk of unattended sessions being hijacked.
O.TSF_SELF_TEST	The TOE will provide the capability to test some subset of its security functionality to ensure it is operating properly.

Table 4: Security Objectives for the TOE

4.2 Security Objectives for the Operational Environment

The Security Objectives that are required to be met by the NetScaler operational environment are adopted from [NDPP, Annex A], and are summarised below using a copy of table 7 from [NDPP, Annex A].

Security Objective	Security Objective Definition
OE.NO_GENERAL_PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.
OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
OE.TRUSTED_ADMIN	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.

Table 5: Security Objectives for the Operational Environment

4.3 Security Objectives Rationale

As with the Threats, Assumptions, Organisational Security Policies, and Security Objectives, the Security Objectives Rationale is also adopted as described in [NDPP, 3] and [NDPP, Annex A].

5. IT Security Requirements

5.1 Conventions

The conventions used in SFRs adopted from [NDPP, 4.1] and are as follows:

- Assignment: Indicated with *italicized* text;
- Refinement made by PP author: Indicated with **bold text** and ~~strikethroughs~~, if necessary;
- Selection: Indicated with underlined text;
- Assignment within a Selection: Indicated with *italicized and underlined text*;
- Iteration: Indicated by appending the iteration number in parenthesis, e.g., (1), (2), (3) and/or by adding a string starting with "/".

Extended SFRs are identified by having a label 'EXT' at the end of the SFR name. Note that since no dependencies are stated in [NDPP] for the extended SFRs, it is assumed that none apply.

[NDPP] itself makes some selection and assignments to SFRs, and this Security Target makes further selections and assignments based on the [NDPP] text. The selections and assignments made in the Security Target are indicated by using the above conventions and by also leaving the relevant text inside square brackets (note that in a few cases square brackets around completed operations are included in the Security Target because they are present around the same completed operations in [NDPP] or [NDPP-err3]).

5.2 Security Functional Requirements

The Security Functional Requirements for the TOE are adopted from [NDPP, 4.2], with selections and assignments made for the NetScaler TOE. These NetScaler-specific SFRs are stated in the sections below. Application Notes from [NDPP] are included only where they clarify the meaning of SFRs (and not where they give instructions to ST authors). For details of Assurance Activities, please refer to [NDPP].

All extended components are adopted from [NDPP] and as such are taken to be defined by [NDPP] and no separate definition statement is given in the Security Target.

5.2.1 Security Audit (FAU)

FAU_GEN.1	**Audit Data Generation**

Dependencies: FPT_STM.1 Reliable time stamps

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shut-down of the audit functions;

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

c) *All administrative actions*;

d) [*Specifically defined auditable events listed in* ***Table 8***].

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [*information specified in column three of* ***Table 8***].

FAU_GEN.2	**User identity association**

Dependencies: FAU_GEN.1 Audit data generation
FIA_UID.1 Timing of identification

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

FAU_STG_EXT.1	**External Audit Trail Storage**

Dependencies: None (assumed)

FAU_STG_EXT.1.1 The TSF shall be able to [transmit the generated audit data to an external IT entity] using a trusted channel implementing the [SSH] protocol.

5.2.2 Cryptographic Support (FCS)

FCS_CKM.1	**Cryptographic Key Generation (for asymmetric keys)**

Dependencies: [FCS_CKM.2 Cryptographic key distribution, or
FCS_COP.1 Cryptographic operation]
FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1 Refinement: The TSF shall generate **asymmetric** cryptographic keys **used for key establishment** in accordance with

[*NIST Special Publication 800-56B, "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography" for RSA-based key establishment schemes*]

and specified cryptographic key sizes *equivalent to, or greater than, a symmetric key strength of 112 bits*.

FCS_CKM_EXT.4	**Cryptographic Key Zeroization**

Dependencies: None (assumed)

FCS_CKM_EXT.4.1 The TSF shall zeroize all plaintext secret and private cryptographic keys and CSPs when no longer required.

FCS_COP.1(1)/HW	**Cryptographic Operation (for data encryption/decryption)**

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security
attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1(1)/HW Refinement: The TSF shall perform [*encryption and decryption*] in accordance with a specified cryptographic algorithm [*AES operating in* [CBC]] and cryptographic key sizes 128-bits and 256-bits that meets the following:

- **FIPS PUB 197, "Advanced Encryption Standard (AES)"**
- **[NIST SP 800-38A]**

ST Application Note: this iteration of FCS_COP.1(1) deals with cryptographic operations carried out in the FIPS 140 card that is part of the NetScaler hardware appliance. These cryptographic operations support TLS channels (apart from LDAP/S, which uses FCS_COP.1(1)/SW).

FCS_COP.1(1)/SW	**Cryptographic Operation (for data encryption/decryption)**

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security
attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1(1)/SW Refinement: The TSF shall perform [*encryption and decryption*] in accordance with a specified cryptographic algorithm [*AES operating in* [CBC]] and cryptographic key sizes 128-bits and 256-bits that meets the following:

- **FIPS PUB 197, "Advanced Encryption Standard (AES)"**
- **[NIST SP 800-38A]**

ST Application Note: this iteration of FCS_COP.1(1) deals with cryptographic operations carried out in the OpenSSL-FIPS software that is part of the NetScaler-OS software. These cryptographic operations support SSH and LDAP/S connections, and derivation of key encryption keys to protect other critical security parameters.

FCS_COP.1(2)/HW	**Cryptographic Operation (for cryptographic signature)**

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security
attributes, or
FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1(2)/HW Refinement: The TSF shall perform **cryptographic signature services** in accordance with a ***[RSA Digital Signature Algorithm (rDSA) with a key size (modulus) of 2048 bits or greater]*** that meets the following:

- **[FIPS PUB 186-2 or FIPS PUB 186-3, "Digital Signature Standard"].**

ST Application Note: this iteration of FCS_COP.1(2) deals with cryptographic operations carried out in the FIPS 140 card that is part of the NetScaler hardware appliance. These cryptographic operations support TLS connections (apart from LDAP/S, which uses FCS_COP.1(2)/SW). Although [NDPP-err3] does not mark the choice of FIPS PUB 186-2 or 186-3 as a selection to be made, the TOE in fact follows FIPS PUB 186-2 as noted in section 6.1.2.

FCS_COP.1(2)/SW	Cryptographic Operation (for cryptographic signature)

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security
attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1(2)/SW Refinement: The TSF shall perform **cryptographic signature services** in accordance with a ***[RSA Digital Signature Algorithm (rDSA) with a key size (modulus) of 2048 bits or greater]*** that meets the following:

- **[FIPS PUB 186-2 or FIPS PUB 186-3, "Digital Signature Standard"].**

ST Application Note: this iteration of FCS_COP.1(2) deals with cryptographic operations carried out in the OpenSSL software that is part of the NetScaler-OS software. These cryptographic operations support SSH and LDAP/S connections. Although [NDPP-err3] does not mark the choice of FIPS PUB 186-2 or 186-3 as a selection to be made, the TOE in fact follows FIPS PUB 186-2 as noted in section 6.1.2.

FCS_COP.1(3)	Cryptographic Operation (for cryptographic hashing)

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security
attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1(3) Refinement: The TSF shall perform [*cryptographic hashing services*] in accordance with a specified cryptographic algorithm [**SHA-1, SHA-256, SHA-512] and message digest sizes** [**160, 256, 512**] **bits** that meet the following: *FIPS Pub 180-3, "Secure Hash Standard."*

ST Application Note: this iteration of FCS_COP.1(3) deals with cryptographic operations carried out in the OpenSSL software that is part of the NetScaler-OS software. The hashes used for SSH are noted in FCS_SSH_EXT.1.6; SHA-256 is also used for hashes on software updates (cf. FPT_TUD_EXT.1 in section 5.2.6).

FCS_COP.1(4)/HW Cryptographic Operation (for keyed-hash message authentication)

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security
attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1(4)/HW Refinement: The TSF shall perform [*keyed-hash message authentication*] in accordance with a specified cryptographic algorithm HMAC-[**SHA-1**]**, key size [160 bits], and message digest sizes** [**160**] **bits** that meet the following: *FIPS Pub 198-1, "The Keyed-Hash Message Authentication Code, and FIPS Pub 180-3, "Secure Hash Standard."*

ST Application Note: this iteration of FCS_COP.1(4) deals with cryptographic operations carried out in the FIPS 140 card that is part of the NetScaler hardware appliance. These cryptographic operations support TLS connections (apart from LDAP/S, which uses FCS_COP.1(4)/SW).

