Unisys Stealth Solution Release v4.0 Windows and Linux Endpoints Security Target

Version 1.0
3 December 2019

Prepared for:

UNISYS

801 Lakeview Drive
Blue Bell, PA 19422

Prepared By:

leidos

Accredited Testing & Evaluation Labs
6841 Benjamin Franklin Drive
Columbia, MD 21046

Table of Contents

List of Tables

1. Introduction

This section introduces the Target of Evaluation (TOE) and provides the Security Target (ST) and TOE identification, ST and TOE conformance claims, ST conventions, glossary and list of abbreviations.

The TOE is the Unisys Stealth Windows Endpoint 4.0 and Unisys Stealth Linux Endpoint 4.0. The Stealth endpoint for Windows and Linux operating systems provides capabilities for protected transmission of private data between Stealth-enabled IPsec VPN endpoints.

The Security Target contains the following additional sections:

- TOE Description (Section 2)—provides an overview of the TOE and describes the physical and logical boundaries of the TOE
- Security Problem Definition (Section 3)—describes the threats and assumptions that define the security problem to be addressed by the TOE and its environment
- Security Objectives (Section 4)—describes the security objectives for the TOE and its operational environment necessary to counter the threats and satisfy the assumptions that define the security problem
- IT Security Requirements (Section 5)—specifies the security functional requirements (SFRs) and security assurance requirements (SARs) to be met by the TOE
- TOE Summary Specification (Section 6)—describes the security functions of the TOE and how they satisfy the SFRs
- Protection Profile Claims (Section 7)—provides rationale for the consistency of the ST with the PPs to which conformance is claimed
- Rationale (Section 8)—provides mappings and rationale for the security problem definition, security objectives, security requirements, and security functions to justify their completeness, consistency, and suitability.

1.1 Security Target, TOE and CC Identification

ST Title – Unisys Stealth Solution Release v4.0 Windows and Linux Endpoints Security Target

ST Version – 1.0

ST Date – 3 December 2019

TOE Identification – Unisys Stealth Solution Release 4.0.026.0 Windows Endpoint
Unisys Stealth Solution Release 4.0.026.0 Linux Endpoint

TOE Developer – Unisys Corporation

Evaluation Sponsor – Unisys Corporation

CC Identification – Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, April 2017

1.2 Conformance Claims

This ST and the TOE it describes are conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Components, Version 3.1, Revision 5, April 2017
  - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security Assurance Components, Version 3.1, Revision 5, April 2017

- Part 3 Extended

This TOE and ST are conformat to the PP-Configuration for Application Software and Virtual Private Network (VPN) Clients, Version 1.0, 04 September 2019 (CFG_APP_VPNC_V1.0).

This PP-Configuration includes the following components:

- Base-PP: Protection Profile for Application Software, Version 1.3 (PP_APP_V1.3)
- PP-Module: PP-Module for Virtual Private Network (VPN) Clients, Version 2.1 (MOD_VPNC_V2.1)

Protection Profile for Application Software, Version 1.3, 1 March 2019 [PP_APP_V1.3]. The following NIAP Technical Decisions apply to this PP and have been accounted for in the ST development and the conduct of the evaluation:

- TD0465: Configuration Storage for .NET Apps
- TD0445: User Modifiable File Definition
- TD0444: IPsec selections
- TD0437: Supported Configuration Mechanism
- TD0435: Alternative to SELinux for FPT_AEX_EXT.1.3
- TD0434: Windows Desktop Applications Test
- TD0427: Reliable Time Source
- TD0416: Correction to FCS_RBG_EXT.1 Test Activity

PP-Module for Virtual Private Network (VPN) Clients, Version 2.1, 5 October 2017 [MOD_VPNC_V2.1]. The following NIAP Technical Decisions apply to this PP and have been accounted for in the ST development and the conduct of the evaluation:

- TD0404: Cryptographic selections and updates for use with App PP v1.3
- TD0385: FTP_DIT_EXT.1 Assurance Activity Clarification
- TD0379: Updated FCS_IPSEC_EXT.1.11 Tests for VPN Client
- TD0378: TOE/TOE Platform Selection in FCS_IPSEC_EXT.1 SFRs
- TD0355: FCS_CKM.1/VPN for IKE authentication

1.3 Conventions

The following conventions have been applied in this document:

- Security Functional Requirements—Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
    - Iteration: allows a component to be used more than once with varying operations. In this ST, iteration is indicated by a number in parentheses placed at the end of the component. For example, FMT_MTD.1(1) and FMT_MTD.1(2) indicate that the ST includes two iterations of the FMT_MTD.1 requirement.
    - Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]). Note that an assignment within a selection would be identified in italics and with embedded bold brackets (e.g., [***[selected-assignment]***]).
    - Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., [***selection***]).
    - Refinement: allows the addition of details. Refinements are indicated using bold for additions and double strike-through for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …").

Refinements in this ST are used to distinguish between the different platform operating systems (e.g. Windows or Linux).

  - Operations performed on the SFRs in [PP_APP_V1.3] and [MOD_VPNC_V2.1] (or in TDs that modify the SFRs) are not specifically identified in the SFRs in [ST]. Instead, the ST reproduces the text of operations completed in the PPs (or TDs) without embellishment.

- Other sections of the ST—other sections of the ST use bolding to highlight text of special interest, such as captions.

1.4 Glossary

This ST uses a number of terms that have a specific meaning within the context of the ST and the TOE. This glossary provides a list of those terms and how they are to be understood within this ST.

Authorization Service	Authorizes endpoints and distributes the COIs and filter material that the administrator defines for users and groups.
Community of Interest (COI)	The TOE enables multiple "secure communities" to share the same network without fear of another group accessing their data or their workstations and servers. These are referred to as Communities of Interest (COIs).
crypto.xml File	The crypto.xml installed with the endpoint package contains the profiles that the endpoint should support.
Enterprise Manager	A management application that allows administrators to create and manage COIs, provision and monitor Stealth endpoints, and authorize, monitor, and license Stealth users. The Enterprise Manager includes the Authorization Service.
Management Server	A server in the operational environment of the TOE that hosts the Enterprise Manager application (which also includes the Stealth Authorization Service).
protectionprofile.xml File	This file contains the settings that are configured to enable Stealth to conform to the Protection Profile for Application Software as well as the PP-Module for VPN Clients. The protectionprofile.xml file is used in the process of creating endpoint packages. During the endpoint package generation process, the Enterprise Manager reads the protectionprofile.xml file, validates it, and inserts the contents into the crypto.xml file.
Secure Community of Interest Protocol (SCIP)	A Unisys proprietary protocol used in conjunction with IPsec for communication among all Stealth endpoints.
Signing Certificate	A signing certificate is used to protect the integrity of the settings.xml and crypto.xml files, which are included in the endpoint software package.
Stealth administrator	An administrator that uses the Stealth Enterprise Manager to configure Stealth endpoint installation packages.
Stealth endpoint	A server or workstation running Stealth endpoint software. The TOE is stealth endpoint software specifically for Windows and Linux platforms.
Tuples	For Stealth endpoints, a set of COIs, their associated filter sets, and any clear text filters for which the user is authorized.

1.5 Abbreviations and Acronyms

The following abbreviations and acronyms are used in this document. A brief definition is provided for abbreviations that are potentially unfamiliar, are specific to the TOE, or not obviously self-explanatory.

AES	Advanced Encryption Standard
CA	Certificate Authority
CBC	Cipher Block Chaining—an AES mode of operation
CC	Common Criteria
COI	Community of Interest (refer to Glossary for definition)
CRL	Certificate Revocation List
CSP	Critical Security Parameter—security-related information whose disclosure or modification can compromise the security of a cryptographic module
DH	Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
ESP	Encapsulating Security Payload—a member of the IPsec protocol suite providing origin authenticity, integrity and confidentiality protection of packets
FIPS	Federal Information Processing Standard
FQDN	Fully Qualified Domain Name
FSF	Free Software Foundation
GCM	Galois/Counter Mode—an AES mode of operation
HMAC	Keyed-Hash Message Authentication Code
IKE	Internet Key Exchange
IPsec	Internet Protocol security
MAC	Message Authentication Code
NIST	National Institute of Standards and Technology
RFC	Request For Comments—an Internet Engineering Task Force memorandum on Internet standards and protocols
SA	Security Association
SAR	Security Assurance Requirement
SCIP	Secure Community of Interest Protocol (refer to Glossary for definition)
SPD	Security Policy Database—packet filtering rules in an IPsec implementation
SHA	Secure Hash Algorithm
SFR	Security Functional Requirement
ST	Security Target
TCP/IP	Transmission Control Protocol/Internet Protocol
TOE	Target of Evaluation
TSF	TOE Security Functionality
VPN	Virtual Private Network
WFP	Windows Filtering Platform
XML	Extensible Markup Language

2. TOE Description

The TOE is the Unisys Stealth Solution Release 4.0 Windows Endpoint and the Unisys Stealth Solution Release 4.0 Linux Endpoint. They are the components of the Unisys Stealth Solution that enables endpoint devices to establish secure IPsec tunnels with each other.

2.1 Overview

The Unisys Stealth Solution is an enterprise networking security solution that utilizes IPsec to protect the confidentiality of data transmitted between devices on the enterprise network. Stealth enables multiple "secure communities" to share the same network while maintaining separation of data and devices. These are referred to as Communities of Interest (COIs).

