Cisco Secure Client - AnyConnect 5.1 for Windows 10

Security Target

Version: 0.5

Date: August 20, 2024

Table of Contents

Table of Tables

Table of Figures

Document Introduction

Prepared By:
Cisco Systems, Inc.
170 West Tasman Dr.
San Jose, CA 95134

This document provides the basis for an evaluation of a specific Target of Evaluation (TOE), Cisco Secure Client - AnyConnect 5.1 for Windows 10. This Security Target (ST) defines a set of assumptions about the aspects of the environment, a list of threats that the product intends to counter, a set of security objectives, a set of security requirements, and the IT security functions provided by the TOE which meet the set of requirements. Administrators of the TOE will be referred to as administrators, Authorized Administrators, TOE administrators, semi-privileged, privileged administrators, and security administrators in this document.

Revision History

Version	Date	Change
0.1	July 25, 2023	Initial Version
0.2	October 18, 2023	Updates
0.3	October 31, 2023	Address check-in comments
0.4	December 6, 2023	Updates to address check-out comments
0.5	August 20, 2024	Updates for v5.1 Assurance Maintenance

1. Security Target Introduction

This Security Target contains the following sections:

- Security Target Introduction
- Conformance Claims
- Security Problem Definition
- Security Objectives
- Security Requirements
- TOE Summary Specification
- CAVP Certificates
- References

The structure and content of this ST comply with the requirements specified in the Common Criteria (CC), Part 1, Annex A, and Part 2.

1.1. ST and TOE Reference

This section provides information needed to identify and control this ST and its TOE.

Table 1. ST and TOE Identification

Name	Description
ST Title	Cisco Secure Client - AnyConnect 5.1 for Windows 10
ST Version	0.5
Publication Date	August 20, 2024
Vendor and ST Author	Cisco Systems, Inc.
TOE Reference	Cisco Secure Client - AnyConnect 5.1 for Windows 10

1.1. TOE Overview

The TOE is Cisco Secure Client - AnyConnect 5.1 for Windows 10 (herein after referred to as the VPN client, or the TOE). The TOE enables remote users within an organization to communicate securely as if their devices were directly connected to a private network.

The TOE is a VPN Client software application. A virtual private network (VPN) extends the organization's private network across a shared or public network. A VPN client establishes a IKEv2/IPsec connection to a VPN Gateway which allows the remote user to securely connect to the organization's private network.

1.2. TOE Product Type

The TOE product type is a VPN client. A VPN client provides protection of data in transit across a shared or public network. The TOE implements IPsec which establishes a cryptographic tunnel to protect the transmission of data between IPsec peers. The VPN client is intended to be located outside an organization's private network, protecting data flows between a host and the VPN Gateway.

Use case 3 (Communication) as described in [PP_APP_V1.4] and use case 1 (TOE to VPN Gateway) as described in [MOD_VPNC_V2.4] both apply to the TOE.

1.3. Required non-TOE Hardware/Software/Firmware

The TOE requires the following hardware/software/firmware in the IT environment when the TOE is configured in its evaluated configuration

Table 2. Required IT Environment Components

Component	Usage/Purpose/Description
Certificate Authority	The Certification Authority provides the TOE with valid certificates. The CA also provides the TOE with a method to check the certificate revocation status of the VPN Gateway.
Windows 10 OS Platform	The Windows 10 platform provides an execution platform for the TOE to run. The TOE requires one of the following Common Criteria certified Microsoft Windows 10 Operating Systems to run: ■ Microsoft Windows 10 version 2004 (May 2020 Update) The Windows 10 Operating Systems listed above have been evaluated for conformance with the Protection Profile for General Purpose Operating System and listed on the NIAP Product Compliant List (PCL).
ASA 5500-X series VPN Gateway	The Cisco ASA 5500-X with software version 9.2.2 or later functions as the head-end VPN Gateway. The Cisco AnyConnect TOE communicates only with the Cisco ASA 5500-X Series Gateway.

ASDM Management Platform	The ASDM 7.7 or later operates from any of the following operating systems: ■ Windows 7, 8, 10 ■ Windows Server 2008, 2012, 2012 R2, 2016 and Server 2019 ■ Apple OS X 10.4 or later Note that that ASDM software is installed on the ASA appliance and the management platform is used to connect to the ASA and run the ASDM. The only software installed on the management platform is a Cisco ASDM Launcher.

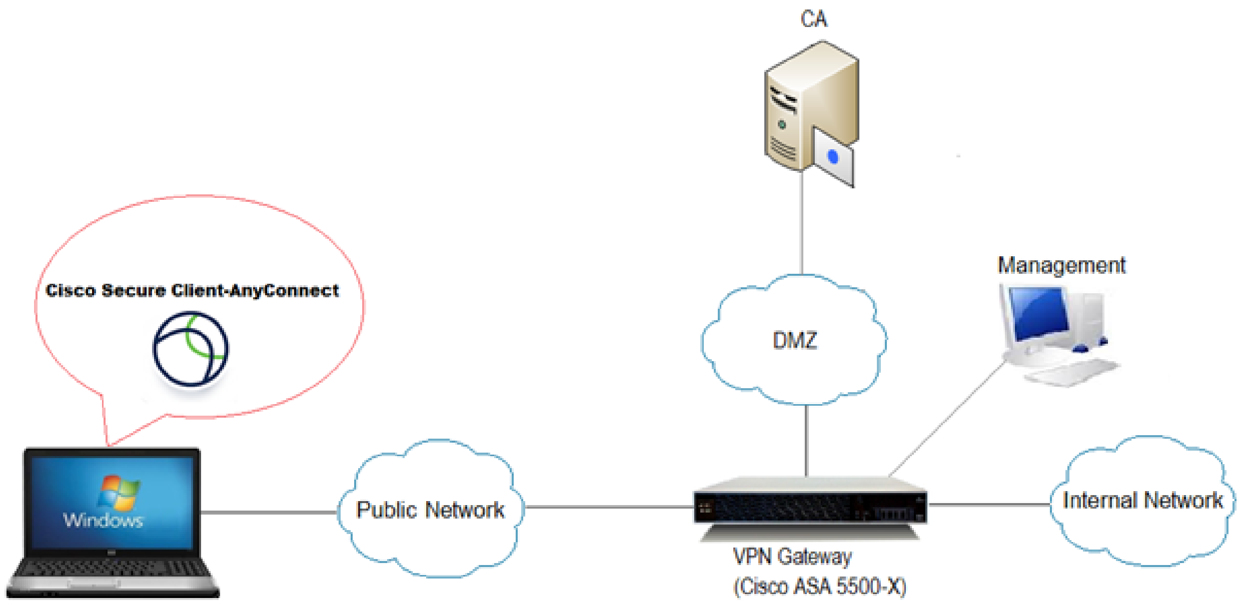
1.4. TOE Description

This section provides an overview of the Target of Evaluation (TOE). The Cisco AnyConnect TOE is a client application that provides remote users a secure VPN tunnel to protect data in transit on both IPv4 and IPv6 networks. The TOE provides IPsec to authenticate and encrypt network traffic travelling across an unprotected public network. By protecting the communication from unauthorized disclosure or modification, remote users can securely connect to an organization's network resources and applications.

1.5. TOE Evaluated Configuration

The following figure provides a visual depiction of the TOE and IT Environment. The TOE is a software app running on Windows 10. The TOE boundary is denoted by the hash red line.

Figure 1. TOE and Environment



Refer to the Common Criteria Administrator's Guide for instructions on installing and configuring the TOE.

1.5.1. Tested Configuration

The Cisco Secure Client-AnyConnect TOE was tested on the following platform in the IT Environment:

Table 3. Tested Platforms

Device Name	Model	Processor	Validated/Certified OS ST
Dell Inspiron	5502	Intel Core i5-1135G7	https://www.niap-ccevs.org/Product/CompliantCC.cfm?CID=2021.1397

1.6. Physical Scope of the TOE

The TOE is a software-only VPN client application. The underlying Windows 10 platform on which the TOE resides is considered part of the IT environment.

1.7. Logical Scope of the TOE

The TOE is comprised of several security features. Each of the security features identified above consists of several security functionalities, as identified below.

- Cryptographic Support

- User Data Protection
- Identification and Authentication
- Security Management
- Privacy
- Protection of the TSF
- Trusted Channels

These features are described in more detail in the subsections below.

Cryptographic Support

The TOE incorporates a cryptographic module, CiscoSSL FIPS Object Module, to provide the cryptography in support of IPsec with ESP symmetric cryptography for bulk AES encryption/decryption and SHA-2 algorithm for hashing. In addition the TOE provides the cryptography to support Elliptic-Curve Diffie-Hellman key exchange and the derivation function used in the IKEv2 and ESP protocols. The cryptographic algorithm implementation has been validated for CAVP conformance. See Table 16 in section 7 for certificate references.

The TOE platform provides asymmetric cryptography, which is used by the TOE for IKE peer authentication using digital signature and hashing services. In addition the TOE platform provides a DRBG.

User Data Protection

The TOE platform ensures that residual information from previously sent network packets processed through the platform are protected from being passed into subsequent network packets.