FCS_COP.1(4)/SW Cryptographic Operation (for keyed-hash message authentication)

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security
attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1(4)/SW Refinement: The TSF shall perform [*keyed-hash message authentication*] in accordance with a specified cryptographic algorithm HMAC-[**SHA-1, SHA-256, SHA-512**]**, key size [***160, 256, 512 bits***], and message digest sizes** [**160, 256, 512**] **bits** that meet the following: *FIPS Pub 198-1, "The Keyed-Hash Message Authentication Code, and FIPS Pub 180-3, "Secure Hash Standard."*

ST Application Note: this iteration of FCS_COP.1(4) deals with cryptographic operations carried out in the OpenSSL software that is part of the NetScaler-OS software. These cryptographic operations support SSH and LDAP/S connections, and derivation of key encryption keys to protect other critical security parameters.

FCS_RBG_EXT.1/HW Cryptographic Operation (Random Bit Generation)

Dependencies: None (assumed)

FCS_RBG_EXT.1.1/HW The TSF shall perform all random bit generation (RBG) services in accordance with [FIPS Pub 140-2 Annex C: X9.31 Appendix 2.4 using AES] seeded by an entropy source that accumulated entropy from [a TSF-hardware-based noise source].

FCS_RBG_EXT.1.2/HW The deterministic RBG shall be seeded with a minimum of [256 bits] of entropy at least equal to the greatest security strength of the keys and hashes that it will generate.

ST Application Note: this iteration of FCS_RBG_EXT.1 deals with random bits generated in the FIPS 140 card that is part of the NetScaler hardware appliance. These random numbers support TLS connections (apart from LDAP/S, which uses FCS_RBG_EXT.1/SW), and protection of other critical security parameters.

FCS_RBG_EXT.1/SW Cryptographic Operation (Random Bit Generation)

Dependencies: None (assumed)

FCS_RBG_EXT.1.1/SW The TSF shall perform all random bit generation (RBG) services in accordance with [FIPS Pub 140-2 Annex C: X9.31 Appendix 2.4 using AES] seeded by an entropy source that accumulated entropy from [a software-based noise source].

FCS_RBG_EXT.1.2/SW The deterministic RBG shall be seeded with a minimum of [256 bits] of entropy at least equal to the greatest security strength of the keys and hashes that it will generate.

ST Application Note: this iteration of FCS_RBG_EXT.1 deals with random bits generated in the OpenSSL software that is part of the NetScaler-OS software. These random numbers support SSH and LDAP/S connections, which uses FCS_RBG_EXT.1/SW).

FCS_TLS_EXT.1 Explicit: TLS

Dependencies: None (assumed)

FCS_TLS_EXT.1.1 The TSF shall implement one or more of the following protocols [TLS 1.0 (RFC 2246), TLS 1.1 (RFC 4346), TLS 1.2 (RFC 5246)] supporting the following ciphersuites:

Mandatory Ciphersuites:
TLS_RSA_WITH_AES_128_CBC_SHA

Optional Ciphersuites:
[*TLS_RSA_WITH_AES_256_CBC_SHA*] .

ST Application Note: for details of NetScaler compliance with RFCs, see section 6.2.

FCS_SSH_EXT.1 Explicit: SSH

Dependencies: None (assumed)

FCS_SSH_EXT.1.1 The TSF shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, 4254, and [no other RFCs].

FCS_SSH_EXT.1.2 The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, password-based.

FCS_SSH_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than [*262,144*] bytes in an SSH transport connection are dropped.

FCS_SSH_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms: AES-CBC-128, AES-CBC-256, [*no other algorithms*].

FCS_SSH_EXT.1.5 The TSF shall ensure that the SSH transport implementation uses [SSH_RSA] and [*no other public key algorithms*] as its public key algorithm(s)

FCS_SSH_EXT.1.6 The TSF shall ensure that data integrity algorithms used in SSH transport connection is [*hmac-sha1, hmac-sha2-256, hmac-sha2-512*].

FCS_SSH_EXT.1.7 The TSF shall ensure that diffie-hellman-group14-sha1 and [ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521] are the only allowed key exchange methods used for the SSH protocol.

ST Application Note: for details of NetScaler compliance with RFCs, see section 6.2.

FCS_HTTPS_EXT.1 Explicit: HTTPS

Dependencies: None (assumed)

FCS_HTTPS_EXT.1.1 The TSF shall implement the HTTPS protocol that complies with RFC 2818.

FCS_HTTPS_EXT.1.2 The TSF shall implement HTTPS using TLS as specified in FCS_TLS_EXT.1.

ST Application Note: for details of NetScaler compliance with RFCs, see section 6.2.

5.2.3 User Data Protection (FDP)

FDP_RIP.2 Full Residual Information Protection

Dependencies: None

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

ST Application Note: As clarified in the Assurance Activity for this SFR in [NDPP, 4.2.3], “resources” in the context of this requirement are network packets being sent through (as opposed to “to”, as is the case when a security administrator connects to the TOE) the TOE. The concern is that once a network packet is sent, the buffer or memory area used by the packet still contains data from that packet, and that if that buffer is re-used, those data might remain and make their way into a new packet.

5.2.4 Identification and Authentication (FIA)

FIA_PMG_EXT.1 Password Management

Dependencies: None (assumed)

FIA_PMG_EXT.1.1 The TSF shall provide the following password management capabilities for administrative passwords:

1. *Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters:* [“!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, [“~”, “`”, “-“, “_”, “=”, “+”, “{“, “}”, “[“, “]”, “|”, “\”, “:”, “<”, “>”, “/”, “.”, “,”]];
2. *Minimum password length shall settable by the Security Administrator, and support passwords of 15 characters or greater;*

ST Application Note: the maximum password length on NetScaler is 127 characters.

FIA_UIA_EXT.1 User Identification and Authentication

Dependencies: None (assumed)

FIA_UIA_EXT.1.1 The TSF shall allow the following actions prior to requiring the non-TOE entity to initiate the identification and authentication process:

- Display the warning banner in accordance with FTA_TAB.1;
- [*responses to ping or ARP, VPN vserver authentication (port 443)*]

FIA_UIA_EXT.1.2 The TSF shall require each administrative user to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that administrative user.

ST Application Note: responses to ping or ARP are allowed by default, but can be disabled by an authorised administrator. VPN vserver authentication enables connection of an authorised user who can then request an SSH connection with the NSIP for remote management (if remote management is not enabled then VPN vserver authentication will not be available). This SFR refers to responding services on the externally connected network ports (the appliance also has a management ethernet interface which in the evaluated configuration is not publicly accessible).

FIA_UAU_EXT.2 Extended: Password-based Authentication Mechanism

Dependencies: None (assumed)

FIA_UAU_EXT.2.1 The TSF shall provide a local password-based authentication mechanism, [none] to perform administrative user authentication.

ST Application Note: this means that the TOE only provides the authentication method in FIA_UIA_EXT.1.

FIA_UAU.7 Protected Authentication Feedback

Dependencies: FIA_UAU.1 Timing of authentication

FIA_UAU.7.1 The TSF shall provide only *obscured feedback* to the administrative user while the authentication is in progress at the local console.

Application Note: "Obscured feedback" implies the TSF does not produce a visible display of any authentication data entered by a user (such as the echoing of a password), although an obscured indication of progress may be provided (such as an asterisk for each character). It also implies that the TSF does not return any information during the authentication process to the user that may provide any indication of the authentication data.

5.2.5 Security Management (FMT)

FMT_MTD.1 Management of TSF Data (for general TSF data)

Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1 The TSF shall restrict the ability to *manage* the *TSF data* to the *Security Administrators*.

Application Note: The word "manage" includes but is not limited to create, initialize, view, change default, modify, delete, clear, and append.

FMT_SMF.1 Specification of Management Functions

Dependencies: None

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

- *Ability to administer the TOE locally and remotely;*
- *Ability to update the TOE, and to verify the updates using [published hash] capability prior to installing those updates;*
- *[Ability to configure the cryptographic functionality]*

FMT_SMR.2	**Restrictions on Security Roles**

Dependencies: FIA_UID.1 Timing of identification

FMT_SMR.2.1 The TSF shall maintain the roles:

- **Authorized Administrator.**

FMT_SMR.2.2 The TSF shall be able to associate users with roles.

FMT_SMR.2.3 The TSF shall ensure that the conditions

- **Authorized Administrator role shall be able to administer the TOE locally;**
- **Authorized Administrator role shall be able to administer the TOE remotely;**

are satisfied.

ST Application Note: "Security Administrator" is used in the manner of a role name throughout [NDPP], and in particular in FMT_PMG_EXT.1.1 bullet 2 (see [NDPP, 4.2.4]) and in FMT_MTD.1 (see [NDPP, 4.2.5]). But FMT_SMR.2 uses only the role name "Authorized Administrator" as in the SFR text above. These terms are therefore treated as synonyms in this Security Target.