Stealth achieves this through implementation of the proprietary Secure Community of Interest Protocol (SCIP) combined with standard IPsec. The SCIP/IPsec protocol is used to negotiate COI membership and cryptographic profiles in order to establish COI-based tunnels between Stealth-enabled endpoints. SCIP also controls the OS-native IKE and IPsec implementations used to setup, rekey, and transport data through encrypted tunnels. The operation of SCIP, and the subsequent configuration and management of the resulting IPsec tunnels, is completely transparent to the user and any applications running on the system.

The TOE comprises software installed on Windows-based servers, Windows workstations, and Linux servers that enables these devices to participate in the Stealth network as Stealth-enabled endpoints. The TOE functions as an IPsec VPN client that enables the endpoint on which it is installed to establish an IPsec tunnel with another Stealth-enabled endpoint belonging to the same Stealth COI. Note that the TOE implements a client-to-client model of operation—Stealth-enabled endpoints establish IPsec tunnels with each other rather than with a VPN gateway.

2.2 TOE Architecture

The Windows and Linux Endpoints comprising the TOE are part of the Unisys Stealth Solution, which encompasses the following additional components, briefly discussed here to aid understanding of the TOE:

- Enterprise Manager—a management application that allows administrators to create and manage COIs, provision and monitor Stealth endpoints, and authorize, monitor, and license Stealth users. The Enterprise Manager includes the Stealth Authorization Service.
- Stealth Secure Virtual Gateway—enables non-Windows and non-Linux endpoints to participate in Stealth networks.
- Secure Remote Access Gateway—enables systems that are physically located outside of the enterprise's intranet to participate in an enterprise's Stealth network.

The Windows and Linux Endpoints are provisioned as endpoint installation packages that are created by the Enterprise Manager in the TOE's operational environment. The Enterprise Manager can create endpoint packages specific for Windows 10 (32-bit and 64-bit), Windows Server 64-bit platforms, and Red Hat Enterprise Linux 64-bit operating systems. The installation package includes the Stealth endpoint software and configuration information that specifies the IKE and IPsec cryptographic profiles (termed "protection profiles" in the product guidance documentation) the TOE will use when negotiating an IPsec tunnel with another endpoint. The TOE establishes the IPsec tunnel in transport mode.

The configuration information is contained in two XML files—"protectionprofile.xml" and "crypto.xml". The protectionprofile.xml file contains the settings the administrator configures to enable the Windows and Linux Endpoints to conform to the PP-Module for Virtual Private Network (VPN) Clients.

The protectionprofile.xml file is used in the process of creating endpoint packages. During the endpoint package generation process, the Enterprise Manager reads the protectionprofile.xml file, validates it, and inserts the contents into the crypto.xml file. The Enterprise Manager signs the resulting crypto.xml file using a signing certificate and includes it in the endpoint package. The resulting endpoint package is then installed on the endpoints.

The endpoints on which the package is installed read the crypto.xml file and validate the signature of the signing certificate using the associated trusted root certificate, which is also installed on these endpoints.

After the signing certificate has been validated, the TOE is able to attempt to establish an IPsec tunnel using the information in the crypto.xml file. If the Protection Profile mode is enabled in the protectionprofile.xml file, the TOE uses the protection profiles specified in that file, rather than the pre-existing profiles defined in crypto.xml. Only the profiles specified in the protectionprofile.xml file are used for IPsec tunnel establishment, and they are used only in the priority order specified in the protectionprofile.xml file.

Endpoints are able to communicate using the first matching protection profile that they share (endpoints might be running Stealth endpoint software created with identical protectionprofile.xml profile priority lists, or they might have been created using a different profile priority list). The profile that is used during Stealth tunnel establishment depends on the profile priority list specified on the endpoint receiving the Stealth tunnel request.

2.2.1 TOE Physical Boundaries

The Common Criteria evaluation was performed on the following Stealth-supported endpoint operating systems:

- Windows 10 (32-bit and 64-bit)
- Windows Server 2016 (64-bit)
- Red Hat Enterprise Linux (RHEL) 7.4 (64-bit)
- Red Hat Enterprise Linux (RHEL) 7.5 (64-bit)

The physical boundary of the TOE comprises an endpoint installation package that is created by the Enterprise Manager software running on the Management Server. The installation package includes the Stealth endpoint software and configuration information that specifies the IKE and IPsec cryptographic profiles the TOE will use when negotiating an IPsec tunnel with another endpoint.

The Windows Stealth Endpoint v4.0.026.0 software comprises:

- Kernel-mode drivers (`mlstpgw.sys`, `stealthii.sys`)
- The following services:
    - Unisys Stealth Logon Service (`USSL_Logon`)
    - Unisys PreLogon Service (`USSL_PreLogon`)
    - Unisys Protocol Service (`USSL_Protocol`)

The Red Hat Linux 7.4 and Red Hat Linux 7.5 Unisys Stealth Linux Endpoint software v4.0.026.0 comprises the following:

- unisys-fips-libdavici-1.1.0-2
- unisys-libnetfilter-conntrack-1.0.5-1
- unisys-fips-libcurl-7.46.0-6
- unisys-fips-libxmlsec1-1.2.22-4
- unisys-fips-strongswan-5.6.3-5
- unisys-fips-stealth-4.0.026.0

All Windows endpoints on which the TOE is to be installed must have at least 2 GB of memory and must run the following software:

- .NET Framework version 3.5 with Service Pack 1 or .NET Framework version 4.x

All Linux endpoints on which the TOE is to be installed must meet the memory and disk space requirements for the specific Linux operating system and must run the following software:

- Red Hat Enterprise Linux (RHEL) 7.4
    - Linux kernel 3.10.0-693.el7
    - OpenSSL 1.0.2k-8.el7
- Red Hat Enterprise Linux (RHEL) 7.5

  - Linux kernel 3.10.0-862.2.3.el7
  - OpenSSL 1.0.2k-12.el7

Additionally, each endpoint on which the TOE is installed must be configured for FIPS mode.

The TOE requires the following in its operational environment:

- Management Server, which runs the services comprising the Enterprise Manager component
- Authorization Server, present on a Management Server.

In addition, the Management Server must be Stealth-enabled (i.e., the TOE is required to be installed on the Management Server as well as the VPN endpoints). The Stealth Management Server is configured to be FIPS mode enabled.

2.2.2 TOE Logical Boundaries

This section summarizes the security functions provided by the TOE.

- Cryptographic Support
- User Data Protection
- Identification and Authentication
- Security Management
- Privacy
- TSF Protection
- Trusted Channel/Path

2.2.2.1 Cryptographic Support

The TOE enables an end user to establish a point-to-point VPN tunnel with another Stealth-enabled endpoint, using the underlying platform's implementation of IKE and IPsec. The Unisys Windows Stealth Endpoint invokes the platform functionality to securely store domain credentials while the Unisys Linux Stealth Endpoint does not store any domain credentials.

2.2.2.2 User Data Protection

The TOE and the TOE platforms ensure that residual information is protected from potential reuse in accessible objects such as network packets. The Unisys Windows Stealth Endpoint does not store any sensitive data in non-volatile memory. The Unisys Linux Stealth Endpoint leverages the platform provided functionality to encrypt sensitive data.

The TOE restricts network communication to user-initiated communication for Stealth-tunneled network traffic to Unisys Stealth peers and communication to the Stealth Authorization server.

2.2.2.3 Identification and Authentication

The TOE provides the ability to use, store, and protect X.509v3 certificates. The TOE supports the use of X.509v3 certificates for IKE peer authentication and integrity verification. In addition, the TOE platform uses X.509v3 certificates.

2.2.2.4 Security Management

The TSF is capable of performing the following management functions:

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

- Specify IKEv2 SA lifetimes
- Configure packet filter rules
- Configure CRL checking
- Configure algorithm suites that can be proposed and accepted during IPsec exchanges.

2.2.2.5 Privacy

The Windows Unisys Stealth Endpoint and the Unisys Linux Unisys Stealth Endpoint application does not collect or transmit PII over a network.

2.2.2.6 TSF Protection

The TOE relies upon its underlying platform to perform self-tests that cover the TOE as well as the functions necessary to securely update the TOE. The TOE does not allocate any memory region with both write and execute permissions and is compiled with stack-based buffer overflow protection enabled. The TOE applications use only documented platform APIs.

2.2.2.7 Trusted Channel/Path

The TOE encrypts all transmitted sensitive data with IPsec between itself and another trusted IT product.

2.3 TOE Documentation

This section identifies the guidance documentation included in the TOE:

- *Unisys Stealth Solution Common Criteria Evaluation Guidance Document*, Release 4.0, October 24, 2019 (8205 5823–005)
- *Unisys Stealth Solution Information Center*, Release 4.0, April 2019, (8222 4189-013)

3. Security Problem Definition

This ST includes by reference the Security Problem Definition (comprising threat statements and assumptions) from the *Protection Profile for Application Software*, Version 1.3, 01 March 2019 and the *PP-Module for Virtual Private Network (VPN) Clients,* Version 2.1, 5 October 2017. The A.PLATFORM was updated per NIAP Technical Decision TD0427: Reliable Time Source.