Identification and Authentication

The TOE and TOE platform perform device-level X.509 certificate-based authentication of the VPN Gateway during IKE v2 key exchange. Device-level authentication allows the TOE to establish a secure channel with a trusted VPN Gateway. The secure channel is established only after each endpoint successfully authenticates each other.

Security Management

The TOE, TOE platform, and VPN Gateway provide the management functions to configure the security functionality provided by the TOE. The TOE provides a Security Administrator role and only the Security Administrator can perform the above security management functions.

Privacy

The TOE does not store or transmit Personally Identifiable Information (PII) over a network.

Protection of the TSF

The TOE performs a suite of self-tests during initial start-up to verify correct operation of its CAVP tested algorithms. Upon execution, the integrity of the TOEs software executables is also verified.

The TOE Platform provides for verification of TOE software updates prior to installation.

Trusted Channels

The TOE's implementation of IPsec provides a trusted channel ensuring sensitive data is protected from unauthorized disclosure or modification when transmitted from the host to a VPN gateway.

1.8. Excluded Functionality

The functionality listed below is not included in the evaluated configuration.

Table 4. Excluded Functionality and Rationale

Function Excluded	Rationale
Non-FIPS mode of operation	This mode of operation includes non-FIPS allowed operations.
SSL Tunnel with DLTS tunneling options	[MOD_VPNC_V2.4] permits only an IPsec VPN tunnel.

2. Conformance Claims

2.1. Common Criteria Conformance Claim

The TOE and ST are compliant with the Common Criteria (CC) Version 3.1, Revision 5, dated: April 2017. The TOE and ST are CC Part 2 extended and CC Part 3 conformant.

2.2. Protection Profile Conformance Claim

The TOE and ST are conformant with the following Protection Profiles:

Table 5. Protection Profile Conformance

Protection Profile	Version	Date
PP-Configuration for Application Software and Virtual Private Network (VPN) Clients	1.3	07 April 2023
The PP-Configuration includes the following components:		
• Base-PP: Protection Profile for Application Software, Version 1.4 (PP_APP_V1.4)	1.4	18 October 2021
• PP-Module: PP-Module for Virtual Private Network (VPN) Clients, Version 2.4 (MOD_VPNC_V2.4)	2.4	31 March 2022

This ST applies the following NIAP Technical Decisions:

Table 6. NIAP Technical Decisions

PP	TD Number	Title	Applicable	Exclusion Rational
[PP_APP_v1.4]	TD0780	FIA_X509_EXT.1 Test 4 Clarification	Yes	
[PP_APP_v1.4]	TD0756	Update for platform-provided full disk encryption	Yes	
[PP_APP_v1.4]	TD0747	Configuration Storage Option for Android	No	The TOE is not for Android Platforms

PP	TD Number	Title	Applicable	Exclusion Rational
[PP_APP_v1.4]	TD0743	FTP_DIT_EXT.1.1 Selection exclusivity	Yes	
[PP_APP_v1.4]	TD0736	Number of elements for iterations of FCS_HTTPS_EXT.1	No	The TOE does not claim HTTPS
[PP_APP_v1.4]	TD0719	ECD for PP APP V1.3 and 1.4	Yes	
[PP_APP_v1.4]	TD0717	Format changes for PP_APP_V1.4	Yes	
[PP_APP_v1.4]	TD0664	Testing activity for FPT_TUD_EXT.2.2	Yes	
[PP_APP_v1.4]	TD0650	Conformance claim sections updated to allow for MOD_VPNC_V2.3 and 2.4	Yes	
[PP_APP_v1.4]	TD0628	Addition of Container Image to Package Format	Yes	
[MOD_VPNC_V2.4]	TD0788	Terminology Change in MOD_VPNC: Extended to Functional Package	Yes	
[MOD_VPNC_V2.4]	TD0753	MOD_VPNC FTP_DIT_EXT.1 Alignment for App PP 1.4	Yes	
[MOD_VPNC_V2.4]	TD0725	Correction to FCS_CKM_EXT.2/4 selections	Yes	
[MOD_VPNC_V2.4]	TD0711	FMT_SMF.1 direction when using MDF 3.3	No	The TOE does not claim MDF as the Base PP.
[MOD_VPNC_V2.4]	TD0697	Alignment with App PP V1.4 for required NIST curves in FCS_CKM.1/AK	Yes	
[MOD_VPNC_V2.4]	TD0690	Missing EAs for FDP_VPN_EXT.1	Yes	
[MOD_VPNC_V2.4]	TD0672	VPN Client PP-Module updated to allow for new PP and PP-Module Versions	Yes	
[MOD_VPNC_V2.4]	TD0662	Changes to Testing IPsec NAT Transversal and XAUTH in MOD_VPNC 2.4	Yes	

PP	TD Number	Title	Applicable	Exclusion Rational
[MOD_VPNC_V2.4]	TD0647	Table 2 Applicability	Yes	

2.3. Protection Profile Conformance Claim Rationale

TOE Appropriateness

The TOE provides all of the functionality at a level of security commensurate with that identified in the U.S. Government Protection Profiles listed in Table 5.

TOE Security Problem Definition Consistency

The Assumptions, Threats, and Organization Security Policies included in the Security Target represent the Assumptions, Threats, and Organization Security Policies specified in [PP_APP_V1.4] and [MOD_VPNC_V2.4] for which conformance is claimed verbatim. All concepts covered in the Protection Profile Security Problem Definition are included in the Security Target Statement of Security Objectives Consistency.

The Security Objectives included in the Security Target represent the Security Objectives specified in [PP_APP_V1.4] and [MOD_VPNC_V2.4] for which conformance is claimed verbatim. All concepts covered in the Protection Profile's Statement of Security Objectives are included in the Security Target.

Statement of Security Requirements Consistency

The Security Functional Requirements included in the Security Target represent the Security Functional Requirements specified in [PP_APP_V1.4] and [MOD_VPNC_V2.4] for which conformance is claimed verbatim. All concepts covered the Protection Profile's Statement of Security Requirements are included in the Security Target. Additionally, the Security Assurance Requirements included in the Security Target are identical to the Security Assurance Requirements included in the claimed Protection Profiles.

3. Security Problem Definition

This section identifies the following:

- Assumptions about the TOE's operational environment. These assumptions include both practical realities in the development of the TOE security requirements and the essential environmental conditions on the use of the TOE.
- Threats addressed by the TOE and the IT Environment.
- Organizational Security Policies imposed by an organization on the TOE to address its security needs.

The security problem definition below has been drawn verbatim from [PP_APP_V1.4] and [MOD_VPNC_V2.4].

3.1. Assumptions

Table 7. TOE Assumptions

Assumption	Assumption Definition
A. PLATFORM	The TOE relies upon a trustworthy computing platform with a reliable time clock for its execution. This includes the underlying platform and whatever runtime environment it provides to the TOE.
A.PROPER_USER	The user of the application software is not willfully negligent or hostile, and uses the software in compliance with the applied enterprise security policy.
A.PROPER_ADMIN	The administrator of the application software is not careless, willfully negligent or hostile, and administers the software in compliance with the applied enterprise security policy.
A.NO_TOE_BYPASS	Information cannot flow onto the network to which the VPN client's host is connected without passing through 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_CONFIG	Personnel configuring the TOE and its operational environment will follow the applicable security configuration guidance.

3.2. Threats

Table 8. Threats

Threat	Threat Definition
T.NETWORK_ATTACK	An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may engage in communications with the application software or alter communications between the application software and other endpoints in order to compromise it.
T.NETWORK_EAVESDROP	An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may monitor and gain access to data exchanged between the application and other endpoints.
T.LOCAL_ATTACK	An attacker can act through unprivileged software on the same computing platform on which the application executes. Attackers may provide maliciously formatted input to the application in the form of files or other local communications.
T.PHYSICAL_ACCESS	An attacker may try to access sensitive data at rest.

T.UNAUTHORIZED_ACCESS	This PP-Module does not include requirements that can protect against an insider threat. Authorized users are not considered hostile or malicious and are trusted to follow appropriate guidance. Only authorized personnel should have access to the system or device that contains the IPsec VPN client. Therefore, the primary threat agents are the unauthorized entities that try to gain access to the protected network (in cases where tunnel mode is used) or to plaintext data that traverses the public network (regardless of whether transport mode or tunnel mode is used). The endpoint of the network communication can be both geographically and logically distant from the TOE, and can pass through a variety of other systems. These intermediate systems may be under the control of the adversary, and offer an opportunity for communications over the network to be compromised. Plaintext communication over the network may allow critical data (such as passwords, configuration settings, and user data) to be read and/or manipulated directly by intermediate systems, leading to a compromise of the TOE or to the secured environmental system(s) that the TOE is being used to facilitate communications with. IPsec can be used to provide protection for this communication; however, there are myriad options that can be implemented for the protocol to be compliant to the protocol specification listed in the RFC. Some of these options can have negative impacts on the security of the connection. For instance, using a weak encryption algorithm (even one that is allowed by the RFC, such as DES) can allow an adversary to read and even manipulate the data on the encrypted channel, thus circumventing countermeasures in place to prevent such attacks. Further, if the protocol is implemented with little-used or

	non-standard options, it may be compliant with the protocol specification but will not be able to interact with other, diverse equipment that is typically found in large enterprises. Even though the communication path is protected, there is a possibility that the IPsec peer could be duped into thinking that a malicious third-party user or system is the TOE. For instance, a middleman could intercept a connection request to the TOE, and respond to the request as if it were the TOE. In a similar manner, the TOE could also be duped into thinking that it is establishing communications with a legitimate IPsec peer when in fact it is not. An attacker could also mount a malicious man-in-the-middle-type of attack, in which an intermediate system is compromised, and the traffic is proxied, examined, and modified by this system. This attack can even be mounted via encrypted communication channels if appropriate countermeasures are not applied. These attacks are, in part, enabled by a malicious attacker capturing network traffic (for instance, an authentication session) and "playing back" that traffic in order to fool an endpoint into thinking it was communicating with a legitimate remote entity.
T.TSF_CONFIGURATION	Configuring VPN tunnels is a complex and time-consuming process, and prone to errors if the interface for doing so is not well-specified or well-behaved. The inability to configure certain aspects of the interface may also lead to the mis-specification of the desired communications policy or use of cryptography that may be desired or required for a particular site. This may result in unintended weak or plaintext communications while the user thinks that their data are being protected. Other aspects of configuring the TOE or using its security mechanisms (for example, the update process) may also result in a reduction in the trustworthiness of the VPN client.