5.2.6 Protection of the TSF (FPT)

FPT_SKP_EXT.1	**Extended: Protection of TSF Data (for reading of all symmetric keys)**

Dependencies: None (assumed)

FPT_SKP_EXT.1.1 The TSF shall prevent reading of all pre-shared keys, symmetric keys, and private keys.

Application Note: The intent of the requirement is that an administrator is unable to read or view the identified keys (stored or ephemeral) through "normal" interfaces. While it is understood that the administrator could directly read memory to view these keys, do so is not a trivial task and may require substantial work on the part of an administrator. Since the administrator is considered a trusted agent, it is assumed they would not endeavour in such an activity.

FPT_APW_EXT.1	**Extended: Protection of Administrator Passwords**

Dependencies: None (assumed)

FPT_APW_EXT.1.1 The TSF shall store passwords in non-plaintext form.

FPT_APW_EXT.1.2 The TSF shall prevent the reading of plaintext passwords.

Application Note: The intent of the requirement is that raw password authentication data are not stored in the clear, and that no user or administrator is able to read the plaintext password through "normal" interfaces. An all-powerful administrator of course could directly read memory to capture a password but is trusted not to do so.

FPT_STM.1	**Reliable Time Stamps**

Dependencies: None

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

[*E Note that this is not the text of the SFR as written in CC/2 v3.1 rev 4, in fact it was last in v2.3. But this is an NDPP legacy problem.]

FPT_TUD_EXT.1	**Trusted Update**

Dependencies: None (assumed)

FPT_TUD_EXT.1.1 The TSF shall provide security administrators the ability to query the current version of the TOE firmware/software.

FPT_TUD_EXT.1.2 The TSF shall provide security administrators the ability to initiate updates to TOE firmware/software.

FPT_TUD_EXT.1.3 The TSF shall provide a means to verify firmware/software updates to the TOE using a [published hash] prior to installing those updates.

ST Application Note: SHA-256 is used for hashes on software updates (cf. FCS_COP.1(3) in section 5.2.2)

FPT_TST_EXT.1	**TSF Testing**

Dependencies: None (assumed)

FPT_TST_EXT.1.1 The TSF shall run a suite of self tests during initial start-up (on power on) to demonstrate the correct operation of the TSF.

5.2.7 TOE Access (FTA)

FTA_SSL_EXT.1	**TSF-initiated Session Locking**

Dependencies: None (assumed)

FTA_SSL_EXT.1.1 The TSF shall, for local interactive sessions, [terminate the session] after a Security Administrator-specified time period of inactivity.

FTA_SSL.3	**TSF-initiated Termination**

Dependencies: None

FTA_SSL.3.1 Refinement: The TSF shall terminate **a remote** interactive session after a [*Security Administrator-configurable time interval of session inactivity*].

FTA_SSL.4 User-initiated Termination

Dependencies: None

FTA_SSL.4.1 The TSF shall allow Administrator-initiated termination of the Administrator’s own interactive session.

FTA_TAB.1 Default TOE Access Banners

Dependencies: None

FTA_TAB.1.1 Refinement: Before establishing **an administrative user** session the TSF shall display a **Security Administrator-specified** advisory **notice and consent** warning message regarding use of the TOE.

Application Note: This requirement is intended to apply to interactive sessions between a human user and a TOE. IT entities establishing connections or programmatic connections (e.g., remote procedure calls over a network) are not required to be covered by this requirement.

5.2.8 Trusted Path/Channels (FTP)

FTP_ITC.1 Inter-TSF trusted channel

Dependencies: None

FTP_ITC.1.1 Refinement: The TSF shall **use [SSH, TLS]** to provide a **trusted** communication channel between itself and **authorized IT entities supporting the following capabilities: audit server, [authentication server]** that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data **from disclosure and detection of modification of the channel data**.

FTP_ITC.1.2 The TSF shall permit *the TSF*, **or the authorized IT entities** to initiate communication via the trusted channel.

FTP_ ITC.1.3 The TSF shall initiate communication via the trusted channel for [*export of audit logs to external audit server, authentication dialogue with authentication servers*].

ST Application Note: in practice in FTP_ITC.1.2 it is always the TSF that initiates communications using the trusted channel.

FTP_TRP.1 Trusted Path

Dependencies: None

FTP_TRP.1.1 Refinement: The TSF shall **use [SSH, TLS/HTTPS]** provide **a trusted** communication path between itself and **remote administrators** that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from *disclosure and detection of modification of the communicated data*.

FTP_TRP.1.2 Refinement: The TSF shall permit **remote administrators** to initiate communication via the trusted path.

FTP_TRP.1.3 The TSF shall require the use of the trusted path for *initial administrator authentication and all remote administration actions*.

ST Application Note: this SFR includes the use of the TLS VPN to create a tunnel using TLS/HTTPS within which a remote management connection using SSH is made to the NetScaler appliance, as described in section 1.2.2.2 and in [CCECG, Chapter 3, *Configuring NetScaler Gateway].*

5.3 Security Assurance Requirements

The security assurance requirements are drawn from [NDPP, 4.3] and are summarised in Table 6 below. In addition, ASE requirements (from EAL1) are included for the evaluation of this Security Target.

Assurance Class	Assurance Components	Assurance Components Description
Security Target	ASE_INT.1	ST Introduction
	ASE_CCL.1	Conformance claims
	ASE_OBJ.1	Security objectives
	ASE_ECD.1	Extended components definition
	ASE_REQ.1	Derived security requirements
	ASE_TSS.1	TOE Summary Specification
Development	ADV_FSP.1	Basic Functional Specification
Guidance Documents	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative user guidance
Tests	ATE_IND.1	Independent testing - conformance
Vulnerability Assessments	AVA_VAN.1	Vulnerability analysis
Life Cycle Support	ALC_CMC.1	Labeling of the TOE

	ALC_CMS.1	TOE CM coverage

Table 6: Security Assurance Requirements

5.4 Security Requirements Rationale

5.4.1 Mapping between SFRs and Security Objectives

The mapping between security objectives for the TOE and the SFRs that implement them is adopted from [NDPP, 3], and is not reproduced here.

5.4.2 SFR Dependencies Analysis

The dependencies between SFRs implemented by the TOE are addressed as follows.

SFR	Dependencies	Rationale Statement
FAU_GEN.1	FPT_STM.1	FPT_STM.1 included
FAU_GEN.2	FAU_GEN.1	FAU_GEN.1 included
	FIA_UID.1	Satisfied by FIA_UIA_EXT.1, which specifies the relevant administrator identification timing.
FAU_STG_EXT.1	None (assumed)	
FCS_CKM.1	FCS_CKM.2 or FCS_COP.1	FCS_COP.1 (8 iterations) included
	FCS_CKM.4	Met by FCS_CKM_EXT.4, which specifies the relevant key destruction requirements.
FCS_CKM_EXT.4	None (assumed)	
FCS_COP.1 (all iterations)	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	FCS_CKM.1 included
	FCS_CKM.4	Met by FCS_CKM_EXT.4, which specifies the relevant key destruction requirements.
FCS_RBG_EXT.1	None (assumed)	
FCS_TLS_EXT.1	None (assumed)	
FCS_SSH_EXT.1	None (assumed)	
FCS_HTTPS_EXT.1	None (assumed)	

SFR	Dependencies	Rationale Statement
FDP_RIP.2	None	
FIA_PMG_EXT.1	None (assumed)	
FIA_UIA_EXT.1	None (assumed)	
FIA_UAU_EXT.2	None (assumed)	
FIA_UAU.7	FIA_UAU.1	Satisfied by FIA_UIA_EXT.1, which specifies the relevant administrator authentication timing
FMT_MTD.1	FMT_SMR.1	FMT_SMR.2 (hierarchic to FMT_SMR.1) included
	FMT_SMF.1	FMT_SMF.1 included
FMT_SMF.1	None	
FMT_SMR.2	FIA_UID.1	Satisfied by FIA_UIA_EXT.1, which specifies the relevant administrator identification timing.
FPT_SKP_EXT.1	None (assumed)	
FPT_APW_EXT.1	None (assumed)	
FPT_STM.1	None	
FPT_TUD_EXT.1	None (assumed)	
FPT_TST_EXT.1	None (assumed)	
FTA_SSL_EXT.1	None (assumed)	
FTA_SSL.3	None	
FTA_SSL.4	None	
FTA_TAB.1	None	
FTP_ITC.1	None	
FTP_TRP.1	None	

Table 7: Analysis of SFR dependencies

6. TOE Summary Specification

6.1 Security Functions

6.1.1 Security Audit

NetScaler generates audit records for commands executed by Administrators using the CLI and for other security-related events as shown in Table 8. In general terms the audit records include the date and time of the event, type of event (including the selected options in the case of administrator commands), subject identity (if applicable), the outcome (success or failure) of the event, and (if connecting remotely) the IP address of the relevant IT entity. Other details specific to each event are indicated in Table 8. (FAU_GEN.1, FAU_GEN.2)

Audit records are stored on the TOE in /var/log until they are regularly copied by a cron job to an external audit server using SCP (the transfer is protected by the same SSH implementation as in FCS_SSH_EXT.1 and section 6.1.2 below). By default the TOE creates new audit log files whenever an existing log file reaches a user-configurable size, and retains a user-configurable number of versions of each log file (with the oldest being deleted when the next file is created, by which time it will have been transferred to the external audit server). The regular copying is configured as described in [CCECG, Chapter 7, *Securely Transfer Audit Records to a Remote Audit Server*], and is protected by a monitor process which checks every minute that the cron daemon is running (if the daemon is not running at the time of the check then it is restarted automatically). (FAU_STG_EXT.1)

While the audit records are held on the NetScaler appliance, the TOE protects them by means of operating system access controls on the files so that only authorized administrators can access them.