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.

The PP offers additional information about the identified threats, but that has not been reproduced here and the PP should be consulted if there is interest in that material.

In general, the PP has presented a Security Problem Definition appropriate for application software providing IPsec VPN client functionality, and as such is applicable to the Unisys Stealth Windows VPN Client and the Unisys Stealth Linux Endpoint.

4. Security Objectives

This ST includes by reference the Security Objectives for the TOE specified in the *Protection Profile for Application Software, Version 1.3, 01 March 2019* and the *PP-Module for Virtual Private Network (VPN) Clients, Version 2.1, 5 October 2017*. The security objectives for the operational environment are reproduced below, since these objectives characterize technical and procedural measures each consumer must implement in their operational environment.

4.1 Security Objectives for the Operational Environment

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. IT Security Requirements

This section defines the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) that represent the security claims for the Target of Evaluation (TOE) and scope the evaluation effort.

All the SFRs have been drawn from the *Protection Profile for Application Software,* Version 1.3, 01 March 2019 [PP_APP_V1.3], and the *PP-Module for Virtual Private Network (VPN) Clients,* Version 2.1, 5 October 2017 [MOD_VPNC_V2.1]. As such, operations already performed in that PP are not identified here. Instead, the requirements have been copied from the PP and any incomplete selections or assignments have been performed herein. Of particular note, the PP makes a number of refinements and completes some SFR operations defined in the CC, so it should be consulted if necessary to identify those changes.

The SARs are the set of SARs specified in [PP_APP_V1.3].

5.1 Extended Requirements

All of the extended requirements in this ST have been drawn from the [PP_APP_V1.3] and the [MOD_VPNC_V2.1]. The [PP_APP_V1.3] and the [MOD_VPNC_V2.1] defines the following extended SFRs and since they are not redefined in this ST, the [PP_APP_V1.3] and the [MOD_VPNC_V2.1] should be consulted for more information in regard to those CC extensions.

- FCS_CKM.1/VPN: Cryptographic Key Generation (IKE)
- FCS_CKM_EXT.1: Cryptographic Key Generation Services
- FCS_CKM_EXT.2: Cryptographic Key Storage
- FCS_CKM_EXT.4: Cryptographic Key Destruction
- FCS_IPSEC_EXT.1: IPsec
- FCS_RBG_EXT.1: Random Bit Generation Services
- FCS_STO_EXT.1(1): Storage of Credentials
- FCS_STO_EXT.1(2): Storage of Credentials
- FDP_DAR_EXT.1(1): Encryption Of Sensitive Application Data
- FDP_DAR_EXT.1(2): Encryption Of Sensitive Application Data
- FDP_DEC_EXT.1: Access to Platform Resources
- FDP_NET_EXT.1: Network Communications
- FIA_X509_EXT.1: X.509 Certificate Validation
- FIA_X509_EXT.2: X.509 Certificate Authentication
- FMT_CFG_EXT.1: Secure by Default Configuration
- FMT_MEC_EXT.1: Supported Configuration Mechanism
- FPR_ANO_EXT.1: User Consent for Transmission of Personally Identifiable Information
- FPT_AEX_EXT.1: Anti-Exploitation Capabilities
- FPT_API_EXT.1: Use of Supported Services and APIs
- FPT_IDV_EXT.1: Software Identification and Versions
- FPT_LIB_EXT.1(1): Use of Third Party Libraries
- FPT_LIB_EXT.1(2): Use of Third Party Libraries
- FPT_TST_EXT.1: TSF Self-Test

- FPT_TUD_EXT.1: Integrity for Installation and Update
- FPT_TUD_EXT.2 Integrity for Installation and Update
- FTP_DIT_EXT.1: Protection of Data in Transit
- ALC_TSU_EXT.1: Timely Security Updates

5.2 Security Functional Requirements

This section specifies the SFRs for the TOE.

Requirement Class	Requirement Component
FCS: Cryptographic Support	FCS_CKM.1(1): Cryptographic Asymmetric Key Generation
	FCS_CKM.1/VPN: Cryptographic Key Generation (IKE)
	FCS_CKM_EXT.1: Cryptographic Key Generation Services
	FCS_CKM.2: Cryptographic Key Establishment
	FCS_CKM_EXT.2: Cryptographic Key Storage
	FCS_CKM_EXT.4: Cryptographic Key Destruction
	FCS_COP.1(1): Cryptographic Operation – Encryption/Decryption
	FCS_COP.1(2): Cryptographic Operation - Hashing
	FCS_COP.1(3): Cryptographic Operation - Signing
	FCS_COP.1(4): Cryptographic Operation - Keyed-Hash Message Authentication
	FCS_IPSEC_EXT.1: IPsec
	FCS_RBG_EXT.1: Random Bit Generation Services
	FCS_STO_EXT.1(1): Storage of Credentials
	FCS_STO_EXT.1(2): Storage of Credentials
FDP: User Data Protection	FDP_DAR_EXT.1(1): Encryption Of Sensitive Application Data
	FDP_DAR_EXT.1(2): Encryption Of Sensitive Application Data
	FDP_DEC_EXT.1: Access to Platform Resources
	FDP_NET_EXT.1: Network Communications
	FDP_RIP.2(1): Full residual information protection
	FDP_RIP.2(2): Full residual information protection
FIA: Identification and Authentication	FIA_X509_EXT.1: X.509 Certificate Validation
	FIA_X509_EXT.2: X.509 Certificate Authentication
FMT: Security Management	FMT_CFG_EXT.1: Secure by Default Configuration
	FMT_MEC_EXT.1: Supported Configuration Mechanism
	FMT_SMF.1/VPN : Specification of Management Functions (VPN)
	FMT_SMF.1: Specification of Management Functions

Requirement Class	Requirement Component
FPR: Privacy	FPR_ANO_EXT.1: User Consent for Transmission of Personally Identifiable Information
FPT: Protection of the TSF	FPT_AEX_EXT.1: Anti-Exploitation Capabilities
	FPT_API_EXT.1: Use of Supported Services and APIs
	FPT_IDV_EXT.1: Software Identification and Versions
	FPT_LIB_EXT.1(1): Use of Third Party Libraries
	FPT_LIB_EXT.1(2): Use of Third Party Libraries
	FPT_TST_EXT.1: TSF self test
	FPT_TUD_EXT.1: Integrity for Installation and Update
	FPT_TUD_EXT.2: Integrity for Installation and Update
FTP: Trusted path/channels	FTP_DIT_EXT.1: Protection of Data in Transit

Table 1: TOE Security Functional Components

5.2.1 Cryptographic Support (FCS)

FCS_CKM.1(1) – Cryptographic Asymmetric Key Generation

FCS_CKM.1.1[1] The application shall [***invoke platform-provided 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 *[**no other curves**]*] that meet the following: [FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.4];

[

- ***[FFC Schemes] using Diffie-Hellman group 14 that meet the following: [RFC 3526, Section 3]];***

[***no other key generation methods***

].

FCS_CKM_EXT.1 – Cryptographic Key Generation Services

FCS_CKM_EXT.1.1[2] The application shall [
invoke platform-provided functionality for asymmetric key generation
] .

FCS_CKM.1/VPN– Cryptographic Key Generation (IKE)

FCS_CKM.1.1/VPN[3] The application 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.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.

[1] Modified per TD0404
[2] Modified per TD0404
[3] Modified per TD0355

FCS_CKM.2 – Cryptographic Key Establishment

FCS_CKM.2.1[4] The application shall [***invoke platform-provided 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

- [***Key establishment scheme using Diffie-Hellman group 14 that meets the following: RFC 3526, Section 3***]; and
- [***No other schemes.***]

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.

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.

FCS_COP.1(1) Cryptographic Operation (Encryption and Decryption

FCS_COP.1.1(1) The application shall perform encryption/decryption in accordance with a specified cryptographic algorithm

- AES-CBC (as defined in NIST SP 800-38A) mode; and
- [AES-GCM (as defined in NIST SP 800-38D)]

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

FCS_COP.1(2) Cryptographic Operation - Hashing

FCS_COP.1.1(2) The application shall perform cryptographic hashing services in accordance with a specified cryptographic algorithm [

- ***SHA-256,***
- ***SHA-384,***
- ***SHA-512***

] and message digest sizes [

- ***256,***
- ***384,***
- ***512***

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

FCS_COP.1(3) Cryptographic Operation - Signing

FCS_COP.1.1(3) 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 ,***
- ***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***

] .

[4] Modified per TD0404

FCS_COP.1(4) Cryptographic Operation - Keyed-Hash Message Authentication

FCS_COP.1.1(4) The application shall perform keyed-hash message authentication in accordance with a specified cryptographic algorithm

- HMAC-SHA-256

and [

- ***SHA-384***

] with key sizes [**512 bits for SHA-256, 1024 bits for SHA-384**] and message digest sizes 256 and [***384***] bits that meet the following: FIPS Pub 198-1 The Keyed-Hash Message Authentication Code and FIPS Pub 180-4 Secure Hash Standard.

FCS_IPSEC_EXT.1 – IPsec

FCS_IPSEC_EXT.1.1 The [***TOE, TOE platform***] shall implement the IPsec architecture as specified in RFC 4301.

FCS_IPSEC_EXT.1.2 The [***TOE***] shall implement [***transport mode***].