T.USER_DATA_REUSE	Data traversing the TOE could inadvertently be sent to a different user; since these data may be sensitive, this may cause a compromise that is unacceptable. The specific threat that must be addressed concerns user data that is retained by the TOE in the course of processing network traffic that could be inadvertently re-used in sending network traffic to a user other than that intended by the sender of the original network traffic.
T.TSF_FAILURE	Security mechanisms of the TOE generally build up from a primitive set of mechanisms (e.g., memory management, privileged modes of process execution) to more complex sets of mechanisms. Failure of the primitive mechanisms could lead to a compromise in more complex mechanisms, resulting in a compromise of the TSF.

3.3. Organizational Security Policies

There are no organizational security policies defined in [PP_APP_V1.4] and [MOD_VPNC_V2.4].

4. Security Objectives

This section identifies the security objectives of the TOE and the IT Environment. The security objectives identify the responsibilities of the TOE and the TOE's IT environment in meeting the security needs.

4.1. Security Objectives for the TOE

The following table identifies the Security Objectives for the TOE. These security objectives reflect the stated intent to counter identified threats and/or comply with any security policies. The security objectives below have been drawn verbatim from [PP_APP_V1.4] and [MOD_VPNC_V2.4].

Table 9. Security Objectives for the TOE

Environment Security Objective	TOE Security Objective Definition
O.INTEGRITY	Conformant TOEs ensure the integrity of their installation and update packages, and also leverage execution environment-based mitigations. Software is seldom if ever shipped without errors. The ability to deploy patches and updates to fielded software with integrity is critical to enterprise network security. Processor manufacturers, compiler developers, execution environment vendors, and operating system vendors have developed execution environment-based mitigations that increase the cost to attackers by adding complexity to the task of compromising systems. Application software can often take advantage of these mechanisms by using APIs provided by the runtime environment or by enabling the mechanism through compiler or linker options.
O.QUALITY	To ensure quality of implementation, conformant TOEs leverage services and APIs provided by the runtime environment rather than implementing their own versions of these services and APIs. This is especially important for cryptographic services and other complex operations such as file and media parsing. Leveraging this platform behavior relies upon using only documented and supported APIs.

O.MANAGEMENT	To facilitate management by users and the enterprise, conformant TOEs provide consistent and supported interfaces for their security-relevant configuration and maintenance. This includes the deployment of applications and application updates through the use of platform-supported deployment mechanisms and formats, as well as providing mechanisms for configuration. This also includes providing control to the user regarding disclosure of any PII.
O.PROTECTED_STORAGE	To address the issue of loss of confidentiality of user data in the event of loss of physical control of the storage medium, conformant TOEs will use data-at-rest protection. This involves encrypting data and keys stored by the TOE in order to prevent unauthorized access to this data. This also includes unnecessary network communications whose consequence may be the loss of data.
O.PROTECTED_COMMS	To address both passive (eavesdropping) and active (packet modification) network attack threats, conformant TOEs will use a trusted channel for sensitive data. Sensitive data includes cryptographic keys, passwords, and any other data specific to the application that should not be exposed outside of the application.
O.AUTHENTICATION	To address the issues associated with unauthorized disclosure of information in transit, a compliant TOE's authentication ability (IPsec) will allow the TSF to establish VPN connectivity with a remote VPN gateway or peer and ensure that any such connection attempt is both authenticated and authorized.
O.CRYPTOGRAPHIC_FUNCTIONS	To address the issues associated with unauthorized disclosure of information in transit, a compliant TOE will implement cryptographic capabilities. These capabilities are intended to maintain confidentiality and allow for detection and modification of data that is transmitted outside of the TOE.

O.KNOWN_STATE	The TOE will provide sufficient measures to ensure it is operating in a known state. At minimum this includes management functionality to allow the security functionality to be configured and self-test functionality that allows it to assert its own integrity. It may also include auditing functionality that can be used to determine the operational behavior of the TOE.
O.NONDISCLOSURE	To address the issues associated with unauthorized disclosure of information at rest, a compliant TOE will ensure that non-persistent data is purged when no longer needed. The TSF may also implement measures to protect against the disclosure of stored cryptographic keys and data through implementation of protected storage and secure erasure methods. The TOE may optionally also enforce split-tunneling prevention to ensure that data in transit cannot be disclosed inadvertently outside of the IPsec tunnel.

4.2. Security Objectives for the Environment

The following table identifies the Security Objectives for the Environment. These security objectives reflect the stated intent to counter identified threats and/or comply with any security policies. The security objectives below have been drawn verbatim from [PP_APP_V1.4] and [MOD_VPNC_V2.4].

Table 10. Security Objectives for the Environment

Environment Security Objective	IT Environment Security Objective Definition
OE.PLATFORM	The TOE relies upon a trustworthy computing platform for its execution. This includes the underlying operating system and any discrete execution environment provided to the TOE.
OE.PROPER_USER	The user of the application software is not willfully negligent or hostile, and uses the software within compliance of the applied enterprise security policy.
OE.PROPER_ADMIN	The administrator of the application software is not careless, willfully negligent or hostile, and administers the software within compliance of the applied enterprise security policy.

OE.NO_TOE_BYPASS	Information cannot flow onto the network to which the VPN client's host is connected without passing through the TOE.
OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the environment.
OE.TRUSTED_CONFIG	Personnel configuring the TOE and its operational environment will follow the applicable security configuration guidance.

5. Security Requirements

This section identifies the Security Functional Requirements for the TOE. The Security Functional Requirements in this section are drawn from [CC_PART2], [PP_APP_V1.4], [MOD_VPNC_V2.4] and NIAP Technical Decisions.

5.1. Conventions

[CC_PART1] defines operations on Security Functional Requirements. This document uses the following conventions to identify the operations permitted by [PP_APP_V1.4], [MOD_VPNC_V2.4] and NIAP Technical Decisions.

Table 11. Security Requirement Conventions

Convention	Indication
Assignment	Indicated with *italicized* text
Refinement	Indicated with **bold** text and ~~strikethroughs~~
Selection	Indicated with <u>underlined</u> text
Assignment within a Selection	Indicated with *<u>italicized and underlined</u>* text
Iteration	indicated by adding a string starting with '/' (e.g. 'FCS_COP.1/Hash')

Where operations were completed in the [PP_APP_V1.4] or [MOD_VPNC_V2.4], the formatting used in the PP has been retained.

The TOE Security Functional Requirements are identified in the following table are described in more detail in the following subsections.

Table 12. Security Functional Requirements

Class Name	Component Identification	Component Name	Drawn From
FCS: Cryptographic Support	FCS_CKM_EXT.1	Cryptographic Key Generation Services	[PP_APP_V1.4]
	FCS_CKM.1/AK	Cryptographic Asymmetric Key Generation	[PP_APP_V1.4]
	FCS_CKM.2	Cryptographic Key Establishment	[PP_APP_V1.4]
	FCS_CKM.1/VPN	VPN Cryptographic Key Generation (IKE)	[MOD_VPNC_V2.4]
	FCS_COP.1/SKC	Cryptographic Operation – Encryption/Decryption	[PP_APP_V1.4]
	FCS_COP.1/Hash	Cryptographic Operation – Hashing	[PP_APP_V1.4]
	FCS_COP.1/KeyedHash	Cryptographic Operation – Keyed–Hash Message Authentication	[PP_APP_V1.4]
	FCS_COP.1/Sig	Cryptographic Operation – Signing	[PP_APP_V1.4]
	FCS_CKM_EXT.2	Cryptographic Key Storage	[MOD_VPNC_V2.4]
	FCS_CKM_EXT.4	Cryptographic Key Destruction	[MOD_VPNC_V2.4]
	FCS_RBG_EXT.1	Random Bit Generation Services	[PP_APP_V1.4]
	FCS_STO_EXT.1	Storage of Credentials	[PP_APP_V1.4]
	FCS_IPSEC_EXT.1	IPsec	[MOD_VPNC_V2.4]
FDP: User Data Protection	FDP_DEC_EXT.1	Access to Platform Resources	[PP_APP_V1.4]
	FDP_NET_EXT.1	Network Communications	[PP_APP_V1.4]
	FDP_DAR_EXT.1	Encryption Of Sensitive Application Data	[PP_APP_V1.4]