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	Startup and shutdown of audit functions.	
FAU_GEN.2	None.	
FAU_STG_EXT.1	None.	
FCS_CKM.1	None.	
FCS_CKM_EXT.4	None.	
FCS_COP.1(1)/HW	None.	
FCS_COP.1(1)/SW	None.	
FCS_COP.1(2)/HW	None.	

Requirement	Auditable Events	Additional Audit Record Contents
FCS_COP.1(2)/SW	None.	
FCS_COP.1(3)	None.	
FCS_COP.1(4)/HW	None.	
FCS_COP.1(4)/SW	None.	
FCS_RBG_EXT.1	None.	
FCS_TLS_EXT.1	Failure to establish a TLS Session. Establishment/Termination of a TLS session.	Reason for failure. Non-TOE endpoint of connection (IP address) for both successes and failures.
FCS_SSH_EXT.1	Failure to establish an SSH session Establishment/Termination of an SSH session	Reason for failure Non-TOE endpoint of connection (IP address) for both successes and failures.
FCS_HTTPS_EXT.1	Failure to establish an HTTPS Session. Establishment/Termination of an HTTPS session.	Reason for failure Non-TOE endpoint of connection (IP address) for both successes and failures.
FDP_RIP.2	None.	
FIA_PMG_EXT.1	None.	
FIA_UIA_EXT.1	All use of the identification and authentication mechanism.	Provided user identity, origin of the attempt (e.g., IP address).
FIA_UAU_EXT.2	All use of the authentication mechanism[7].	Origin of the attempt (e.g., IP address).
FIA_UAU.7	None.	
FMT_MTD.1	None.	
FMT_SMF.1	None.	

[7] As noted in section 5.2.4, this SFR uses the same authentication mechanism as in FIA_UIA_EXT.1.

Requirement	Auditable Events	Additional Audit Record Contents
FMT_SMR.2	None.	
FPT_SKP_EXT.1	None.	
FPT_APW_EXT.1	None.	
FPT_STM.1	Changes to the time.	The old and new values for the time. Origin of the attempt (e.g., IP address).
FPT_TUD_EXT.1	Initiation of update.	No additional information.
FPT_TST_EXT.1	None.	
FTA_SSL_EXT.1	Termination of session due to timeout[8].	No additional information.
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	No additional information.
FTA_SSL.4	The termination of an interactive session.	No additional information.
FTA_TAB.1	None.	
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1	Initiation of the trusted path. Termination of the trusted path. Failures of the trusted path functions.	Identification of the claimed user identity.

[8] Since sessions that time out are terminated, there is no “unlocking of an interactive session” as referred to in the FTA_SSL_EXT.1 auditable events in [NDPP, Table 1]

Table 8: Security Audit events for the TOE

This aspect of NetScaler therefore implements FAU_GEN.1, FAU_GEN.2, FAU_STG_EXT.1, and FMT_MTD.1.

6.1.2 Cryptographic Support

NetScaler incorporates a FIPS 140 Level 3 card as a component of the hardware appliance, and the OpenSSL-FIPS software as part of the NetScaler-OS software. Both of these products are separately validated against FIPS 140-2 (see [FIPS-card] & [FIPS-ossl]). The FIPS 140 card is used for TLS channels (apart from LDAP/S) and OpenSSL-FIPS is used for SSH, TLS for LDAP/S, and for TOE internal keys that protect other critical security parameters.

The TOE generates asymmetric cryptographic keys, for key establishment in cryptographic protocols, in accordance with NIST SP 800-56B, with key sizes equivalent to, or greater than, a symmetric key strength of 112 bits. (FCS_CKM.1)

The TOE supports the following cryptographic operations using the FIPS 140 card and OpenSSL-FIPS components:

- Encryption and decryption using AES-128 and AES-256 in CBC mode, meeting FIPS PUB 197 and NIST SP 800-38A. (FCS_COP.1(1)/HW, FCS_COP.1(1)/SW)

- Cryptographic signature services using the RSA Digital Signature Algorithm (rDSA) with a key size (modulus) of 2048 bits or greater, meeting FIPS PUB 186-2. (FCS_COP.1(2)/HW, FCS_COP.1(2)/SW)

- Cryptographic hashing services using SHA-1, SHA-256 and SHA-512 (giving message digest sizes of 160, 256, or 512 bits respectively), meeting FIPS Pub 180-3. (FCS_COP.1(3))

- Keyed hash message authentication using HMAC-SHA-1, HMAC_SHA-256 and HMAC-SHA512, meeting FIPS Pub 198-1 and FIPS Pub 180-3. (FCS_COP.1(4)/HW, FCS_COP.1(4)/SW)

NetScaler uses the SHA-1 cryptographic function to create the stored form of Administrator passwords (cf. section 6.1.6), and uses SHA-256 to create a hash value on software updates that is used to check against a published hash value in order to identify genuine updates (cf. section 6.1.6).

The TOE also generates random bits using the FIPS 140 card and OpenSSL-FIPS components, using a hardware-based noise source in the FIPS 140 card and a software-based noised source in the OpenSSL-FIPS software. The RBG operates in accordance with in accordance with FIPS Pub 140-2 Annex C, and is seeded with a minimum of 256 bits of entropy. (FCS_RBG_EXT.1/HW, FCS_RBG_EXT.1/SW,)

The TOE implements TLS versions 1.0 (RFC 2246), 1.1 (RFC 4346) and 1.2 (RFC 5246) with ciphersuites TLS_RSA_WITH_AES_128_CBC_SHA and TLS_RSA_WITH_AES_256_CBC_SHA. (FCS_TLS_EXT.1)

The TOE implements SSH (compliant with RFCs 4251, 4252, 4253 and 4254) for administrators to make remote connections to access the CLI (as an alternative to the use of the local console). The TOE supports both public key-based and password-based authentication methods for SSH. When using public-key authentication methods, public keys are stored in /nsconfig/ssh/authorized_keys. Host private keys are stored in /nsconfig/ssh, and user-authentication private keys are stored in user-chosen subdirectories of /var or /flash. When connecting over SSH, the ssh daemon looks up the relevant public key in the authorized_keys file. If a public key is present then it will be used for authentication, otherwise password-based authentication is used, passing the username and passphrase details to the PAM library to confirm their validity. If the authentication is successful, then the login process uses an exec system call to launch the CLI.

SSH packets larger than 256KB (262,144 bytes) are dropped by the TOE.

For SSH transport the TOE uses SSH_RSA, AES-CBC-128 and AES-CBC-256. The data integrity algorithms used are hmac-sha1, hmac-sha2-256, hmac-sha2-512.

The TOE uses only diffie-hellman-group14-sha1, ecdh-sha2-nistp256, ecdh-sha2-nistp384, and ecdh-sha2-nistp521 as the SSH key exchange methods (this is configurable in the sshd_config file and is described in [CCECG, Chapter 5, *Configuring SSH for NetScaler*]

(FCS_SSH_EXT.1)

The TOE implements HTTPS (compliant with RFC 2818) using TLS. The TOE implements TLS versions 1.0 (RFC 2246), 1.1 (RFC 4346) and 1.2 (RFC 5246) with ciphersuites TLS_RSA_WITH_AES_128_CBC_SHA and TLS_RSA_WITH_AES_256_CBC_SHA. TLS keys are held in the FIPS 140 card and are not directly accessible to TOE users or administrators. (FCS_TLS_EXT.1, FCS_HTTPS_EXT.1)

For details of NetScaler compliance with RFCs, see section 6.2.