FCS_IPSEC_EXT.1.3 The [***TOE***] 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 [***TOE, TOE platform***] 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, [***no other algorithms***].

FCS_IPSEC_EXT.1.5 The [***TOE, TOE platform***] shall implement the protocol:[

- ***IKEv2 as defined in RFCs 7296 (with mandatory support for NAT traversal as specified in section 2.23), 4307,*** and [***no other RFCs for hash functions***]].

FCS_IPSEC_EXT.1.6 The [***TOE***] 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 [***no other algorithm***].

FCS_IPSEC_EXT.1.7 The [***TOE, TOE platform***] shall ensure that [***IKEv2 SA lifetimes can be configured by [an Administrator] 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[5] The [***TOE***] shall ensure that all IKE protocols implement DH groups 14 (2048-bit MODP), 19 (256-bit Random ECP), 20 (384-bit Random ECP), and [***no other DH groups***].

FCS_IPSEC_EXT.1.9[6] The [***TOE platform***] 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 [**224 for DH Group 14, 256 for DH Group 19, 384 for DH Group 20**] bits.

FCS_IPSEC_EXT.1.10[7] The [***TOE platform***] 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^[**112 for DH Group 14, 128 for DH Group 19, 192 for DH Group 20**].

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

[5] Modified per TD0404

[6] Modified per TD0404

[7] Modified per TD0404

FCS_IPSEC_EXT.1.12[8] The [***TOE Platform***] shall not establish an SA if the [***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[9] The [***TOE Platform***] 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 [***TOE***] 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.

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.

FCS_STO_EXT.1 – Storage of Credentials

FCS_STO_EXT.1.1(1) The **Unisys Windows Stealth Endpoint** application shall [***invoke the functionality provided by the platform to securely store [domain credentials for USSL logon service]***
] to non-volatile memory.

FCS_STO_EXT.1.1(2) The **Unisys Linux Stealth Endpoint** application shall [***not store any credentials***] to non-volatile memory.

5.2.2 User Data Protection (FDP)

FDP_DAR_EXT.1 – Encryption Of Sensitive Application Data

FDP_DAR_EXT.1.1(1) The **Unisys Windows Stealth Endpoint** application shall **[*not store any sensitive data*]** in non-volatile memory.

FDP_DAR_EXT.1.1(2) The **Unisys Linux Stealth Endpoint** application shall **[*leverage platform-provided functionality to encrypt sensitive data*]** in non-volatile memory.

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 [

system logs

] .

[8] Modified per TD0378

[9] Modified per TD0378

FDP_NET_EXT.1 – Network Communications

FDP_NET_EXT.1.1 The application shall restrict network communication to

[

user-initiated communication for [Stealth-tunneled network traffic to Unisys Stealth peers]

[Communication to the Stealth Authorization server]

] .

FDP_RIP.2 – Full Residual Information Protection

FDP_RIP.2.1(1) The [***TOE***] shall enforce that any previous information content of a resource is made unavailable upon the [***deallocation of the resource from***] all objects.

FDP_RIP.2.1(2) 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.

5.2.3 Identification and Authentication (FIA)

FIA_X509_EXT.1 – Extended: X.509 certificate

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 and that the CA flag is set to TRUE for all CA certificates.
- The application shall validate the revocation status of the certificate using [***a Certificate Revocation List (CRL) as specified in RFC 5759***].
- The application shall validate the extendedKeyUsage 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 extendedKeyUsage 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 extendedKeyUsage 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 extendedKeyUsage 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 extendedKeyUsage 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 extendedKeyUsage 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.

Application Note: *The TOE uses certificates in support of trusted update and integrity verification while the TOE platform uses certificates in support of integrity verification.*

FIA_X509_EXT.2 – X.509 Certificate Authentication

FIA_X509_EXT.2.1[10] The application shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [**IPsec**].

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

5.2.4 Security Management (FMT)

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.

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*]**.[11]

FMT_SMF.1 – Specification of Management Functions

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

[

[listed in FMT_SMF.1.1/VPN]

].

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 IPsec VPN Clients to use for connections***]
- Specify client credentials to be used for connections,
- Configure the reference identifier of the peer
- ***[specify IKEv2 SA lifetimes***
- ***configure packet filter rules***
- ***configure CRL checking***
- ***configure algorithm suites that can be proposed and accepted during IPsec exchanges]***].

Application Note: *The TOE implements a client-to-client model of operation. The TOE establishes IPsec tunnels with other TOE-equipped clients rather than with a VPN gateway. Therefore, the security management function implemented by the TOE is to specify VPN clients to use for connections.*

[10] Modified per TD0444

[11] Modified per TD0437

5.2.5 Privacy (FPR)

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

].

5.2.6 Protection of the TSF (FPT)

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.

FPT_API_EXT.1 – Use of Supported Services and APIs

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

FPT_IDV_EXT.1 Software Identification and Versions

FPT_IDV_EXT.1.1 The application shall be versioned with [[***Unisys internal versioning control***]].

FPT_LIB_EXT.1 – Use of Third Party Libraries

FPT_LIB_EXT.1.1(1) The **Unisys Stealth Windows Endpoint** application shall be packaged with only [
Microsoft Visual C++ 2015 Redistributable (x64) or Microsoft Visual C++ 2015 Redistributable (x86) depending of version of Windows (x64) vs. (x86), ChilkatDotnet2.dll
].

FPT_LIB_EXT.1.1(2) The **Unisys Stealth Linux Endpoint** application shall be packaged with only [
no third party libraries
].

FPT_TST_EXT.1 – TSF Self-Test

FPT_TST_EXT.1.1(1) The **Windows** [***TOE platform***] 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(1) The **Windows** [***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 signatures signed using X.509 certificates**].

FPT_TST_EXT.1.1(2) The **Linux** [***TOE platform***] 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(2) The **Linux** [***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 signatures using SHA256 and RSA 4096-bit key pairs generated by the TOE platform**].

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, leverage the platform***] 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 installation package and its updates shall be digitally signed such that its 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***].

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.

5.2.7 Trusted Path/Channels (FTP)

FTP_DIT_EXT.1.1 – Protection of Data in Transit

FTP_DIT_EXT.1.1[12] The application shall [***encrypt all transmitted sensitive data, data***] ***with*** [***IPsec as defined in the PP-Module for VPN Client***] between itself and another trusted IT product.

5.3 Security Assurance Requirements

This section specifies the SARs for the TOE. The SARs are included by reference from [VPNPP].

Requirement Class	Requirement Component
ADV: Development	ADV_FSP.1: Basic functional specification
AGD: Guidance documents	AGD_OPE.1: Operational user guidance
	AGD_PRE.1: Preparative procedures
ALC: Life-cycle support	ALC_CMC.1: Labelling of the TOE
	ALC_CMS.1: TOE CM coverage
	ALC_TSU_EXT.1 Timely Security Updates
ATE: Tests	ATE_IND.1: Independent testing – conformance
AVA: Vulnerability assessment	AVA_VAN.1: Vulnerability survey

[12] Modified per TD0404

Table 2: Assurance Components

6. TOE Summary Specification

This chapter describes the following security functions implemented by the TOE or its underlying platform to satisfy the SFRs claimed in Section 5.2 of this ST:

- Cryptographic Support
- User Data Protection
- Identification and Authentication
- Security Management
- Privacy
- TSF Protection
- Trusted Channel/Path.

As described in Section 2.2.1, the TOE comprises an endpoint installation package that is created by the Enterprise Manager software in the TOE's operational environment. The installation package includes the Stealth Endpoint software and configuration information that specifies the IKE and IPsec cryptographic profiles the TOE will use when negotiating an IPsec tunnel with another endpoint. The cryptographic profiles are termed "protection profiles" in the TOE's guidance documentation.

6.1 Cryptographic Support

6.1.1 FCS_CKM_EXT.1

The Unisys Stealth Windows Endpoint and the Unisys Stealth Linux Endpoint each invoke the respective platform-provided functionality for asymmetric key generation.

6.1.2 FCS_CKM.1(1), FCS_CKM.1/VPN

The application invokes platform-provided functionality for asymmetric key generation. The key generation functionality is invoked through calls to the relevant crypto API.

The Unisys Stealth Windows Endpoint relies on the underlying Windows platform for generation of asymmetric keys used for key establishment purposes and for IKE peer authentication. The Kernel Mode Cryptographic Primitives Library (cng.sys) cryptomodule included in each of the underlying platforms is invoked for this purpose. The Stealth Windows endpoint software calls the function BCryptGenerateKeyPair to invoke this functionality.

The Unisys Stealth Linux Endpoint relies on OpenSSL libcrypto for generation of asymmetric keys used for key establishment purposes and strongSwan for IKE peer authentication. strongSwan is configured to also use OpenSSL. Asymmetric key generation is done via OpenSSL by strongSwan for IKE and negotiation among all Stealth endpoints.

strongSwan calls the following OpenSSL APIs to generate keys:

- EC_KEY_generate_key() - Called to generate public/private key pairs used during IKEv2 negotiation and key exchanges. Use of this function versus DH_generate_key() is based on the Diffie Hellman group of the selected protection profile. This function is used for Diffie Hellman groups 19 and 20.
- DH_generate_key_ex() - Called to generate public/private key pairs used during IKEv2 negotiation and key exchanges. Use of this function versus EC_KEY_generate_key() is based on the Diffie Hellman group of the selected protection profile. This function is used for Diffie Hellman group 14.