Class Name	Component Identification	Component Name	Drawn From
	FDP_RIP.2	Full Residual Information Protection	[MOD_VPNC_V2.4]
FIA: Identification and authentication	FIA_X509_EXT.1	X.509 Certificate Validation	[PP_APP_V1.4]
	FIA_X509_EXT.2	X.509 Certificate Authentication	[PP_APP_V1.4]
FMT: Security management	FMT_MEC_EXT.1	Supported Configuration Mechanism	[PP_APP_V1.4]
	FMT_CFG_EXT.1	Secure by Default Configuration	[PP_APP_V1.4]
	FMT_SMF.1	Specification of Management Functions	[PP_APP_V1.4]
	FMT_SMF.1/VPN	Specification of Management Functions (VPN)	[MOD_VPNC_V2.4]
FPR: Privacy	FPR_ANO_EXT.1	User Consent for Transmission of Personally Identifiable Information	[PP_APP_V1.4]
FPT: Protection of the TSF	FPT_API_EXT.1	Use of Supported Services and APIs	[PP_APP_V1.4]
	FPT_AEX_EXT.1	Anti-Exploitation Capabilities	[PP_APP_V1.4]
	FPT_TST_EXT.1/VPN	TSF Self-Test	[MOD_VPNC_V2.4]
	FPT_TUD_EXT.1	Integrity for Installation and Update	[PP_APP_V1.4]
	FPT_TUD_EXT.2	Integrity for Installation and Update	[PP_APP_V1.4]
	FPT_LIB_EXT.1	Use of Third Party Libraries	[PP_APP_V1.4]
	FPT_IDV_EXT.1	Software Identification and Versions	[PP_APP_V1.4]

Class Name	Component Identification	Component Name	Drawn From
FTP: Trusted path/channels	FTP_DIT_EXT.1	Protection of Data in Transit	[PP_APP_V1.4]

5.2. Class: Cryptographic Support (FCS)

5.2.1. FCS_CKM_EXT.1. Cryptographic Key Generation Services

FCS_CKM_EXT.1.1 The TSF shall [***implement asymmetric key generation***]

Application Note: This SFR has been modified by [MOD_VPNC_V2.4] and application of NIAP TD0717

5.2.2. FCS_CKM.1/AK Cryptographic Asymmetric Key Generation

FCS_CKM.1.1/AK The **application** shall [

- ***implement functionality***

] **to** generate **asymmetric** cryptographic keys in accordance with a specified cryptographic key generation algorithm [

- *[ECC schemes]* using *[**"NIST curves" ~~P-256~~ P-384 and [P-256, no other curves]*** that meet the following: ***[FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4]*, and,**
- ***no other key generation methods***

]

Application Note: This SFR has been modified by [MOD_VPNC_V2.4] and application of NIAP TD0717

5.2.3. FCS_CKM.2 Cryptographic Key Establishment

FCS_CKM_2.1 The application shall [implement functionality] to perform cryptographic key establishment in accordance with a specified cryptographic key establishment method:

- [***Elliptic curve-based key establishment schemes] that meets the following: [NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"***]***; and,***
- [*No other schemes*]**.**

Application Note: This SFR has been modified by [MOD_VPNC_V2.4]

5.2.4. FCS_CKM.1.1/VPN Cryptographic Key Generation (IKE)

The TSF shall [***invoke platform-provided functionality***] **to** generate **asymmetric** cryptographic keys **used for IKE peer authentication** in accordance with: [

- ***FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3 for RSA schemes,***
- ***FIPS PUB 186-4, "Digital Signature Standard (DSS)," Appendix B.4 for ECDSA schemes and implementing "NIST curves," P-256, P-384 and [no other curves]***

]

and specified cryptographic key sizes [*equivalent to, or greater than, a symmetric key strength of 112 bits*] ~~that meet the following: [assignment: list of standards]~~.

5.2.5. FCS_COP.1/SKC Cryptographic Operation – Encryption/Decryption

FCS_COP.1.1/SKC The application shall perform [encryption/decryption] in accordance with a specified cryptographic algorithm

- **AES-CBC (as defined in NIST SP 800-38A) mode,**
- **AES-GCM (as defined in NIST SP 800-38D) mode,**

and [

- ***no other modes***

] and cryptographic key sizes [*128-bit, 256-bit*].

Application Note: This SFR has been modified by [MOD_VPNC_V2.4]

5.2.6. FCS_COP.1/Hash Cryptographic Operation – Hashing

FCS_COP.1.1/Hash The **application** shall perform [*cryptographic hashing services*] in accordance with a specified cryptographic algorithm [

- *SHA-256*,
- *SHA-384*,
- *No other*

] and **message digest** sizes [

- *256,*
- *384,*
- *No other*

] **bits** that meet the following: [FIPS Pub 180-4].

Application Note: This SFR has been modified by application of NIAP TD0717

5.2.7. FCS_COP.1/KeyedHash Cryptographic Operation – Keyed-Hash Message Authentication

FCS_COP.1.1/KeyedHash The **application** shall perform *keyed-hash message authentication* in accordance with a specified cryptographic algorithm [

- *HMAC-SHA-256*
- *HMAC-SHA-384*

and [

- *no other algorithms*

] **with** key sizes [*256, 384 bits used in HMAC*] and **message digest sizes** [*256 384*] and [no other size] **bits** that meet the following: [FIPS Pub 198-1 *The Keyed-Hash Message Authentication Code and FIPS Pub 180-4 Secure Hash Standard*].

Application Note: This SFR has been modified by application of NIAP TD0717

5.2.8. FCS_COP.1/Sig Cryptographic Operation – Signing

FCS_COP.1.1/Sig The **application** shall perform [*cryptographic signature services (generation and verification)*] in accordance with a specified cryptographic algorithm [

- ***RSA schemes*** *using cryptographic key sizes of [2048-bit or greater] that meet the following: [FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section ~~4~~ 5],*
- ***ECDSA schemes*** *using ["NIST curves" P-256, P-384 and [no other curves]] that meet the following: [FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section ~~5~~ 6].*

].

Application Note: This SFR has been modified by application of NIAP TD0717

5.2.9. FCS_CKM_EXT.2 Cryptographic Key Storage

FCS_CKM_EXT.2.1 The [TOE Platform] shall store persistent secrets and private keys when not in use in platform-provided key storage.

5.2.10. FCS_CKM_EXT.4 Cryptographic Key Destruction

FCS_CKM_EXT.4.1 The [TOE, TOE Platform] shall zeroize all plaintext secret and private cryptographic keys and CSPs when no longer required.

5.2.11. FCS_RBG_EXT.1 Random Bit Generation Services

FCS_RBG_EXT.1.1 The application shall [invoke platform-provided DRBG functionality] for its cryptographic operations.

5.2.12. FCS_STO_EXT.1 Storage of Credentials

FCS_STO_EXT.1.1 The application shall [

- invoke the functionality provided by the platform to securely store [*X.509 Certificates*]

] to non-volatile memory.

5.2.13. FCS_IPSEC_EXT.1 IPsec Protocol

FCS_IPSEC_EXT.1.1 The TSF shall implement the IPsec architecture as specified in RFC 4301.

FCS_IPSEC_EXT.1.2 The TSF shall implement [tunnel mode].

FCS_IPSEC_EXT.1.3 The TSF shall have a nominal, final entry in the SPD that matches anything that is otherwise unmatched, and discards it.

FCS_IPSEC_EXT.1.4 The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms [AES-GCM-128, AES-GCM-256 as specified in RFC 4106, [AES-CBC-128, AES-CBC-256 (both specified by RFC 3602) together with a Secure Hash Algorithm (SHA)-based HMAC]].

FCS_IPSEC_EXT.1.5 The TSF shall implement the protocol: [

- IKEv2 as defined in RFCs 7296 (with mandatory support for NAT traversal as specified in section 2.23), RFC 8784, RFC 8247, and [RFC 4868 for hash functions]].

FCS_IPSEC_EXT.1.6 The TSF shall ensure the encrypted payload in the [IKEv2] protocol uses the cryptographic algorithms [AES-CBC-128, AES-CBC-256 as specified in RFC 6379 and [AES-GCM-128, AES-GCM-256 as specified in RFC 5282]].

FCS_IPSEC_EXT.1.7 The TSF shall ensure that [IKEv2 SA lifetimes can be configured by [VPN Gateway] based on [length of time]]. If length of time is used, it must include at least one option that is 24 hours or less for Phase 1 Sas and 8 hours or less for Phase 2 Sas.

FCS_IPSEC_EXT.1.8 The TSF shall ensure that all IKE protocols implement DH groups [*19 (256-bit Random ECP), 20 (384-bit Random ECP), and* [*no other DH groups*]].

FCS_IPSEC_EXT.1.9 The TSF shall generate the secret value x used in the IKE Diffie-Hellman key exchange (“x” in g^x mod p) using the random bit generator specified in FCS_RBG_EXT.1, and having a length of at least [*256 (for DH Group 19), 384 (for DH Group 20)*] bits.