Cryptographic keys for SSH and TLS are generated by users as described in [CCECG, Chapter 5, *Configuring Cryptography for NetScaler]*.

Cryptographic keys and critical security parameters are zeroised as follows:

- Keys and key components for protection of critical security parameters: memory buffer overwritten with NULL characters after use, on system shutdown or core dump ***If the key encryption key (KEK) variables that are used to derive the master key encryption key must be replaced, the system administrator must first securely delete the files that hold the key variables. This is done using the POSIX srm command. Refer to [CCECG Chapter 5, Creating the System Master Key for Data Protection] for this procedure.***

- VPN user passwords: plaintext data in memory is overwritten with zeroes after use, on system shutdown or before a core-dump is created in the event of a crash (other system user passwords are not held unencrypted in memory)

- TLS keys in hardware: overwritten with zeroes in the FIPS 140 card in accordance with its FIPS 140 security policy

- TLS & SSH keys in software: overwritten with pseudo-random data in the OpenSSL fipscanister in accordance with its FIPS 140 security policy. ***If the SSH host keys must be replaced, the administrator must securely delete the old host key files using the POSIX srm command. Refer to [CCECG Chapter 5, Configuring SSH for NetScaler] for this procedure.***

(FCS_CKM_EXT.4)

This aspect of NetScaler therefore implements FCS_CKM.1, FCS_CKM_EXT.4, FCS_COP.1(1)/HW, FCS_COP.1(1)/SW, FCS_COP.1(2)/HW, FCS_COP.1(2)/SW, FCS_COP.1(3), FCS_COP.1(4)/HW, FCS_COP.1(4)/SW, FCS_RBG_EXT.1/HW, FCS_RBG_EXT.1/SW, FCS_TLS_EXT.1, FCS_SSH_EXT.1 and FCS_HTTPS_EXT.1.

6.1.3 User Data Protection

NetScaler ensures proper reuse of network buffers to prevent data leakage across connections. When a new incoming packet arrives, the NetScaler NIC driver writes the packet data as specified by the frame, sets the network buffer's data pointer to the start of the Ethernet packet header, and sets the data length of the packet. This means that data previously in the buffer is overwritten when the relevant part of the buffer is reallocated to contain the new packet, and ensures that only the data in the packet is available to the packet-engine for further processing. Badly formatted packets and those which do not conform to protocol specifications are discarded.

When a packet is transmitted from NetScaler, the NetScaler NIC driver will only transmit the data as specified by the relevant protocols (e.g. TCP/IP, UDP) by using the network buffer's physical address data pointer and the data length field – no additional data is transmitted (short packets are padded with zeroes by the NIC driver).

This aspect of NetScaler implements FDP_RIP.2.

6.1.4 Identification and Authentication

Administrators access the TOE through the CLI, either using a local connection or via a remote connection using SSH over a TLS VPN tunnel. Identification and authentication is required for administrators before access is given to any of the TOE functions except for the display of the warning banner (as in FTA_TAB.1), responses to ping or ARP, or authentication as a VPN user in order to establish an SSH connection to the NSIP for remote management *(if remote management is not enabled then VPN vserver authentication will not be available)*. (FIA_UIA_EXT.1)

After the NetScaler appliance has initialised, a login process listens for connections at the local console and administrators may then use either the local console or a remote SSH connection (via a VPN tunnel) to access the CLI. The login process requests a user identity and authenticates the user by passing the user identity and passphrase to the PAM library to confirm their validity. If the authentication is successful, then the login process uses an exec system call to launch the CLI. (FIA_UAU_EXT.2)

Administrator passwords allow special characters as listed in FIA_PMG.1.1 (see section 5.2.4) and allow the System-Admin or superuser roles[9] to set a minimum password length. Passwords have a maximum length of 127 characters. (FIA_PMG_EXT.1)

When a user enters their password at the local console then no characters are displayed on the console. (FIA_UAU.7)

This aspect of NetScaler implements FIA_PMG_EXT.1, FIA_UIA_EXT.1, FIA_UAU_EXT.2, and FIA_UAU.7.

6.1.5 Security Management

NetScaler defines authorized administrator roles by binding command policies to NetScaler System Users (or to groups that these users belong to) as described in [CCECG, Chapter 4]. This means that the abilities of an authorized administrator can be defined at the level of the individual commands that are available to that user. NetScaler maintains four pre-defined administrator roles based on pre-defined command policies as described in [CCECG, Chapter 4, *Binding Command Policies to the System User Account*] and allows additional roles to be defined by authorized administrators through custom command policies (CCECG, Chapter 4). The System-Admin role is one such additional role that is created according to [CCECG, Chapter 4], and defines a role that can carry out routine administration operations without requiring superuser privileges. A System User therefore has a single role that is defined by the relevant set of bound command policies. System Users with no associated command policies are subject to the default (DENYALL) command policy, and are unable to execute any configuration commands[10] until the proper command policies are bound to their accounts. (FMT_MTD.1)

The NDPP term "Authorized Administrator" is used in this Security Target to indicate any System User associated with a defined (non-empty) set of command policies on the TOE. The capabilities of each Authorized Administrator can be defined according to their assigned tasks (this definition is performed by an Authorized Administrator who has access to the commands to define and bind command policies as described in [CCECG, Chapter 4, *Superuser Capabilities and Entitlements*]. Initial appliance configuration is accomplished using a built-in superuser account named *nsroot*. The nsroot account cannot be deleted and is used to provision other system user accounts (both superuser and sysadmin) for managing the appliance. [CCECG, Chapter 4, *SysAdmin Capabilities and Entitlements*] describes the capabilities of a specific role called 'SysAdmin', that enables routine administration tasks to be carried out without requiring superuser privilege[11]. (FMT_SMR.2)

[9] A superuser can use a NetScaler command policy to make the relevant commands available to other users if required, but the default configuration allows only these two roles (superuser and System-Admin) to set the minimum password length.

[10] There are a small number of non-configuration commands available in the default (DENYALL) state, as specified in [CCECG Chapter 4]

[11] The actions requiring the superuser role are listed in [CCECG, Chapter 4, *Superuser Capabilities and Entitlements*].

NetScaler System Users can perform administration actions, including configuring the cryptographic functionality, either locally from a console that has a direct serial connection to the NetScaler appliance (this must be used for initial configuration), or remotely via an SSH connection. Only authorized administrators with superuser privilege can install software updates, in the form of new builds, on the NetScaler appliance, as described in [CCECG, Chapter 3*, Upgrading the NetScaler Software Version*]. (FMT_SMF.1)

This aspect of NetScaler therefore implements FMT_MTD.1, FMT_SMF.1, and FMT_SMR.2.

6.1.6 Protection of the TSF

NetScaler does not store passwords in plaintext form, and does not provide an interface to view plaintext passwords. Administrator passwords are stored in hashed (SHA-1) form (after adding a random 64-bit salt to each password); and password strings contained in audit log entries are obscured with asterisks. Passwords for the accounts used to make the TLS VPN connections through which SSH administration sessions are established are held encrypted in files, and in an obfuscated form when in memory, (FPT_APW_EXT.1)

NetScaler does not provide a CLI interface designed to permit the viewing of pre-shared keys, symmetric keys or private keys. These keys are protected from unauthorised access by the use of file permissions. TLS keys are held in the FIPS 140 card and cannot be accessed by authorized administrators. (FPT_SKP_EXT.1)

NetScaler hardware provides a system clock, which is used for timestamps in audit log entries, and to measure periods of inactivity during local and remote administrator sessions in order to determine when an inactive session is to be terminated (cf. section 6.1.7). (FPT_STM.1)

An Authorized Administrator can determine the current version of the NetScaler using the ‘show version’ command to display the software version identifier and the ‘show hardware command to display the hardware model identifier. Updates to the NetScaler software are obtained by an Administrator by download from the Citrix website, and the full update process is described in [CCECG, Chapter 3, *Upgrading the NetScaler Software Version*]. Each update is accompanied by a hash value that is also published on the Citrix website: before applying any update the administrator applies the OpenSSL hash tool on the appliance (as described in [CCECG, Chapter 3, *Upgrading the NetScaler Software Version*], using the SHA-256 hash function in section 6.1.2) and verifying that the hash value obtained matches the value published for that item on the Citrix website. Provided that the hash value is correct the Administrator then applies the update as described in [CCECG, Chapter 3, *Install the Firmware Update*]. (FPT_TUD_EXT.1)

When the NetScaler device is powered-on, self-tests are run in the FIPS 140 card as described in [SecP-card, 8] and in the OpenSSL-FIPS software as described in [SecP-ossl, 6] and [UG-ossl, 2.6.1]. (FPT_TST_EXT.1).