The kernel is not involved in key generation beyond supplying bits via /dev/urandom.

The table below identifies the asymmetric key generation schemes used by the TOE, the purpose of the key, and the key sizes.

Scheme	Scheme Usage	Key Sizes
ECC schemes using "NIST curves" P-256 and P-384 that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4.	This key pair is used for signing and validating blocks of data that use the SCIP/IPsec protocol. (The signing algorithm used is the EC Digital Signing Algorithm.)	256-bit Elliptic Curve (EC) keys
	The IKE negotiation certificate must be signed using the Elliptic Curve Digital Signature Algorithm (ECDSA) using a 256-bit or 384-bit private key.	256, 384-bit
FFC schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.	IKE protocols	2048

Table 3 – Platform Provided Asymmetric Key Generation

6.1.3 FCS_CKM.2

The Windows Stealth Endpoints and the Linux Stealth Endpoints invoke platform-provided functionality to perform cryptographic key establishment in accordance with specified cryptographic key establishment methods:

- Elliptic curve-based key establishment schemes that meet NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"
- Key establishment scheme using Diffie-Hellman group 14 that meets RFC 3526, Section 3

The Unisys Stealth Windows Endpoints and the Unisys Stealth Linux Endpoints support the DH group 14 key establishment scheme that meets standard RFC 3526, section 3 for interoperability.

The key usage for each scheme is identified in **Table 3**.

6.1.4 FCS_CKM_EXT.2

The Table below lists the keys and critical security parameters (CSPs) managed by the underlying Windows platform that are needed to meet the requirements in this ST.

Security Relevant Data Item	Description	How Stored?	Persistent?
Symmetric encryption/decryption keys	Keys used for AES encryption/decryption for IKEv2 and IPsec ESP.	Unencrypted	N
HMAC keys	Keys used for HMAC-SHA-256.	Unencrypted	N
Asymmetric ECDSA Public Keys	Keys used for the verification of ECDSA digital signatures for IPsec traffic and peer authentication.	Unencrypted	Y
Asymmetric ECDSA Private Keys	Keys used for the calculation of ECDSA digital signatures for IPsec traffic and peer authentication.	Unencrypted	Y

Security Relevant Data Item	**Description**	**How Stored?**	**Persistent?**
Asymmetric RSA Public Keys	Keys used for the verification of RSA digital signatures for signed product updates and code integrity verification.	Unencrypted	Y
AES-CTR DRBG Seed	A secret value maintained internal to the module that provides the seed material for AES-CTR DRBG output.	Unencrypted	N
AES-CTR DRBG Entropy Input	A secret value maintained internal to the module that provides the entropy material for AES-CTR DRBG output.	Unencrypted	N
AES-CTR DRBG V	A secret value maintained internal to the module that provides the entropy material for AES-CTR DRBG output.	Unencrypted	N
AES-CTR DRBG Key	A secret value maintained internal to the module that provides the entropy material for AES-CTR DRBG output.	Unencrypted	N
DH Private and Public values	Private and public values used for Diffie-Hellman key establishment.	Unencrypted	N
ECDH Private and Public values	Private and public values used for EC Diffie-Hellman key establishment.	Unencrypted	N

Table 4 – Windows Keys and CSPs

Each Windows platform includes a key isolation service designed specifically to host secret and private keys in a protected process to mitigate tampering or access to sensitive key materials. As such, they are stored in plaintext in the underlying Windows platform. The Windows platforms perform a key error detection check on each transfer of key (internal, intermediate transfers). The Windows platforms prevent archiving of expired (private) signature keys.

The Table below lists the keys and critical security parameters (CSPs) managed by the underlying Linux platform that are needed to meet the requirements in this ST.

Security Relevant Data Item	**Description**	**How Stored?**	**Persistent?**
Symmetric encryption/decryption keys	Keys used for AES encryption/decryption for IKEv2 and IPsec ESP.	Unencrypted	N
HMAC keys	Keys used for HMAC-SHA-256.	Unencrypted	N
Asymmetric ECDSA Public Keys	Keys used for the verification of ECDSA digital signatures for IPsec traffic and peer authentication.	Encrypted	Y
Asymmetric ECDSA Private Keys	Keys used for the calculation of ECDSA digital signatures for IPsec traffic and peer authentication.	Encrypted	Y
Asymmetric RSA Public Keys	Keys used for the verification of RSA digital signatures for signed product updates and code integrity verification.	Unencrypted	Y
AES-CBC Key	Keys used to encrypt passwords/passphrases using AES 256 CBC.	Unencrypted	N

Security Relevant Data Item	Description	How Stored?	Persistent?
AES-CTR DRBG Seed	A secret value maintained internal to the module that provides the seed material for AES-CTR DRBG output.	Unencrypted	N
AES-CTR DRBG Entropy Input	A secret value maintained internal to the module that provides the entropy material for AES-CTR DRBG output.	Unencrypted	N
AES-CTR DRBG V	A secret value maintained internal to the module that provides the entropy material for AES-CTR DRBG output.	Unencrypted	N
AES-CTR DRBG Key	A secret value maintained internal to the module that provides the entropy material for AES-CTR DRBG output.	Unencrypted	N
DH Private and Public values	Private and public values used for Diffie-Hellman key establishment.	Unencrypted	N
ECDH Private and Public values	Private and public values used for EC Diffie-Hellman key establishment.	Unencrypted	N

Table 5 – Linux Keys and CSPs

For each Linux platform identified in the ST, the persistent secrets and private keys listed as being stored by the platform are protected via the mandated full disk encryption.

6.1.5 FCS_CKM_EXT.4

The secret keys (keys used for symmetric encryption), private keys, and CSPs are identified in **Table 4** and **Table 5**.

All certificates used by Stealth on a Windows system are stored in a certificate store managed by the Windows operating systems. The Windows Certificate Manager is used to import certificates into a certificate store and delete certificate from a certificate store.

The Windows platforms destroy non-persistent cryptographic keys after a cryptographic administrator-defined period of time of inactivity. The Windows platforms overwrite each intermediate storage area for plaintext key/critical cryptographic security parameter (i.e., any storage, such as memory buffers, that is included in the path of such data). This overwriting is performed as follows:

- For non-volatile memories other than EEPROM and Flash, the overwrite is executed three or more times using a different alternating data pattern each time upon the transfer of the key/critical cryptographic security parameter to another location.
- For volatile memory and non-volatile EEPROM and Flash memories, the overwrite is a single direct overwrite consisting of a pseudo random pattern, followed by a read-verify upon the transfer of the key/critical cryptographic security parameter to another location.

On the Linux platforms, cryptographic key destruction is handled by the OpenSSL Cryptographic Module as follows: For non-volatile memory other than EEPROM and Flash, the module doesn't perform persistent storage of CSPs

- For volatile memory and non-volatile EEPROM and Flash memories, the module overwrites memory with pre-defined values and then deallocates the memory.

The Stealth Linux Endpoint performs cryptographic key destruction as follows:

- For non-volatile memory other than EEPROM and Flash, any certificates used by the Stealth Linux Endpoint software remain on the system while Stealth is installed. To securely remove the certificates after the Stealth Linux Endpoint software is removed, execute the following Linux commands:

  - shred -ux /etc/stealth/ca-certs/*
  - shred -ux /etc/stealth/certs/*
  - shred -ux /etc/stealth/ssl-client-certs/*
- For volatile memory and non-volatile EEPROM and Flash memories, the Stealth Linux Endpoint software overwrites memory with zeros and then frees the memory.

  Stealth implements a function that accepts a buffer pointer and buffer length. This function writes a NULL to each byte within the buffer. Buffers are cleared as soon as they are no longer needed.

6.1.6 FCS_COP.1.1(1), FCS_COP.1.1(2), FCS_COP.1.1(3), FCS_COP.1.1(4)

The TOE performs the following cryptographic operations by invoking the appropriate algorithms provided by third party cryptographic modules.

The TOE performs encryption/decryption using AES-CBC as defined in NIST SP 800-38A mode; and AES-GCM as defined in NIST SP 800-38D with 128-bit and 256-bit cryptographic key sizes.

The TOE performs SHA-256, SHA-384, and SHA-512 cryptographic hashing services with message digests of 256, 384, and 512 bits that meet the following: FIPS Pub 180-4. The SHA-256 and SHA-384 are used in the IKE_SA_INIT Integrity algorithm. Digital signatures for all the Stealth executables (binary and scripts) are generated and released with Stealth. The signatures are generated using SHA256 and a RSA 4096-bit key-pair. SHA-512 is used to verify the integrity of the Stealth installation packages.

The TOE performs cryptographic signature services using RSA schemes using cryptographic key sizes of 2048-bit or greater for digital signature verification of the Windows Stealth executables. The TOE can be configured to support the IKEv2/IPsec cryptographic protection profiles. All of the profiles specify the use of ECDSA-signed X.509v3 certificates for IKE peer authentication. The TOE supports the use of NIST curves P-256 and P-384 with ECDSA X.509v3 certificates.