FCS_IPSEC_EXT.1.10 The TSF shall generate nonces used in IKE exchanges in a manner such that the probability that a specific nonce value will be repeated during the life a specific IPsec SA is less than 1 in 2^[*256*].

FCS_IPSEC_EXT.1.11 The TSF shall ensure that all IKE protocols perform peer authentication using a [ECDSA, RSA] that use X.509v3 certificates that conform to RFC 4945 and [no other method].

FCS_IPSEC_EXT.1.12 The TSF shall not establish an SA if the [IP address, Fully Qualified Domain Name (FQDN)] and [no other reference identifier type] contained in a certificate does not match the expected value(s) for the entity attempting to establish a connection.

FCS_IPSEC_EXT.1.13 The TSF shall not establish an SA if the presented identifier does not match the configured reference identifier of the peer.

FCS_IPSEC_EXT.1.14 The [*VPN Gateway*] shall be able to ensure by default that the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [IKEv2 IKE_SA] connection is greater than or equal to the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [IKEv2 CHILD_SA] connection.

5.3. Class: User Data Protection (FDP)

5.3.1. FDP_DEC_EXT.1 Access to Platform Resources

FDP_DEC_EXT.1.1 The application shall restrict its access to [network connectivity].

FDP_DEC_EXT.1.2 The application shall restrict its access to [no sensitive information repositories].

5.3.2. FDP_NET_EXT.1 Network Communications

FDP_NET_EXT.1.1 The application shall restrict network communication to [

- *user-initiated communication for* [*IKEv2/IPsec tunnel establishment*]

].

5.3.3. FDP_DAR_EXT.1 Encryption Of Sensitive Application Data

FDP_DAR_EXT.1.1 The application shall [

- protect sensitive data in accordance with FCS_STO_EXT.1

] in non-volatile memory.

5.3.4. FDP_RIP.2 Full Residual Information Protection

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

5.4. Class: Identification and Authentication (FIA)

5.4.1. FIA_X509_EXT.1 X.509 Certificate Validation

FIA_X509_EXT.1.1 The application shall [invoked platform-provided functionality] to validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation
- The certificate path must terminate with a trusted CA certificate
- The application shall validate a certificate path by ensuring the presence of the basicConstraints extension, that the CA flag is set to TRUE for all CA certificates, and that any path constraints are met
- The application shall validate that any CA certificate includes caSigning purpose in the key usage field
- The application shall validate the revocation status of the certificate using [CRL as specified in RFC 8603].
- The application shall validate the extendedKeyUsage (EKU) field according to the following rules:
  - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
  - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the EKU field.
  - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the EKU field.
  - S/MIME certificates presented for email encryption and signature shall have the Email Protection purpose (id-kp 4 with OID 1.3.6.1.5.5.7.3.4) in the EKU field.
  - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the EKU field.
  - Server certificates presented for EST shall have the CMC Registration Authority (RA) purpose (id-kp-cmcRA with OID 1.3.6.1.5.5.7.3.28) in the EKU field.

FIA_X509_EXT.1.2 The application shall treat a certificate as a CA certificate only if the basicConstraints extension is present and the CA flag is set to TRUE.

5.4.2. FIA_X509_EXT.2 X.509 Certificate Authentication

FIA_X509_EXT.2.1 The application shall use X.509v3 certificates as defined by RFC 5280 to support authentication for **IPsec** and [**no other protocols**].

FIA_X509_EXT.2.2 When the application cannot establish a connection to determine the validity of a certificate, the TSF shall [*not accept the certificate*].

Application Note: This SFR has been modified by [MOD_VPNC_V2.4]

5.5. Class: Security Management (FMT)

5.5.1. FMT_MEC_EXT.1 Supported Configuration Mechanism

FMT_MEC_EXT.1.1 The application shall [*invoke the mechanisms recommended by the platform vendor for storing and setting configuration options*].

5.5.2. FMT_CFG_EXT.1 Secure by Default Configuration

FMT_CFG_EXT.1.1 The application shall provide only enough functionality to set new credentials when configured with default credentials or no credentials.

FMT_CFG_EXT.1.2 The application shall be configured by default with file permissions which protect the application's binaries and data files from modification by normal unprivileged user.

5.5.3. FMT_SMF.1 Specification of Management Functions

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

- *no management functions*

].

5.5.4. FMT_SMF.1/VPN Specification of Management Functions (VPN)

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

- Specify VPN gateways to use for connections,
- Specify client credentials to be used for connections,
- Configure the reference identifier of the peer

]

5.6. Class: Privacy (FPR)

5.6.1. FPR_ANO_EXT.1 User Consent for Transmission of Personally Identifiable Information

FPR_ANO_EXT.1.1 The application shall [not transmit PII over a network].

5.7. Class: : Protection of the TSF (FPT)

5.7.1. FPT_API_EXT.1 Use of Supported Services and APIs

FPT_API_EXT.1.1 The application shall use only documented platform APIs.

5.7.2. FPT_AEX_EXT.1 Anti-Exploitation Capabilities

FPT_AEX_EXT.1.1 The application shall not request to map memory at an explicit address except for [*no exceptions*].

FPT_AEX_EXT.1.2 The application shall [

- *not allocate any memory region with both write and execute permissions*

].

FPT_AEX_EXT.1.3 The application shall be compatible with security features provided by the platform vendor.

FPT_AEX_EXT.1.4 The application shall not write user-modifiable files to directories that contain executable files unless explicitly directed by the user to do so.

FPT_AEX_EXT.1.5 The application shall be built with stack-based buffer overflow protection enabled.

5.7.3. FPT_TST_EXT.1/VPN TSF Self-Test

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

FPT_TST_EXT.1.2/VPN The [TOE platform] shall provide the capability to verify the integrity of stored TSF executable code when it is loaded for execution through the use of the [*digital signature verification using SHA256 and RSA or ECDSA key provided by the TOE platform*].

5.7.4. FPT_TUD_EXT.1 Integrity for Installation and Update

FPT_TUD_EXT.1.1 The application shall [leverage the platform] to check for updates and patches to the application software.

FPT_TUD_EXT.1.2 The application shall [provide the ability] to query the current version of the application software.

FPT_TUD_EXT.1.3 The application shall not download, modify, replace or update its own binary code.

FPT_TUD_EXT.1.4 The application updates shall be digitally signed such that the application platform can cryptographically verify them prior to installation.

FPT_TUD_EXT.1.5 The application is distributed [as an additional software package to the platform OS].

5.7.5. FPT_TUD_EXT.2 Integrity for Installation and Update

FPT_TUD_EXT.2.1 The application shall be distributed using [the format of the platform-supported package manager].

FPT_TUD_EXT.2.2 The application shall be packaged such that its removal results in the deletion of all traces of the application, with the exception of configuration settings, output files, and audit/log events.

FPT_TUD_EXT.2.3 The application installation package shall be digitally signed such that its platform can cryptographically verify them prior to installation.

5.7.6. FPT_LIB_EXT.1 Use of Third Party Libraries

FPT_LIB_EXT.1.1 The application shall be packaged with only [*OpenSSL, Boost, Libcurl, Rapidxml*].

5.7.7. FPT_IDV_EXT.1 Software Identification and Versions

FPT_IDV_EXT.1.1 The application shall be versioned with [[*sequence-based versioning control*]].

5.8.Class: Trusted Path/Channels (FTP)

5.8.1. FTP_DIT_EXT.1 Protection of Data in Transit

FTP_DIT_EXT.1.1 The application shall encrypt all transmitted [sensitive data] using IPsec as specified in FCS_IPSEC_EXT.1 for [*VPN tunnel*] and [no other protocols] between itself and another trusted IT product.

Application Note: This SFR has been modified by [MOD_VPNC_V2.4] and application of NIAP TD0753

5.9.TOE SFR Dependencies Rationale

[PP_APP_V1.4] and [MOD_VPNC_V2.4] contain all the requirements claimed in this Security Target. As such the dependencies are not applicable since the PPs themselves have been approved.

5.10. Security Assurance Requirements

The TOE assurance requirements for this ST are taken directly from [PP_APP_V1.4] and [MOD_VPNC_V2.4] which are derived from [CC_PART3]. The assurance requirements are summarized in the table below.

Table 13. Assurance Requirements

Assurance Class	Components Description
Security Target (ASE)	Conformance claims (ASE_CCL.1)
	Extended components definition (ASE_ECD.1)
	ST introduction (ASE_INT.1)
	Security objectives for the operational environment (ASE_OBJ.1)
	Stated security requirements (ASE_REQ.1)
	TOE summary specification (ASE_TSS.1)

Development (ADV)	Basic functional specification (ADV_FSP.1)
Guidance Documents (AGD)	Operational user guidance (AGD_OPE.1)
	Preparative procedures (AGD_PRE.1)
Life Cycle Support (ALC)	Labeling of the TOE (ALC_CMC.1)
	TOE CM coverage (ALC_CMS.1)
	Timely Security Updates (ALC_TSU_EXT.1)
Tests (ATE)	Independent testing – conformance (ATE_IND.1)
Vulnerability Assessment (AVA)	Vulnerability survey (AVA_VAN.1)

5.11. Security Assurance Requirements Rationale

The Security Functional Requirements included in the ST represent all mandatory, optional, and selection-based SFRs specified in [PP_APP_V1.4] and [MOD_VPNC_V2.4] against which exact compliance is claimed.