The power-on self tests completed by the FIPS 140 card during system start-up are as follows:

Algorithm	Type	Test Attributes
AES CBC	KAT	Encrypt and decrypt KATs (Cert. #1265)
AES EBC	KAT	Encrypt & Decrypt KATs (Cert. #1266)
AES EBC	KAT	GCM Encrypt & Decrypt KATs (Cert. #2899)
Triple DES	KAT	Encrypt & Decrypt KATs (Cert. #898)
DSA	KAT	Sig Gen/Ver, PQG Gen/Ver and KeyGen KATs (Cert. #474)
ECDSA	KAT	KeyGen and PKV KAT (Cert. #150)
ECDSA	KAT	Sig Gen/Ver, KeyGen and PKV KATs (Cert. #188)
HMAC-SHA-1	KAT	(Cert. #443)
HMAC-SHA-512	KAT	(Cert. #736)
HMAC-SHA-256, -384, -512	KAT	(Cert. #1677)
RNG	KAT	ANSI X9.31 KAT (Cert. #707)
SHS	KAT	160 bit (Cert. #801)
SHS	KAT	256, 384, 512-bit (Cert. #1379)
SP800-90 CTR_DRBG	KAT	Cert. 32
RSA	KAT	SigVer and KeyGen KATs (Cert. #607)
RSA	KAT	Sig Gen/Ver and KeyGen KATs (Cert. #742)
RSA	KAT	Encrypt & Decrypt KAT
KAS	KAT	per IG 9.6 (Q=dG and KDF)
Firmware Integrity	KAT	CRC16

Power-on self-test (POST) for OpenSSL-FIPS is accomplished during system boot-up when the following daemon processes are initialized:

1. sshd
2. nsaad
3. nsnetsvc

During initialization, the services invoke the fips_mode_set() OpenSSL API call, which in turn executes POST for the OpenSSL-FIPS module and returns a status code that indicates success or failure for the test. If the POST is unsuccessful for any of the above services, the daemon will fail to start and initialize. Refer to [CCECG, *Chapter 3*, *FIPS Mode Self-Test*] for a detailed description of system behaviour in the event of POST failure.

The self-tests performed by the OpenSSL-FIPS module during initialization are shown in the table below:

Algorithm	**Type**	**Attributes**
Software Integrity	KAT	HMAC-SHA1
HMAC	KAT	One KAT per SHA1, SHA224, SHA256, SHA384 and SHA512 Per IG 9.3, this testing covers SHA POST requirements.
AES	KAT	Separate encrypt and decrypt, ECB mode, 128 bit key length
AES CCM	KAT	Separate encrypt and decrypt, 192 key length
AES GCM	KAT	Separate encrypt and decrypt, 256 key length
XTS-AES	KAT	128, 256 bit key sizes to support either the 256-bit key size (for XTS-AES-128) or the 512-bit key size (for XTS-AES-256)
AES CMAC	KAT	Sign and verify CBC mode, 128, 192, 256 key lengths
TDES	KAT	Separate encrypt and decrypt, ECB mode, 3-Key
TDES CMAC	KAT	CMAC generate and verify, CBC mode, 3-Key
RSA	KAT	Sign and verify using 2048 bit key, SHA-256, PKCS#1
DSA	PCT	Sign and verify using 2048 bit key, SHA-

		384
DRBG	KAT	CTR_DRBG: AES, 256 bit with and without derivation function HASH_DRBG: SHA256 HMAC_DRBG: SHA256 Dual_EC_DRBG: P-256 and SHA256
ECDSA	PCT	Keygen, sign, verify using P-224, K-233 and SHA512. The K-233 self-test is not performed for operational environments that support prime curve only (see Table 2).
ECC CDH	KAT	Shared secret calculation per SP 800-56A §5.7.1.2, IG 9.6
X9.31 RNG	KAT	128, 192, 256 bit AES keys

This aspect of NetScaler therefore implements FPT_SKP_EXT.1, FPT_APW_EXT.1, FPT_STM.1, FPT_TUD_EXT.1 and FPT_TST_EXT.1.

6.1.7 TOE Access

An Authorized Administrator can specify a maximum inactivity time period for both local and remote interactive sessions after which a session will be automatically terminated by NetScaler, as specified in [CCECG, Chapter 7, *Configure Session Inactivity Timeout*]. (FTA_SSL_EXT.1, FTA_SSL.3).

An Administrator can choose to terminate their own interactive session from the CLI at any time using the ‘logout’ (or ‘exit’ or ctrl-d) command. (FTA_SSL.4). The administrator must also log off from the TLS VPN to complete administrative session termination. This is accomplished by explicitly logging off from the Citrix Receiver after logging off from the CLI.

An Authorized Administrator can specify a banner message that is displayed at the beginning of each administrative user session, as specified in [CCECG, Chapter5, *Configuring a Warning Message for SSH*]. This applies to both local and remote interactive sessions. (FTA_TAB.1)

This aspect of NetScaler therefore implements FTA_SSL_EXT.1, FTA_SSL.3, FTA_SSL.4 and FTA_TAB.1.

6.1.8 Trusted Path/Channels

NetScaler uses trusted channels based on TLS versions 1.0, 1.1 and 1.2 (cf. section 6.1.2) to optionally communicate with external authentication servers, as described in [CCECG, Chapter 6, *Configuring External User Authentication and Chapter 5, Configuring TLS for*

NetScaler]. Additionally, NetScaler uses trusted channels based on SSH for external communication with remote audit servers (configuration of SSH connections detailed in [CCECG, Chapter 5: *Configuring SSH for NetScaler*] and [CCECG, Chapter 7: *Securely Transfer Audit Records to a Remote Audit Server*]). These channels protect the communications from unauthorized disclosure or modification. (FTP_ITC.1)

The trusted path used for remote administrator connections is provided using SSH within a TLS tunnel as described in section 1.2.2.2 and in [CCECG, Chapter 3, *Accessing the NetScaler Appliance Remotely and CCECG, Chapter 5, Configuring SSH for NetScaler*], which provides AES-based protection of the communications against unauthorized disclosure or modification. Both password-based and public-key-based authentication methods are supported. (FTP_TRP.1)

This aspect of NetScaler therefore implements FTP_ITC.1 and FTP_TRP.1.

6.2 External Document Conformance

The sections below describe the conformance to RFCs and to [SP800-56B] as identified in the SFRs and Security Functions.

6.2.1 TLS

NetScaler supports TLS versions 1.0, 1.1 and 1.2, and conforms to RFCs 2246, 4346 and 5246 as indicated below.

RFC	RFC Synopsis	TOE Handling of Security Related Protocol Options
2246	TLS v1.0	The TOE's implementation of this RFC includes all of the 'must' statements, and does not violate the 'must not' statements. Further notes on specific aspects of the RFC are given below. **General**: The TOE implements TLS v1.0 to support the requirements of FTP_TRP.1 and FTP.ITC.1. The primary vehicle for implementing TLS v1.0 is via the FIPS 140 hardware module. Additionally, the OpenSSL-FIPS module is used for Trusted Channel Communication with remote authentication servers (i.e.; LDAP/S). **Random Number Generation and Seeding**: The TOE uses only cryptographically secure pseudo-random number generators for all operations requiring random number support. **Certification and Authentication**: The TOE verifies the integrity of certificates and supports certificate revocation messages. **CipherSuite Support**: The TOE supports the following mandatory ciphersuite: TLS_RSA_WITH_AES_128_CBC_SHA The TOE supports the following optional ciphersuite TLS_RSA_WITH_AES_256_CBC_SHA