The TOE performs keyed-hash message authentication for the Unisys Stealth Windows Endpoints and the Unisys Stealth Linux Endpoints to implement IPsec using the ESP protocol using the HMAC-SHA-256 and HMAC-SHA-384 algorithms.

6.1.7 FCS_IPSEC_EXT.1

The TOE and the TOE platform implement the IPsec architecture as specified in RFC 4301 to establish a point-to-point VPN tunnel with another Stealth-enabled Endpoint. The TOE supports only transport mode.

The Internet Key Exchange (IKE) permits two Stealth-enabled Endpoints to build an IPsec Security Association (SA). IKEv2 consists of 2 phases: SA and Child SA. The SA creates the first tunnel and creates a subsequent set of keys. The key negotiated in the SA phase enables the Stealth-enabled Endpoints to communicate securely using the Child SA.

Each of the supported Windows platforms provides an implementation of IPsec that conforms to RFC 4301. The platforms implement the IPsec Security Policy Database (SPD) via the Windows Filtering Platform (WFP) which provides the IKEv2 and IPsec implementation. The WFP allows for filtering, monitoring and/or modification of TCP/IP packets, as well as filtering of IPsec traffic. The WFP allows for access to TCP/IP processing at different layers and can be used to filter on incoming or outgoing traffic.

The TOE creates WFP filters and directs the Windows Base Filter Engine to use IPsec for specific Endpoint to Endpoint traffic. The TOE uses the WFP to build conditions and filters that map to filter definitions provided to the TOE by the Authorization Service.

For the Stealth Windows endpoint, the USSL_Protocol service is responsible for establishing the IPSec tunnel. The Windows Filter Platform API function named FwpmFilterAdd is primarily responsible for defining filters. Conditions are a property of filters. The list of Windows Filtering Platform APIs used is documented in Section 9.1. See the APIs imported from the fwpuclnt.dll under the USSL_Protocol service.

For Stealth Windows endpoints, the TOE provides:

- Implementation of SCIP
- Overall configuration (i.e. calls to setup filter rules)
- Tunnel setup.

The TOE platform provides:

- The filtering platform to allow/queue/drop traffic
- Certificate handling: decrypt, read, and validate certificates
- Generation of keys and key handling
- Encryption/Decryption of data and traffic
- Implementation of IPsec and IKEv2 via the Windows Filtering Platform.

The Unisys Stealth Linux Endpoints call unisys-fips-strongswan to provide an IPsec implementation that conforms to RFC 4301. strongSwan is part of the TOE and provides the implementation of IKEv2 and IPsec and the TOE platform provides the authentication and authorization services used in the implementation of IKEv2 and IPsec.

For Stealth Linux endpoints, the TOE provides:

- Implementation of SCIP
- Implementation of IPsec and IKEv2
- Overall configuration (i.e. calls to setup filter rules)
- Tunnel setup.

The TOE platform provides:

- The filtering platform to allow/queue/drop traffic
- Certificate handling: decrypt, read, and validate certificates
- Generation of keys and key handling
- Encryption/Decryption of data and traffic.

Stealth on Linux endpoint does not insert itself in the network stack via kernel mods. Stealth calls the netfilter libraries, including the libiptc library, to interact with the network stack. The netfilter libraries are used to create, modify, and delete iptables rules to allow, drop, and queue network packets. The TOE creates 'iptables' rules that map to filter definitions provided to the COI by the Authorization Service. Queueing of packets allows Stealth to intercept packets. The packets are processed by the Stealth daemon to see if a tunnel should be created and if the packet should be allowed out on the network interface.

SCIP is used to perform initial negotiation of the Stealth tunnel between the endpoints. The result of a successful negotiation is creation of a tunnel that IPsec traffic can be sent through. Once a Stealth tunnel has been set up, traffic between the two endpoints goes through the Stealth tunnel. Optionally filters can be put in place to allow some traffic to be sent as Clear Text traffic.

Initial packets are allowed, queued, or dropped based on the defined filters and Communities of Interest (COIs). This results in specified Clear Text traffic being allowed, other traffic being queued to be sent through a Stealth tunnel (IPsec traffic), and the rest of the traffic being dropped.

To configure filters on the Unisys Stealth Windows Endpoints and the Unisys Stealth Linux Endpoints, the Stealth administrator specifies the following:

- Filter Qualifier—a filter qualifier specifies a list of IP address exceptions and the port or protocol ranges allowed for a qualified filter.
- Filter List—a filter list is a list of qualified filters (IP addresses or ranges, with optional qualifiers), a list of filter lists, or a mix of qualified filters and filter lists.
- Filter Set—a filter set is a group of filter lists, with each list configured in a certain order and each set defined as either Allow or Deny. The order of filter lists in the filter set determines the order in which the filter lists are processed.

Filter sets work differently depending on whether they are applied to COIs or roles, as follows:

- A filter set applied to a COI is designated as a *Stealth Filter*. Stealth Filters control Stealth-enabled network traffic, allowing or denying information passed between Stealth Endpoints that share COIs.

- A filter set applied to a role is designated as a *Clear Text Filter*. Clear Text Filters control clear text network traffic, allowing or denying information passed between Stealth Endpoints and non-Stealth-enabled (clear text) components.

Directional COIs are used to manage inbound and outbound Stealth tunnel establishment. When Stealth workgroup COIs are assigned to a role, the following directional attributes can be selected to control how endpoints in that role establish tunnels:

- Initiate – Restricts the Stealth workgroup COI to initiate outbound Stealth tunnels.
  When a workgroup COI is added to a workgroup role as an Initiate COI, the endpoints in that role can only use that COI to initiate outbound tunnel requests. (Once a tunnel is established, communication can flow in either direction.)
- Accept – Restricts the Stealth workgroup COI to accept inbound Stealth tunnel requests.
  When a workgroup COI is added to a workgroup role as an Accept COI, the endpoints in that role can only use that COI to accept inbound tunnel requests. (Once a tunnel is established, communication can flow in either direction.)
- Default – Enables the Stealth workgroup COI to both initiate outbound tunnels and accept inbound tunnels.

The Directional COI attributes only affect Stealth tunnel establishment. After a Stealth tunnel is established, communication can pass in either direction, as long as any assigned filter qualifiers allow that communication.

The Stealth administrator can apply either Allow filter sets or Deny filter sets to a COI or a role. Filter sets applied to roles are checked before filter sets applied to COIs.

- Allow Filter—an Allow filter set applied to a COI (Stealth Filter) operates as a set of IPsec PROTECT rules. The Stealth administrator specifies rules for filtering the network traffic to be protected. Messages that meet the filter criteria are transmitted using an IPsec tunnel. If the filter criteria are not met, then the data is discarded. An Allow filter set applied to a role (Clear Text Filter) operates as a set of IPsec BYPASS rules. The Stealth administrator specifies rules for filtering the network traffic to be allowed through without modification. Messages that meet the filter criteria are transmitted to or from the clear text network. If the filter criteria are not met, then the data is transmitted using an IPsec tunnel. No frames are discarded.

- Deny Filter—a Deny filter set applied to a COI (Stealth Filter) operates as a set of IPsec DISCARD rules. The Stealth administrator specifies rules to filter network traffic to be denied. Messages that meet the filter criteria are discarded. All other traffic is sent using an IPsec tunnel. A Deny filter set applied to a role (Clear Text Filter) operates as a set of IPsec PROTECT rules. The Stealth administrator specifies rules for filtering the network traffic to be denied passing through the network without modification. Messages that meet the filter criteria (that match the deny filter) are transmitted using an IPsec tunnel. All others are transmitted to or from the clear text network.

An Allow filter set has a default behavior of deny; that is, any condition that does not match the allow filter is denied by default. In the same way, a Deny filter set has a default behavior of allow, so that any condition that does not match the deny filter is allowed.

The TOE can be configured to support each of the IKEv2/IPsec cryptographic protection profiles specified in **Table 6** below. Note that all profiles specify the use of ECDSA-signed X.509v3 certificates for IKE peer authentication. The TOE supports the use of NIST curves P-256 and P-384 with ECDSA X.509v3 certificates.

Endpoints must share at least one crypto profile and at least one COI in order to communicate. To enable Unisys Stealth Windows Endpoints to communicate with other Unisys Stealth Windows Endpoints, or to enable Unisys Stealth Linux Endpoints to communicate with other Unisys Stealth Linux Endpoints, any crypto profile is permitted. However, for Unisys Windows and Unisys Linux Stealth endpoints to communicate with one another, a crypto profile with an IKE_SA_INIT Integrity algorithm that matches the IKE Negotiation certificate private key must be chosen. That is, if a certificate with a 256-bit private key is chosen, then the use of crypto profile 0x100 or 0x200, which uses an IKE_SA_INIT Integrity algorithm of SHA-256, as shown in the “Int” column in **Table 6** is chosen. If a certificate with a 384-bit key is chosen, the crypto profile 0x400, 0x800, or 0x1000, which use an IKE_SA_INIT Integrity algorithm of SHA-384 must be chosen. If the wrong cryptographic profile for the certificate key size is used, then Stealth tunnels cannot be established between Unisys Stealth Windows and Unisys Stealth Linux endpoints.

SCIP processing attempts to negotiate the crypto profiles in the order in which they appear in this profile priority list (from left to right). It uses the first crypto profile that is shared between two endpoints.