All dependency rationale in the ST are considered to be identical to those that are defined in the claimed PP.

5.12. Assurance Measures

The TOE satisfies the identified assurance requirements. The table below identifies the Assurance Measures applied by Cisco to satisfy the assurance requirements.

Table 14. Assurance Measures

Assurance Component	Rationale
ASE_INT.1 ASE_CCL.1 ASE_OBJ.1 ASE_ECD.1 ASE_REQ.1 ASE_TSS.1	Cisco provided this Security Target document.
ADV_FSP.1	No additional “functional specification” documentation was provided by Cisco to satisfy the Evaluation Activities.

AGD_OPE.1 AGD_PRE.1	Cisco will provide the guidance documents with the ST.
ALC_CMC.1 ALC_CMS.1	Cisco will identify the TOE such that it can be distinguished from other products or versions from the Cisco and can be easily specified when being procured by an end user.
ALC_TSU_EXT.1	Cisco will provide a Security Vulnerability Policy.
ATE_IND.1	Cisco will provide the TOE for testing.
AVA_VAN.1	Cisco will provide the TOE for Vulnerability Analysis.

6. TOE Summary Specification

The table below identifies and describes how the Security Functional Requirements identified above are met by the TOE.

Table 15. TSS Rationale

TOE SFR	Rationale
FCS_CKM_EXT.1 FCS_CKM.1/AK	Key generation for asymmetric keys used by IPsec for key establishment is provided by the TOE and is implemented using ECDSA with NIST curve sizes P-256 and P-384 according to FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4.
FCS_CKM.2	To support IPsec the TOE implements the following algorithms to perform key establishment: ECC key establishment schemes that meet SP800-56A.
FCS_CKM.1/VPN	The TOE Platform provides a specified key generation algorithm to generate asymmetric cryptographic keys for IKE authentication. The key sizes are 2048-bit for RSA scheme and NIST curve sizes P-256 and P-384 when ECDSA is used. The key generation function is invoked by the TOE platform Administrator using the Microsoft Management Console which creates keys and certificates used by the TOE for IKE authentication.
FCS_COP.1/SKC	The TOE provides symmetric encryption and decryption capabilities using AES supporting the following modes: ■ CBC mode as specified in NIST SP 800-38A. ■ GCM mode as specified in NIST SP 800-38D. The TOE uses AES in IPsec using the following modes and key sizes: CBC mode with key size of 128 and 256 bits. GCM mode with key sizes of 128 and 256 bits.
FCS_COP.1/Hash	The TOE provides cryptographic hashing services in support of IKEv2 and IPsec using SHA-256 and SHA-384 as specified in FIPS Pub 180-4 "Secure Hash Standard."
FCS_COP.1/KeyedHash	The TOE provides keyed-hash message authentication services in support of IKEv2 and IPsec. The TOE supports both HMAC-SHA-256 and HMAC-SHA-384 cryptographic algorithms with supported key size of 256 and 384 bits used in HMAC. The message digest sizes supported are 256 bits and 384 bits.
FCS_COP.1/Sig	The TOE provides cryptographic signature services using Elliptic Curve Digital Signature Algorithm with a key size of 256 and 384 bits and RSA Digital Signature Algorithm with key size of 2048 and greater, as specified in FIPS PUB 186-4, "Digital Signature Standard."

FCS_CKM_EXT.4	The TOE ensures volatile memory areas containing the following keys are zeroized:

Key, Secret, or CSP	Purpose	Zeroization Method
SK_ei	IKE SA Initiator Encryption Key	Overwritten with zeros when no longer in use by the IPsec VPN trusted channel.
SK_er	IKE SA Responder Encryption Key	Overwritten with zeros when no longer in use by the IPsec VPN trusted channel.
SK_ai	IKE SA Initiator Integrity Key	Overwritten with zeros when no longer in use by the IPsec VPN trusted channel.
SK_ar	IKE SA Responder Integrity Key	Overwritten with zeros when no longer in use by the IPsec VPN trusted channel.
Diffie-Hellman Shared Secret	IKE v2 SA setup	Overwritten with zeros when no longer in use by the IPsec VPN trusted channel.
SK_d	IKEv2 SA key from which child IPsec keys are derived.	Overwritten with zeros when no longer in use by the IPsec VPN trusted channel.

<table>
<tr><td>TOE SFR</td><td>Rationale</td></tr>
<tr><td></td><td>
<table>
<tr><td>Initiator encryption and integrity key</td><td>IPsec child SA key that encrypts and authenticates outgoing ESP traffic.</td><td>Overwritten with zeros when no longer in use by the IPsec VPN trusted channel.</td></tr>
<tr><td>Responder encryption and integrity key</td><td>IPsec child SA key that decrypts and authenticates incoming ESP traffic.</td><td>Overwritten with zeros when no longer in use by the IPsec VPN trusted channel.</td></tr>
</table>
The TOE platform zeroizes private keys it manipulates and stores on the TOE platform:
<table>
<tr><th>Key, Secret, or CSP</th><th>Purpose</th><th>Zeroization Method</th></tr>
<tr><td>Asymmetric RSA Private Key stored on the Windows 10 platform</td><td>RSA digital signature generation</td><td>Performed exclusively by the TOE Platform.</td></tr>
<tr><td>Asymmetric ECDSA Private Key stored on the Windows 10 platform</td><td>ECDSA digital signature generation</td><td>Performed exclusively by the TOE Platform.</td></tr>
</table>
</td></tr>
<tr><td>FCS_RBG_EXT.1</td><td>The TOE invokes the BCryptGenRandom API on the platform when needed to generate a cryptographic key. This applies to the following SFRs:
FCS_CKM.2 – Cryptographic Key Establishment
FCS_IPSEC_EXT.1 – IPsec Protocol</td></tr>
<tr><td>FCS_STO_EXT.1</td><td>The Cisco AnyConnect TOE leverages the platform to store X.509v3 certificates used by the TOE for IKE peer authentication. Certificates are stored in the Windows Certificate Store.</td></tr>
</table>

TOE SFR	Rationale
FCS_CKM_EXT.2	The TOE platform stores RSA and ECDSA private keys used by the TOE for IKE peer authentication. Private Keys are stored in the Windows Key Storage Provider (KSP). The TOE does not use pre-shared keys for IPsec.

FCS_IPSEC_EXT.1	The TOE's implementation of the IPsec standard (in accordance with RFC 4301) uses the Encapsulating Security Payload (ESP) protocol to provide authentication, encryption and anti-replay services. By default ESP operates in tunnel mode. No configuration is required by the user or administrator for the TOE to operate in tunnel mode. Remote access policies managed by the administrator of the ASA VPN Gateway provide an interface to create ACLs defining network segments that require IPsec protection. The default behavior of the remote access policy is for the TOE to protect all traffic with IPsec. If an organization explicitly permits use of split-tunneling, a remote access policy on the ASA VPN Gateway allows the administrator to define IPsec protection for the organization's network(s) but bypass protection for other traffic. The Cisco Secure Client-AnyConnect TOE is distributed as a separate software package to the platform OS. The TOE relies on the TOE Platform's SPD table, which processes packets in a very specific order. The TOE only injects SPD rules into the table based on rules to protect all traffic or to protect specific traffic. Effectively, this allows the TOE to be configured in either Protect and Drop or Protect and Bypass mode. When the VPN is connected, one of the two mentioned configurations for packet processing is enforced, and the TOE will always protect traffic first before determining whether or not traffic should be discarded or bypassed. The TOE allows configuring packet processing from one of three options: 1. Tunnel All Networks - Explicitly disable split-tunneling, protects all network traffic (default action). 2. Tunnel Network List Below - Protect only specified networks specified in the Network List. 3. Exclude Network List Below - Bypass networks specified in Network List, and protect all other traffic. The Tunnel All Networks configuration will protect all network traffic. The tunnel will always force traffic through the tunnel. Any network that cannot be reached on the other end of the IPsec tunnel is ultimately dropped. The TOE implements IKEv2 and does not support IKEv1.