		Null Ciphers: The TOE does not support NULL ciphers. **Compression**: The TOE does not support TLS compression. **Authentication and Key Exchange**: The TOE supports both server and client authentication. Anonymous key exchange is not supported by the TOE. The TOE supports RSA key exchange and authentication. **Session Resumption**: The TOE supports session resumption.
4346	TLS v1.1	**General**: The TOE implements TLS v1.1 to support the requirements of FTP_TRP.1 and FTP.ITC.1. The primary vehicle for implementing TLS v1.1 is via the FIPS 140 hardware module. Additionally, the OpenSSL-FIPS module is used for Trusted Channel Communication with remote authentication servers (i.e.; LDAP/S). **Random Number Generation and Seeding**: The TOE uses only cryptographically secure pseudo-random number generators. **Certification and Authentication**: The TOE verifies the integrity of certificates and supports certificate revocation messages. **CipherSuite Support:** The TOE supports the following mandatory ciphersuite: TLS_RSA_WITH_AES_128_CBC_SHA The TOE supports the following optional ciphersuite TLS_RSA_WITH_AES_256_CBC_SHA **Null Ciphers**: The TOE does not support NULL ciphers. **Compression**: The TOE does not support TLS compression. **Authentication and Key Exchange**: The TOE supports both server and client authentication. Anonymous key exchange is not supported by the TOE. • The TOE will support RSA key exchange and authentication~~.~~ **Session Resumption**: The TOE supports session resumption. **Explicit IV's**: The TOE uses explicit IV's for cryptographic operations in lieu of using the CBC residue of previous record as has been the case in previous versions of the TLS protocol.
5246	TLS v1.2	**General**: The TOE implements TLS v1.2 to support the requirements of FTP_TRP.1 and FTP.ITC.1. The primary vehicle for implementing TLS v1.2 is via the FIPS 140 hardware module. Additionally, the OpenSSL-FIPS module is used for Trusted Channel Communication with remote authentication servers (i.e.; LDAP/S). **Random Number Generation and Seeding**: The TOE uses only cryptographically secure pseudo-random number generators. **Certification and Authentication**: The TOE verifies the integrity of

		certificates and supports certificate revocation messages. **CipherSuite Support:** The TOE supports the following mandatory ciphersuite: TLS_RSA_WITH_AES_128_CBC_SHA The TOE supports the following optional ciphersuite TLS_RSA_WITH_AES_256_CBC_SHA. **Null Ciphers**: The TOE does not support NULL ciphers. **Compression**: The TOE does not support TLS compression. **Authentication and Key Exchange**: The TOE supports both server and client authentication. Anonymous key exchange is not supported by the TOE. • The TOE will support RSA key exchange and authentication. **Session Resumption**: The TOE supports session resumption. **Explicit IV's**: The TOE uses explicit IV's for cryptographic operations in lieu of using the CBC residue of previous record as has been the case in previous versions of the TLS protocol.

6.2.2 SSH

NetScaler supports SSH, and conforms to RFCs 4251, 4252, 4253 and 4254 as indicated below.

RFC	RFC Synopsis	TOE Handling of Security Related Protocol Options
4251	Secure Shell (SSH) Protocol Architecture	The TOE's implementation of this RFC includes all of the 'must' statements, and does not violate the 'must not' statements. Further notes on specific aspects of the RFC are given below. **Host Keys**: The TOE has one RSA Host Key for SSH v2. This key is randomly generated on initial setup of the TOE so that it is unique to each TOE instance. The TOE sends its public key to the client, which matches this key against the keys in its known_hosts list. When a client connects to the TOE, the client is able to determine whether the same host key was used in previous connections, or if the key is different (as in the SSHv2 protocol). RSA host keys are 2048 bits in length. The superuser role can manage SSH host keys via the ssh-keygen utility. **Policy Issues**: The TOE implements all mandatory algorithms and methods. The TOE can be configured to accept public-key based authentication and/or password-based authentication, and does not require multiple authentication mechanisms for users. The TOE allows port forwarding and sessions to clients. X11 forwarding is prohibited by the TOE. **Confidentiality**: The TOE does not accept the "none" cipher. For ciphers whose blocksize is 16 or greater, the TOE rekeys after every 2^32 blocks

		that have been sent/received. The client may explicitly request rekeying as a valid SSHv2 message at any time and the TOE will carry out this request. **Denial of Service**: When an SSH connection is lost, the TOE does not attempt to re-establish it. ACLs can be used to control the clients that are able to make an SSH connection to the TOE. **Ordering of Key Exchange Methods**: The TOE orders key exchange algorithms as follows: diffie-hellman-group14-sha1, ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521. **Debug Messages**: The TOE does not provide debug messages. **End Point Security**: The TOE permits port forwarding. **Proxy Forwarding**: The TOE permits proxy forwarding. **X11 Forwarding**: The TOE does not support X11 forwarding.
4252	The Secure Shell (SSH) Authentication Protocol	The TOE's implementation of this RFC includes all of the 'must' statements, and does not violate the 'must not' statements. Further notes on specific aspects of the RFC are given below. **Authentication Protocol:** The TOE does not accept a "none" method for authentication. The TOE allows 120 seconds for authentication to be completed, and disconnects a client after that time if the authentication is not successfully completed. The TOE allows 6 authentication retry attempts before disconnecting the client. **Authentication Requests:** Authentication requests for services that do not exist are rejected by the TOE. Authentication requests for non-existent usernames are rejected by the TOE by sending back a disconnect in the same way as it would do for failed authentications, and the TOE therefore prevents enumeration of valid usernames. The TOE does not accept a "none" method for authentication and responds in such a case with a list of permitted authentication methods. **Public Key Authentication Method:** The TOE supports public key authentication, and authentication succeeds only if the correct private key is used. The TOE does not require multiple authentications (public key and password) for users. **Password Authentication Method:** The TOE supports password authentication. Expired passwords are not supported and cannot be used for authentication. **Host-Based Authentication:** The TOE does not support the configuration of host-based authentication methods.
4253	The Secure Shell (SSH) Transport Layer Protocol	The TOE's implementation of this RFC includes all of the 'must' statements, and does not violate the 'must not' statements. Further notes on specific aspects of the RFC are given below. **Encryption:** The TOE does not allow the "none" algorithm for encryption. The TOE offers the following for encryption of SSH sessions: aes128-cbc and aes256-cbc. The TOE permits negotiation of encryption algorithms in each direction.

		Data Integrity: The TOE supports hmac-sha1, hmac-sha2-256, hmac-sha2-512.The TOE permits negotiation of MAC algorithms in each direction. **Key Re-Exchange:** The TOE performs a key re-exchange when SSH_MSG_KEXINIT is received. The TOE supports diffie-hellman-group14-sha1, ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521. **Public Key Algorithms**: The TOE transport mechanism uses SSH_RSA as its public key algorithm **Maximum Packet Size**: As defined in OpenSSH V6.0p1, PACKET_MAX_SIZE for sshd is 256 KB (256 * 1024). Packet sizes in excess of this limit are discarded.
4254	Secure Shell (SSH) Connection Protocol	The TOE's implementation of this RFC includes all of the 'must' statements, and does not violate the 'must not' statements. Further notes on specific aspects of the RFC are given below. **Multiple channels:** The TOE assigns each channel a number (as detailed in RFC 4251). **Data transfers:** The TOE supports a maximum window size of 262144 bytes for data transfer. **Interactive sessions:** The TOE supports only interactive sessions that do not involve X11 forwarding. **Forwarded X11 connections:** The TOE does not support X11 forwarding. **Starting shells/commands:** The TOE supports starting one of: shell, application program or command (only one request per channel). These run in the context of a channel, and do not halt the execution of the protocol stack. **Window dimension change notices:** The TOE accepts notifications of changes to the terminal size (dimensions) from the client. **Port forwarding :** The TOE fully supports port forwarding.

6.2.3 HTTPS

NetScaler supports HTTPS, and conforms to RFCs 4251, 4252, 4253 and 4254 as indicated below.

RFC	RFC Synopsis	TOE Handling of Security Related Protocol Options
2818	HTTP over TLS	The TOE's implementation of this RFC includes all of the 'must' statements, and does not violate the 'must not' statements. Further notes on specific aspects of the RFC are given below. **Server Behavior:** The TOE recovers gracefully if the client closes the connection. **Port Number**: By default the TOE runs HTTPS over port 443.

		URI Format: The TOE recognizes the "HTTPS" identifier as denoting a request for an HTTPS connection. **Endpoint Identification** a) **Server Identity**: the TOE will use the Common Name field as specified by the Administrator to establish server identity. The TOE will use the matching rules specified by RFC2459 for name matching as defined above. If the hostname does not match the identity in the certificate, the TOE will terminate the connection with a bad certificate error and will log the error to the appropriate audit log. b) **Client Identity**: Typically, the server has no external knowledge of what the client's identity ought to be and so checks (other than that the client has a certificate chain rooted in an appropriate CA) are not possible. Client identity is established by extracting the client certificate subject name and comparing it with the appropriate authentication policy for client certificate authentication.

6.2.4 [SP800-56B]

While the TOE generally fulfills all of the NIST SP 800-56B requirements without extensions, the following table specifically identifies the "should" and "should not" (i.e.; recommended) conditions from the publication along with an indication of how the TOE conforms to those conditions. It is understood that the TOE satisfies all "shall" and "shall not" conditions stated within the NIST SP 800-56B publication.