The Unisys Stealth Windows Endpoints and the Unisys Stealth Linux Endpoints implement IPsec using the ESP protocol as defined in RFC 4303 using the AES-GCM-128 and AES-GCM-256 cryptographic algorithms as specified in RFC 4106 along with the HMAC-SHA-256 and HMAC-SHA-384 algorithms. The Unisys Stealth Endpoints encrypt the IKE payloads using AES-CBC-128 and AES-CBC-256 as specified in RFC 6379.

The Unisys Stealth Windows Endpoints and the Unisys Stealth Linux Endpoints use the IKEv2 protocol as specified in RFCs 7296 with support for NAT traversal as specified in section 2.23, and 4307 to authenticate and establish session keys with the Stealth Endpoints.

A certificate that expires after a Stealth tunnel is opened is detected by the endpoint operating system IKE module when re-authentication occurs. (Re-authentication occurs when the “phase1” attribute for the lifetime element in the protectionprofile.xml expires, which is set to 24 hours by default). This causes the Stealth tunnel to close.

The Stealth Endpoints implement IKEv2 with support for the following DH groups 14 (2048-bit MODP), 19 (256-bit Random ECP), and 20 (384-bit Random ECP). The IKE protocol is determined by the supported protection profiles, as indicated in **Table 6**, each supported protection profile ensures the strengths, in terms of the number of bits in the symmetric key, of the algorithms allowed for the IKE exchange are always equal to the strengths of the encryption algorithms allowed for ESP.

The TOE platform is responsible for generating nonces and the secret value x used in the IKE Diffie-Hellman key exchange (i.e., “x” in g^x mod p). A specific nonce value will be repeated during the life a specific IPsec SA is less than 1 in 2^{224} bits for DH Group 14, 2^{256} bits for DH Group 19, and 2^{384} bits for DH Group 20. When a random number is needed for either a nonce or for key agreement, the platform uses its NIST-approved random bit generator. When requested, the platform’s random bit generator can generate 256 or 512 bits for the caller. The TOE platform can generate nonces used in IKE exchanges in a manner such that the probability that a specific nonce value will be repeated during the life of a specific IPsec SA is less than 1 in 2^{112} for DH Group 14, 2^{128} for DH Group 19, and 2^{192}. A 256-bit random value provides sufficient strength for nonces for all of the Diffie-Hellman groups supported by the TOE, while a 512-bit random value for x in g^x mod p is at least twice the “bits of security” value associated with each supported Diffie-Hellman group.

The IKEv2 protocols implement Stealth-enabled Endpoint authentication using the ECDSA algorithm using x509v3 certificates. The TOE platforms processes X.509 certificates for IPsec, in accordance with RFC 5280 (Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile), which implies that if the platform deems that a certificate is invalid, such as for a DN mismatch, a revoked certificate, or an expired certificate, it will not establish an IPsec association.

The two Stealth Endpoints negotiate a complete profile. Each supported protection profile is defined so that the Phase 1 and Phase 2 SAs have the same strength – 128 bits in Profiles 0x100 and 0x200, and 256 bits in the 0x400, 0x800, and 0x1000 Profiles.

The Unisys Stealth Windows Endpoint and Unisys Stealth Linux Endpoint software both use FQDN as reference identifier type. The reference identifier type and expected reference identifier value are included in the protection profile configuration file. The certificate used by a Stealth endpoint must include a Subject Alternative Name field with a DNS value matching the configured reference identifier value.

Bit Mask/ Profile Name	IKE							IPsec			
	IKE_AUTH			IKE_SA_INIT							
	Auth.	Cipher	Trans	Auth.	Cipher	Int.	DH	Mode	PFS	Encr.	Auth
0x100 Secret IKEv2-DH14-X509v3	AES 128	AES 128	ESP	X509 v3	AES 128 CBC	SHA 256	DH 14	Transport	DH 14	AES 128 GCM	GCM
0x200 Secret IKEv2-DH19-X509v3	AES 128	AES 128	ESP	X509 v3	AES 128 CBC	SHA 256	DH 19	Transport	DH 19	AES 128 GCM	GCM
0x400 TopSecret IKEv2-DH14-X509v3	AES 256	AES 256	ESP	X509 v3	AES 256 CBC	SHA 384	DH 14	Transport	DH 14	AES 256 GCM	GCM
0x800 TopSecret-IKEv2-DH19-X509v3	AES 256	AES 256	ESP	X509 v3	AES 256 CBC	SHA 384	DH 19	Transport	DH 19	AES 256 GCM	GCM
0x1000 TopSecret IKEv2-DH20-X509v3	AES 256	AES 256	ESP	X509 v3	AES 256 CBC	SHA 384	DH 20	Transport	DH 20	AES 256 GCM	GCM

Table 6 – Supported Protection Profiles

6.1.8 FCS_RBG_EXT.1

The TOE platform is responsible for generating nonces and the secret value *x* used in the IKE Diffie-Hellman key exchange (i.e., "*x*" in g^x mod *p*). When a random number is needed for either a nonce or for key agreement, the platform uses its NIST-approved random bit generator. When requested, the platform's random bit generator can generate 256 or 512 bits for the caller. The TOE platform can generate nonces used in IKE exchanges in a manner such that the probability that a specific nonce value will be repeated during the life of a specific IPsec SA is less than 1 in 2^{256}. A 256-bit random value provides sufficient strength for nonces for all of the Diffie-Hellman groups supported by the TOE, while a 512-bit random value for *x* in g^x mod *p* is at least twice the "bits of security" value associated with each supported Diffie-Hellman group.

Each Windows platform implements a deterministic random bit generation (DRBG) function in accordance with NIST Special Publication 800-90. The interface for the Windows random number generator is BCryptGenRandom.

On each FIPS-validated Red Hat 7.4 or 7.5 Linux operating system the TOE relies on a call to the RAND_bytes function in the FIPS-validated Red Hat OpenSSL. The Unisys Stealth Linux Endpoints initialize the PRNG by calling RAND_load_file("/dev/urandom", 32).

6.1.9 FCS_STO_EXT.1

The Unisys Stealth Windows Endpoint invokes the functionality provided by the platform to securely store the domain credentials for USSL_Logon service. The Unisys Stealth Windows Endpoint provides three possible authentication methods:

- Integrated Windows Authentication

  On a Windows workstation Windows Stealth Endpoint uses the identity of the logged on domain user.

  On a Windows server, a domain account can be associated with the Stealth USSL_Logon service. The Windows Credential Manager maintains the credential information.

- Certificate Based Authentication

  A certificate is stored in the Windows Certificate Store under Local Computer Personal. The certificate Subject Alternative Name field must contain a User Principal Name that maps to a domain account or a domain computer account. The certificate trust chain is stored under Trusted Root Certification Authorities and Intermediate Certification Authorities.

- LDAP Authentication

  The user is prompted for domain credentials.

The Unisys Stealth Linux Endpoint invokes the functionality to retrieve the private key associated with the IKE negotiation and passes it to strongSwan. Stealth does not save the private key to non-volatile storage.

6.2 User Data Protection

6.2.1 FDP_DAR_EXT.1

The Unisys Stealth Windows Endpoint does not store any sensitive data and cannot write sensitive data to non-volatile memory. When the Unisys Stealth Windows Endpoint software is running in Client Always On mode or Client On Demand mode, the software impersonates the logged on Domain User. When the Stealth Windows Endpoint software is running in Server Always On mode the Windows MMC Service Snap-in is used to specify a Domain account that the USSL_Logon service will logon as.

The Unisys Stealth Linux Endpoint invokes the functionality provided by the platform to securely store the domain user passphrase and the private key passphrase. The Unisys Stealth Linux Endpoint provides two possible authentication methods:

- Certificate Based Authentication

  A PKCS #12 file is stored in /etc/stealth/ssl-client-certs. The certificate Subject Alternative Name field must contain a User Principal Name that maps to a domain account or a domain computer account. The ca-cert configuration parameter defines a fully qualified path to the certificate trust chain. This configuration parameter is located in the Stealth system.ini configuration file. The private key passphrase is stored in the Stealth Secrets database.

- LDAP Authentication

  The username configuration parameter defines a domain account. This configuration parameter is located in the Stealth system.ini configuration file. The domain account passphrase is stored in the Linux file system. The Unisys Stealth Common Criteria Evaluation Guidance Document instructs the administrator to configure the Linux platform for full disk encryption using AES-256 for the secure storing of the credentials

For the Unisys Stealth Linux Endpoint, the passwords and private key passphrases the application requires are not actually needed to meet the requirements and therefore are considered application data within the scope of FDP_DAR_EXT.1. The Operational Guidance specifies the need for an administrator to activate platform encryption.

6.2.2 FDP_DEC_EXT.1

The Unisys Stealth Windows Endpoint application and the Unisys Stealth Linux Endpoint application restrict their access to platform hardware resources to network connections. The Unisys Stealth application restricts its access to sensitive information repositories to the system logs. The Unisys Stealth Windows Endpoint application writes log messages to the Windows Application Event Log. The Unisys Stealth Linux Endpoint application writes log messages to the system log (/var/log/messages).