	IPsec Internet Key Exchange is the negotiation protocol that lets the TOE and a VPN Gateway agree on how to build an IPsec Security Association (SA). IKE separates negotiation into two phases: phase 1 and phase 2. During IKE Phase 1, the TOE authenticates the remote VPN Gateway using device-level authentication with RSA or ECDSA X.509v3 certificates provided by the TOE platform. The TOE compares its reference identifier to the identifier presented by the VPN Gateway peer. The TOE supports reference identifiers as configured by the Administrator to be either FQDN or IP address and compares it to the Subject Alternative Name (SAN) or the Common Name (CN) fields in the certificate of the peer. The order of comparison is SAN followed by CN. If the TOE successfully matches the reference identifier to the presented identifier, IKE Phase 1 authentication will succeed. Otherwise it will fail if it does not match. Phase 1 creates the first tunnel, which protects later IKE negotiation messages. The key negotiated in phase 1 enables IKE to communicate securely in phase 2. The TOE supports only IKEv2 session establishment. As part of this support, the TOE by default does not support aggressive mode used in IKEv1 exchanges. The TOE supports Diffie-Hellman Group 19 (256-bit Random ECP) and 20 (384-bit Random ECP) in support of IKE Key Establishment negotiated in phase 1. These keys are generated using the DRBG specified in FCS_RBG_EXT.1 having 256 bits of entropy. The administrator is instructed in the CC Configuration Guide to select a supported DH group using one of the following corresponding key sizes (in bits): 256 (for DH Group 19), and 384 (for DH Group 20) bits. For each DH Group, the TOE generates the secret value 'x' used in the IKEv2 Diffie-Hellman key exchange ('x' in g^x mod p) using its DH private key, the IPsec peer's public key and a nonce. When a random number is needed for a nonce, the probability that a specific nonce value will be repeated during the life a specific IPsec SA is less than 1 in 2^{256}. The nonce is likewise generated using the DRBG specified in FCS_RBG_EXT.1. During Phase 2, IKE negotiates the IPsec SA and includes: • The negotiation of mutually acceptable IPsec SA parameters; • The Pseudo-Random Function (PRF) is used for the construction of keying material for cryptographic algorithms used in the SA. • The establishment of IPsec Security Associations to protect packet flows using Encapsulating Security Payload (ESP). The resulting potential strength of the symmetric key will be 128 or 256 bits of security depending on the algorithms negotiated between the two IPsec peers. The VPN

TOE SFR	Rationale
	Gateway ensures by default the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the IKEv2 IKE_SA connection is greater than or equal to the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the IKEv2 CHILD_SA connection. After IKE phase 2 completes, the IPsec SA is established, providing a secure tunnel to a remote VPN Gateway. The TOE uses active SA settings or creates new SAs for initial connections with the ASA VPN Gateway. The TOE supports administratively configured lifetimes for both Phase 1 SAs and Phase 2 SAs. The default time value for Phase 1 SAs is 24 hours. The value for Phase 2 SAs is configurable to 8 hours. Both values are configurable using management functions provided by the VPN Gateway. All ESP processing to authenticate, encrypt, and tunnel the traffic is performed by the TOE. The TOE performs IKEv2 payload and bulk IPsec encryption using AES-GCM-128, AES_GCM-256, AES-CBC-128, or AES-CBC-256 algorithms. The VPN Gateway allows the administrator to configure AES-GCM-128, AES_GCM-256, AES-CBC-128, and AES-CBC-256 encryption algorithms.
FDP_DEC_EXT.1	The Cisco Secure Client-AnyConnect TOE restricts access to network connectivity resources.
FDP_NET_EXT.1	The Cisco Secure Client-AnyConnect TOE limits network communication to user initiated communication for IKEv2/IPsec tunnel establishment
FDP_DAR_EXT.1	Sensitive data in the TOE is defined as the private key used for X.509 certificate generation and peer authentication, which is protected in accordance with FCS_STO.EXT.1
FDP_RIP.2	The processing of network packets for residual information is handled by the TOE platform. The TOE platform ensures that packets transmitted from the TOE platform do not contain residual information from previous network packets. Buffers allocated for a network packet are not reused for subsequent network packets. The TOE platform ensures the memory allocated to the buffer once it's no longer needed is released back to the Windows Operating System.

TOE SFR	Rationale
FIA_X509_EXT.1	The Cisco Secure Client-AnyConnect TOE implements functionality and invokes functionality provided by the TOE platform to validate X.509 certificates used for IP-sec connections. The X.509 certificates are validated using the certificate path validation algorithm defined in RFC 5280, which can be summarized as follows: • the public key algorithm and parameters are checked • the current date/time is checked against the validity period • revocation status is checked using CRL • issuer name of X matches the subject name of X+1 • extensions are processed The certificate validity check is performed when the TOE receives the certificate during an IPsec connection to the ASA VPN Gateway. The TOE invokes functionality provided by the TOE platform to ensure all CA certs contain the basic constraints extension and that the CA=TRUE flag is set. The TOE invokes functionality provided by the TOE platform to ensure that the certificate path terminates in a trusted root CA (i.e. a CA certificate configured on the TOE as trusted). These checks ensure certificate validation results in a trusted root certificate. At any point if a certificate cannot be successfully validated, the CC Configuration Guide instructs the administrator to configure the TOE to not allow the user an option for continuing the connection. In all cases, if a certificate or certificate path cannot be validated, the TOE will not establish an IPsec connection to an untrusted ASA VPN Gateway.

TOE SFR	Rationale
FIA_X509_EXT.2	During TOE installation the user imports a new certificate to the certificate store. The user can also select the certificate used by tapping 'Import' and then 'Device Credential Storage'. The Cisco Secure Client-AnyConnect TOE compares the FQDN of the server it is establishing connectivity with, against the Subject Alternate Name-dnsName attributes in the certificate. If AnyConnect determines there is a mismatch, it will not establish the IPsec trusted channel. At any point if a certificate cannot be successfully validated, the CC Configuration Guide instructs the administrator to configure the TOE to not allow the user an option for continuing the connection. In all cases, if a certificate or certificate path cannot be validated, the TOE will not establish an IPsec connection to an untrusted VPN Gateway.
FMT_MEC_EXT.1	All IPsec configuration for the Cisco AnyConnect TOE is stored remotely on the Cisco ASA VPN Gateway. As described in guidance the following settings which must enabled: " FIPS Mode" "Strict Certificate Trust" "Enable CRL Check"
FMT_CFG_EXT.1	The Cisco Secure Client-AnyConnect TOE requires client credentials to be used for connection but is not installed with any preset default credentials. In context of the AnyConnect TOE, client credentials are a X.509 certificate which is used to authenticate the ASA VPN Gateway during establishment of an IPsec session. Users can only access files which are associated to the installation that user performed.
FMT_SMF.1	The Cisco Secure Client-AnyConnect TOE does not perform any security management functions from [PP_APP_V1.4].
FMT_SMF.1/VPN	The Cisco Secure Client-AnyConnect TOE is capable of the following security management functions from [MOD_VPNC_V2.4]: ■ Specify VPN gateways to use for connections ■ Specify client credentials to be used for connections ■ Configuring the reference identifier of the peer

TOE SFR	Rationale
FPR_ANO_EXT.1	The Cisco Secure Client-AnyConnect TOE does not transmit PII.

FPT_API_EXT.1	The Cisco Secure Client-AnyConnect TOE uses the following platform APIs: ▪ WinHttp.h o WinHttpSetOption o WinHttpQueryOption o WinHttpOpen o WinHttpConnect o WinHttpOpenRequest o WinHttpSendRequest o WinHttpCloseHandle o WinHttpQueryHeaders o WinHttpReadData o WinHttpQueryDataAvailable o WinHttpSetCredentials o WinHttpReceiveResponse o WinHttpSetStatusCallback o WinHttpGetProxyForUrl o WinHttpQueryAuthSchemes o WinHttpSetDefaultProxyConfiguration o WinHttpCrackUrl o WinHttpWriteData o WinHttpAddRequestHeaders o WinHttpGetDefaultProxyConfiguration ▪ Cryptuiapi.h o CryptUIDlgViewCertificateW o CryptUIDlgViewCertificateA ▪ Ncrypt.h

	o NCryptSignHash o NCryptSetProperty o NCryptOpenStorageProvider o NCryptOpenKey o NCryptFreeObject ■ WinCrypt.h o CryptAcquireCertificatePrivateKey o CryptSetProvParam o CryptAcquireContextA o CryptCreateHash o CryptSetHashParam o CryptSignHash o CryptDestroyHash o CryptReleaseContext o CryptHashData o CryptGetHashParam o CertGetIntendedKeyUsage o CertFindExtension o CryptDecodeObjectEx o CryptHashPublicKeyInfo o CertOpenStore o CertCloseStore o CertEnumCertificatesInStore o CertFreeCertificateContext o CertFreeCertificateChain o CertAddCertificateContextToStore

	○ CertGetSubjectCertificateFromStore ○ CertDeleteCertificateFromStore ○ CertGetNameStringW ○ CertAddEncodedCertificateToStore ○ CertDuplicateCertificateContext ○ PFXIsPFXBlob ○ PFXImportCertStore ○ CryptFindCertificateKeyProvInfo ○ CryptGetDefaultProvider ○ CertVerifyCertificateChainPolicy ■ Wininet.h ○ InternetErrorDlg ○ InternetSetCookie ○ InternetSetOption ○ InternetQueryOption ○ InternetOpen ○ InternetConnect ○ HttpOpenRequest ○ HttpSendRequest ○ InternetCloseHandle ○ HttpQueryInfo ○ InternetReadFile ○ InternetQueryDataAvailable ○ InternetGetConnectedState ■ Securitybaseapi.h ○ CreateRestrictedToken

	o DuplicateTokenEx o GetTokenInformation o GetLengthSid o CopySid o IsValidSid o GetSidSubAuthorityCount o GetSidIdentifierAuthority o GetSidSubAuthority o AdjustTokenPrivileges o EqualSid o AllocateAndInitializeSid o FreeSid ▪ Processthreadsapi.h o SetThreadToken o OpenProcess o OpenProcessToken ▪ Winbase.h o LogonUser o LookupAccountSid o LookupAccountName o LookupPrivilegeValue ▪ Sddl.h o ConvertSidToStringSidA o ConvertSidToStringSidW ▪ Tlhelp32.h o Process32First