SP800-56B Section	SP800-56B Subsection	Status of Recommendations
5: Cryptographic Elements	**5.1: Cryptographic Hash Functions**	N/A
	5.2: Message Authentication Code (MAC) Algorithm	N/A
	5.2.1: MacTag Computation	N/A
	5.2.2: MacTag Checking	N/A
	5.2.3: Implementation Validation Message	N/A
	5.3: Random Bit Generation	N/A
	5.4: Prime Number Generators	N/A
	5.5: Primality Testing Methods	N/A

	5.6: Nonces	¶4: Card: Satisfies *should* condition ¶4: OpenSSL: Satisfies *should* condition
	5.7: Symmetric Key-Wrapping Algorithms	N/A
	5.8: Mask Generation Function	N/A
	5.9: Key Derivation Functions for Key Establishment Schemes	N/A
	5.9.1: Concatenation Key Derivation Function	N/A
	5.9.2: ASN.1 Key Derivation Function	N/A
6: RSA Key Pairs	**6.1: General**	¶1.6: Card: Satisfies *should not* condition ¶1.6: OpenSSL: Satisfies *should not* condition ¶1.7: Card: Satisfies *should* condition ¶1.7: OpenSSL: Satisfies *should* condition ¶1.8: Card: Satisfies *should* conditions ¶1.8: OpenSSL: Satisfies *should* conditions ¶1.9: Card: Satisfies *should* condition ¶1.9: OpenSSL Satisfies *should* condition
	6.2: Criteria for RSA Key Pairs for Key Establishment	N/A
	6.2.1: Definition of a Key Pair	N/A
	6.2.2: Formats	N/A
	6.2.3: Parameter Length Sets	¶3: Card: Satisfies

		should condition ¶3: OpenSSL: Satisfies *should* condition
	6.3: RSA Key Pair Generators	
	6.3.1: RSAKPG1 Family: RSA Key Pair Generation with a Fixed Public Exponent	N/A
	6.3.1.1: rsakpg1-basic	N/A
	6.3.1.2: rsakpg1-prime-factor	N/A
	6.3.1.3: rsakpg1-crt	N/A
	6.3.2: RSAKPG2 Family: RSA Key Pair Generation with a Random Public Exponent	N/A
	6.3.2.1: rsakpg2-basic	N/A
	6.3.2.2: rsakpg2-prime-factor	N/A
	6.3.2.3: rsakpg2-crt	N/A
	6.4: Assurances of Validity	N/A
	6.4.1: Assurance of Key Pair Validity	N/A
	6.4.1.1: General Method for Obtaining Owner Assurance of Key Pair Validity	N/A
	6.4.1.2: RSAKPV1 Family: RSA Key Pair Validation with a Fixed Exponent	N/A
	6.4.1.2.1: rsakpv1-basic	N/A
	6.4.1.2.2: rsakpv1-prime-factor	N/A
	6.4.1.2.3: rsakpv1-crt	N/A
	6.4.1.3: RSAKPV2 Family: RSA Key Pair Validation with a Random Exponent	N/A
	6.4.1.3.1: rsakpv2-basic	N/A
	6.4.1.3.2: rsakpv2-prime-factor	N/A
	6.4.1.3.3: rsakpv2-crt	N/A
	6.4.2: Recipient Assurances of Public Key Validity	N/A

	6.4.2.1: General Method for Obtaining Assurance of Public Key Validity	N/A
	6.4.2.2: Partial Public Key Validation for RSA	N/A
	6.5: Assurances of Private Key Possession	N/A
	6.5.1: Owner Assurance of Private Key Possession	¶2: Card: Satisfies *should* condition ¶2: OpenSSL Satisfies *should* condition
	6.5.2: Recipient Assurance of Owner's Possession of a Private Key	¶3: Card: Satisfies *should* condition ¶3: OpenSSL: Satisfies *should* condition
	6.5.2.1: Recipient Indirectly Obtains Assurance of Possession Using a Trusted Third Party	¶1: Card: Satisfies *should* condition ¶1: OpenSSL: Satisfies *should* condition
	6.5.2.2: Recipient Obtains Assurance of Possession Directly from the Claimed Owner	N/A
	6.6: Key Confirmation	N/A
	6.6.1: Unilateral Key Confirmation for Key Establishment Schemes	N/A
	6.6.2: Bilateral Key Confirmation for Key Establishment Schemes	N/A
	6.7: Authentication	N/A
7: IFC Primitives and Operations	**7.1: Encryption and Decryption Primitives**	N/A
	7.1.1: RSAEP	N/A
	7.1.2: RSADP	Note ¶1: Card: Satisfies *should* condition Note ¶1: OpenSSL: Satisfies *should* condition
	7.2 Encryption and Decryption Operations	N/A

	7.2.1: RSA Secret Value Encapsulation (RSASVE)	N/A
	7.2.1.1: RSASVE Components	N/A
	7.2.1.2: RSASVE Generate Operation	N/A
	7.2.1.3: RSASVE Recovery Operation	Note ¶1: Card: Satisfies *should* conditions Note ¶1: OpenSSL: Satisfies *should* conditions
	7.2.2: RSA with Optimal Asymmetric Encryption Padding (RSA-OAEP)	N/A
	7.2.2.1: RSA-OAEP Components	N/A
	7.2.2.2: RSA-OAEP Encryption Operation	N/A
	7.2.2.3: RSA-OAEP Decryption Operation	Notes ¶1: Card: Satisfies *should* conditions Notes ¶1: OpenSSL: Satisfies *should* conditions Notes ¶2: Card: Satisfies *should* condition Notes ¶2: OpenSSL: Satisfies *should* condition Notes ¶2: Card: Satisfies *should not* condition Notes ¶2: OpenSSL: Satisfies *should not* condition
	7.2.3: RSA-based Key-Encapsulation Mechanism with a Key-Wrapping Scheme (RSA-KEM-KWS)	N/A
	7.2.3.1: RSA-KEM-KWS Components	N/A
	7.2.3.2: RSA-KEM-KWS Encryption Operation	N/A
	7.2.3.3: RSA-KEM-KWS Decryption Operation	Notes ¶1: Card: Satisfies

		should condition Notes ¶1: OpenSSL: Satisfies *should* condition Notes ¶2: Card: Satisfies *should* conditions Notes ¶2: OpenSSL: Satisfies *should* conditions Notes ¶2: Card: Satisfies *should not* conditions Notes ¶2: OpenSSL: Satisfies *should not* conditions
8: Key Agreement Schemes		¶2: Card: Satisfies *should* condition ¶2: OpenSSL: Satisfies *should* condition
	8.1: Common Components for Key Agreement	N/A
	8.2: The KAS1 Family	N/A
	8.2.1: KAS1 Family Prerequisites	N/A
	8.2.2: KAS1-basic	N/A
	8.2.3: KAS1 Key Confirmation	N/A
	8.2.3.1: KAS1 Key Confirmation Components	N/A
	8.2.3.2: KAS1-responder-confirmation	N/A
	8.2.4: KAS1 Security Properties	N/A
	8.3: The KAS2 Family	N/A
	8.3.1: KAS2 Family Prerequisites	N/A
	8.3.2: KAS2-basic	¶5: Card: Satisfies

		should not condition ¶5: OpenSSL: Satisfies *should not* condition
	8.3.3: KAS2 Key Confirmation	N/A
	8.3.3.1: KAS2 Key Confirmation Components	N/A
	8.3.3.2: KAS2-responder-confirmation	N/A
	8.3.3.3: KAS2-initiator-confirmation	N/A
	8.3.3.4: KAS2-bilateral-confirmation	N/A
	8.3.4: KAS2 Security Properties	N/A
9: IFC based Key Transport Schemes		N/A
	9.1: Additional Input	N/A
	9.2 KTS-OAEP Family: Key Transport Using RSA-OAEP	N/A
	9.2.1: KTS-OAEP Family Prerequisites	N/A
	9.2.2: Common components	N/A
	9.2.3: KTS-OAEP-basic	N/A
	9.2.4: KTS-OAEP Key Confirmation	N/A
	9.2.4.1: KTS-OAEP Common Components for Key Confirmation	N/A
	9.2.4.2: KTS-OAEP-receiver-confirmation	N/A
	9.2.5: KTS-OAEP Security Properties	N/A
	9.3: KTS-KEM-KWS Family: Key Transport using RSA-KEM-KWS	N/A
	9.3.1: KTS-KEM-KWS Family	N/A

	Prerequisites	
	9.3.2: Common Components of the KTS-KEM-KWS Schemes	N/A
	9.3.3: KTS-KEM-KWS-basic	N/A
	9.3.4: KTS-KEM-KWS Key Confirmation	N/A
	9.3.4.1: KTS-KEM-KWS Common Components for Key Confirmation	N/A
	9.3.4.2: KTS-KEM-KWS-receiver-confirmation	N/A
	9.3.5: KTS-KEM-KWS Security Properties	N/A

End of Document