TOE SFR	Rationale
	○ Process32Next ○ CreateToolhelp32Snapshot
FPT_AEX_EXT.1	The compiler flags used to enable ASLR when the Cisco AnyConnect TOE is compiled is: /DYNAMICBASE The compiler flag used to enable stack-based buffer overflow protection in the Cisco AnyConnect TOE is: /GS
FPT_TUD_EXT.1 FPT_TUD_EXT.2 ALC_TSU_EXT.1	The TOE has specific versions that can be queried by a user. A TOE update is not a patch applied to the existing TOE, it is a new version of the TOE. When TOE updates are made available by Cisco, an administrator can obtain and install the update. Upon installation of a TOE update, a digital signature verification check will automatically be performed to ensure it has not been modified since distribution. The authorized source for the digitally signed updates is "Cisco Systems, Inc.". All Cisco communications relating to security issues are handled by the Cisco Product Security Incident Response Team (PSIRT). Cisco aims to provide fixes in 30 days but depending on the timing it may be greater than 30 days though not more than 60 days for most security issues. Fixes may be delayed longer for low-risk security issues. Updates are then made available at Cisco Software Central available at: https://software.cisco.com. Customers can subscribe to the Cisco Notification Service allows users to subscribe and receive important information regarding product updates. Full information is provide in the Cisco Security Vulnerability Policy available at: https://tools.cisco.com/security/center/resources/security_vulnerability_policy.html
FPT_LIB_EXT.1	The Cisco Secure Client-AnyConnect TOE is packaged with the following third-party libraries: ▪ OpenSSL ▪ Boost ▪ Libcurl ▪ Rapidxml

TOE SFR	Rationale
FPT_IDV_EXT.1	The Cisco Secure Client-AnyConnect TOE uses a sequence-based versioning control system. The application uses the major.minor.build format for versioning control. For example: 4.10.00093. ▪ Major (4 in the example above) designates a release where significant new features are added. ▪ Minor (10 in the example above) designates a release where minor new features are added. ▪ Build (00093 in the example above) designates a software build number.

FPT_TST_EXT.1	As a software product incorporating a cryptographic module, the TOE runs a suite of self-tests during start-up to verify its correct operation. These tests include: ▪ AES Known Answer Test – For the encrypt test, a known key is used to encrypt a known plain text value resulting in an encrypted value. This encrypted value is compared to a known encrypted value to ensure that the encrypt operation is working correctly. The decrypt test is just the opposite. In this test a known key is used to decrypt a known encrypted value. The resulting plaintext value is compared to a known plaintext value to ensure that the decrypt operation is working correctly. ▪ RSA Signature Known Answer Test (both signature/verification) – This test takes a known plaintext value and Private/Public key pair and used the public key to encrypt the data. This value is compared to a known encrypted value to verify that encrypt operation is working properly. The encrypted data is then decrypted using the private key. This value is compared to the original plaintext value to ensure the decrypt operation is working properly. ▪ ECDSA Signature Test – This test takes a known plaintext value and Private/Public key pair and used the public key to encrypt the data. This value is compared to a known encrypted value to verify that encrypt operation is working properly. The encrypted data is then decrypted using the private key. This value is compared to the original plaintext value to ensure the decrypt operation is working properly. ▪ HMAC Known Answer Test– For each of the hash values (256 and 384), the HMAC implementation is fed known plaintext data and a known key. These values are used to generate a MAC. This MAC is compared to a known MAC to verify that the HMAC and hash operations are operating correctly. ▪ SHA Known Answer Test – For each of the values (256 and 384), the SHA implementation is fed known data and key. These values are used to generate a hash. This hash is compared to a known value to verify they match and the hash operations are operating correctly. ▪ Software Integrity Test - The Software Integrity Test is run automatically whenever the module is loaded and confirms the image has maintained its integrity. If any self-test fails subsequent invocation of any cryptographic function calls is prevented. If all components of the power-up self-test are successful then the product is in FIPS mode. Integrity verification is performed each time the AnyConnect app is loaded and it will wait for the integrity verification to complete. Cryptographic services provided by the TOE platform are invoked to verify the digital signature of the TOE's executable files.

TOE SFR	Rationale
	If the integrity verification fails to successfully complete, the GUI will not load, rendering the app unusable. If the integrity verification is successful, the app GUI will load and operate normally. These tests are sufficient to verify that the TOE software is operating correctly as well as the cryptographic operations are all performing as expected.
FTP_DIT_EXT.1	The Cisco Secure Client-AnyConnect TOE itself is the application and does not maintain any sensitive data of its own. Therefore, there is no need to protect (through FTP_DIT_EXT.1.1) VPN-client-specific data.

7. CAVP Certificates

The table below lists the CAVP certificates for the TOE

Table 16. CAVP Certificates

SFR	Selection	Algorithm	Certificate Number	CPU
FCS_CKM.1.1/AK	P-256 P-384	ECDSA KeyGen and KeyVer	A1420 (Cisco)	Intel Core i5-1135G7 (Tiger Lake)
FCS_CKM.2.1	P-256 P-384	ECC Key Establishment(KAS-ECC Component)	A1420 (Cisco)	Intel Core i5-1135G7 (Tiger Lake)
FCS_COP.1/SKC	128-bit 256-bit	AES-CBC Encrypt/Decrypt AES-GCM Encrypt/Decrypt	A1420 (Cisco)	Intel Core i5-1135G7 (Tiger Lake)
FCS_COP.1/Hash	SHA-256 SHA-384	SHS	A1420 (Cisco)	Intel Core i5-1135G7 (Tiger Lake)
FCS_COP.1/Sig	RSA schemes using cryptographic key sizes of 2048-bits	RSA SigGen and SigVer	A1420 (Cisco)	Intel Core i5-1135G7 (Tiger Lake)
	ECDSA schemes using “NIST curves” P-256, P-384	ECDSA SigGen and SigVer		Intel Core i5-1135G7 (Tiger Lake)

SFR	Selection	Algorithm	Certificate Number	CPU
FCS_COP.1/ KeyedHash	HMAC-SHA-256 HMAC-SHA-384	HMAC	A1420 (Cisco)	Intel Core i5-1135G7 (Tiger Lake)

The functionality for following cryptographic SFRs is satisfied by the platform from the listing referenced in section 1.5.1:

- FCS_CKM.1.1/VPN
- FCS_RBG_EXT.1

8. References

The documentation listed below was used to prepare this ST

Table 17. References

Identifier	Description
[CC_PART1]	Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and general model, dated September 2012, version 3.1, Revision 5, CCMB-2017-04-001
[CC_PART2]	Common Criteria for Information Technology Security Evaluation – Part 2: Security functional components, dated September 2012, version 3.1, Revision 5, CCMB-2017-04-002
[CC_PART3]	Common Criteria for Information Technology Security Evaluation – Part 3: Security assurance components, dated September 2012, version 3.1, Revision 5, CCMB-2017-04-003
[CEM]	Common Methodology for Information Technology Security Evaluation – Evaluation Methodology, dated September 2012, version 3.1, Revision 5, CCMB-2017-04-004
[PP_APP_V1.4]	Protection Profile for Application Software Version 1.4, 18 October 2021.
[MOD_VPNC_V2.4]	PP-Module for VPN Client Version 2.4, 31 March 2022
[SD]	Supporting Document – PP-Module for Virtual Private Network (VPN) Client, Version 2.4, 31 March 2022

8.1. Acronyms and Terms

The following acronyms and terms are common and may be used in this Security Target.

Table 18. Acronyms and Terms

Acronym/Term	Definition
AES	Advanced Encryption Standard
CC	Common Criteria for Information Technology Security Evaluation
CEM	Common Evaluation Methodology for Information Technology Security
CM	Configuration Management
DRBG	Deterministic Random Bit Generator
EAL	Evaluation Assurance Level
EC-DH	Elliptic Curve-Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
ESP	Encapsulating Security Payload
GCM	Galois Counter Mode
HMAC	Hash Message Authentication Code
IKE	Internet Key Exchange
IPsec	Internet Protocol Security
IT	Information Technology
NGE	Next Generation Encryption
OS	Operating System
PP	Protection Profile
PRF	Pseudo-Random Functions
RFC	Request For Comment
SHS	Secure Hash Standard
SPD	Security Policy Database
ST	Security Target
TCP	Transport Control Protocol
TIMA	TrustZone Integrity Measurement Architecture

TOE	Target of Evaluation
TSC	TSF Scope of Control
TSF	TOE Security Function
TSP	TOE Security Policy
UDP	User datagram protocol
VPN	Virtual Private Network
AES	Advanced Encryption Standard

8.2. Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, using the Cisco Bug Search Tool (BST), submitting a service request, and gathering additional information, see *What's New in Cisco Product Documentation*.

To receive new and revised Cisco technical content directly to your desktop, you can subscribe to the What's New in Cisco Product Documentation RSS feed. The RSS feeds are a free service.

8.3. Contacting Cisco

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco website at www.cisco.com/go/offices.