6.2.3 FDP_NET_EXT.1

The Unisys Stealth Windows Endpoint application and the Unisys Stealth Linux Endpoint application restrict network communication to the Stealth Authorization server for authentication, session management, and configuration reporting. The user initiates communication for Stealth-tunneled network traffic to other Unisys Stealth peers.

6.2.4 FDP_RIP.2

Both the Unisys Stealth Windows Endpoint application and the Unisys Stealth Linux Endpoint application as well as the underlying Windows and Linux platforms ensure that no residual information exists in network packets. After processing either an incoming or outgoing network packet, the Unisys Stealth Windows Endpoint application and the Unisys Stealth Linux Endpoint application overwrites with zeros the contents of each memory buffer it used in packet processing, prior to releasing the buffer to the memory pool of the underlying platform. The Windows and Linux platforms ensure that previous information contents of resources used for new objects are not discernible in the new object via zeroing or overwriting of memory and tracking read/write pointers for disk storage. Every process is allocated new memory and an execution context. Memory is zeroed or overwritten before allocation.

6.3 Identification and Authentication

6.3.1 FIA_X509_EXT.1, FIA_X509_EXT.2

The TOE ensures the use of X.509v3 certificates, as defined in RFC 5280, for authentication of IPsec exchanges and to support integrity verification. For certificates used to support IPsec authentication, the certificate to use is identified by its Common Name, which is specified in the protectionprofile.xml file used when configuring the Stealth endpoint installation package.

The Unisys Stealth Windows Endpoint and Unisys Stealth Linux Endpoint platforms process X.509v3 certificates for IPsec in accordance with RFC 5280 and use Certificate Revocation Lists (CRLs), as specified in RFC 5759, to determine revocation status. The Unisys Stealth Windows Endpoint and Unisys Stealth Linux Endpoint platforms validate all applicable usage constraints, including:

- Ensuring the basicConstraints extension is present and the cA flag is set to TRUE for all CA certificates
- Validating the extendedKeyUsage field to ensure certificates used for executable code integrity verification and trusted updates have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3).

The TOE platforms validate the full path of X.509v3 certificates by validating they can construct a certificate path from the certificate through any intermediary CAs to the trusted root CA certificate. If the TOE platforms can successfully build the certificate path, then they will next check the validity of the certificates in the path. Assuming the certificates are valid, the TOE platforms finally check the revocation status of all certificates. If the TOE platforms deem that a certificate is invalid, such as for a Fully Qualified Domain Name (FQDN) mismatch, a revoked certificate, or an expired certificate, they will not establish an IPsec association. The Stealth administrator can configure a value in the protectionprofile.xml file that controls CRL checking. If strong CRL checking is enabled and the TOE platforms cannot establish a connection to determine the validity of a certificate, the platforms will not accept the certificate. If weak CRL checking is enabled and the TOE platforms cannot establish a connection to determine the validity of a certificate, they assume the certificate is valid. The weak CRL checking only rejects a certificate if the CRL is successfully accessed but the certificate is found to be revoked.

The TOE uses X.509v3 certificates to support integrity verification of the endpoint installation package. During the endpoint package generation process, the Enterprise Manager reads the protectionprofile.xml file, validates it, and inserts the contents into the crypto.xml file. Enterprise Manager signs the resulting crypto.xml file using a signing certificate and includes it in the endpoint package. The resulting endpoint package is then installed on the endpoints. The endpoints on which the package is installed read the crypto.xml file and validate the signature of the signing certificate using the associated trusted root certificate, which is also installed on these endpoints.

The Windows platform uses X.509v3 certificates to support integrity verification of TSF kernel-mode executables when they are loaded for execution, while the TOE uses X.509v3 certificates to support integrity verification of TSF user-mode services. This is described fully in Section 6.6.4 below.

The Linux platform uses X.509v3 certificates to support integrity verification of the FSF kernel-mode executables when they are loaded for execution.

The Windows X.509v3 certificate used for IKE peer authentication (termed the IKE Negotiation certificate in the TOE guidance documentation) is imported to the Local Computer Personal certificate store on the TOE platform. The physical location of the certificate store is the registry hive for the computer. Certificates can only be loaded into the certificate store by an authorized administrator who has write (i.e., modify) and delete access rights to the registry keys that serve as the certificate store. Certificates can be loaded using GUI administrator tools, command line tools, or through local and group policy.

The Linux X.509v3 IKE Negotiation certificate used for IKE peer authentication, including the private key as a .pfx file, is installed to the /etc/stealth/ssl-client-certs directory on all Linux endpoints. The trusted root certificate and any intermediate certificates are installed to the appropriate file or directory.

6.4 Security Management

6.4.1 FMT_CFG_EXT.1

The Unisys Stealth Windows Endpoint application and the Unisys Stealth Linux Endpoint application installer do not create default credentials during installation. The Linux domain user passphrase and private key passphrase are created by the administrator.

6.4.2 FMT_MEC_EXT.1

The TOE invokes the mechanisms recommended by the platform vendor for storing and setting configuration options. The configuration options consist of:

- configured cipher suites,
- the setting for CRL checking,
- certificate details.

Configuration information is stored in the Windows registry when installing a Unisys Stealth Windows Endpoint package. No additional configuration changes are allowed after installation of a Unisys Stealth Windows Endpoint package.

The Unisys Stealth Linux Endpoint stores configuration information in the Linux file system at /etc using restricted file permissions. No additional configuration changes are allowed after installation of a Unisys Stealth Linux Endpoint package.

6.4.3 FMT_SMF.1, FMT_SMF.1/VPN

The TSF is capable of performing the following management functions:

- Specify IPsec VPN Clients to use for connections
- Specify client credentials to be used for connections
- Configure the reference identifier of the peer
- Specify IKEv2 SA lifetimes

- Configure packet filter rules
- Configure CRL checking
- Configure algorithm suites that can be proposed and accepted during IPsec exchanges.

The Unisys Stealth Windows Endpoint application and the Unisys Stealth Linux Endpoint application allow the user to select from the VPN clients configured by the administrator and provided to the Endpoint as part of the Endpoint package to specify other IPsec VPN clients to use for connections.

The Windows X.509v3 certificate used for IKE peer authentication (termed the IKE Negotiation certificate in the TOE guidance documentation) is imported to the Local Computer Personal certificate store on the TOE platform.

The Linux X.509v3 IKE Negotiation certificate used for IKE peer authentication, including the private key as a .pfx file, is installed to the /etc/stealth/ssl-client-certs directory on all Linux endpoints. The TOE uses the IKE Negotiation certificate and the configured reference identifier of the peer from the Enterprise Manager for peer connections.

The TOE implements the Unisys proprietary Secure Community of Interest Protocol, which is combined with standard Internet Protocol Security to encrypt and authenticate each IP packet of a communication session. This combination of protocols is known as SCIP/IPsec.

Stealth processing uses the SCIP/IPsec protocol and involves authorization and session processing phases. These phases occur automatically and seamlessly in a Stealth environment managed by Enterprise Manager, as follows:

- Authorization Phase
  Once authenticated by the local identity management system, the user that is logged onto the endpoint is assigned COI access and policies during the authorization phase of processing. The Unisys Stealth Windows Endpoints and Unisys Stealth Linux Endpoints issue a tuples request to a configured Stealth Authorization server. The tuples contain a set of COIs, their associated filter sets, and any clear text filters for which the user is authorized. This request is an HTTP request through an IPsec tunnel.

  Enterprise Manager generates an install or runtime XML file for the endpoint software, which contains the information necessary to initiate and maintain a successful session. Enterprise Manager uses the signing certificate to protect the integrity of the settings.xml and crypto.xml files, which are included in the endpoint software package. The endpoint uses the corresponding root and intermediate certificates to validate these XML files before enabling Stealth.

- Session Phase

  Once authorized, the endpoint can open data transfer tunnels to other endpoints. The activity of opening tunnels by an authorized endpoint constitutes a session. All traffic between the endpoints with a data transfer tunnel operating is protected (subject to filtering processing).

- Session Initiation

  The process of starting a SCIP/IPsec tunnel begins when the Stealth endpoint software receives a network packet that should be sent between an initiating endpoint and a destination endpoint (to which it does not already have a tunnel). The initiating endpoint sends Session 0 (Sess0), and the data that initiated this process is queued for the duration of the tunnel establishment. However, if the tunnel is not established after three retries (at two-second intervals), the data is discarded, the process of opening the tunnel fails, and a tunnel open failure message is logged.

  When an endpoint opens a tunnel to another endpoint, it sends the authorization token that it received during the authorization phase. The authorization token contains a signature to validate and information about supported COIs. This information allows the endpoint to determine whether it is allowed to communicate with the remote endpoint. When the target endpoint receives the authorization token, it attempts to validate the signatures against its list of available COI certificates received from the Authorization Service during the authorization phase. If no match is found, then the endpoint does not honor the request. If a match is found, then the target endpoint responds positively to the initiating endpoint. Both sides log the successful initiation of a tunnel, including the identification of the users.

6.5 Privacy

6.5.1 FPR_ANO_EXT.1

The Unisys Stealth Windows Endpoint and the Unisys Stealth Linux Endpoint do not collect or transmit PII over a network.

6.6 Protection of the TSF

6.6.1 FPT_AEX_EXT.1

Windows

The Unisys Stealth Windows Endpoint application uses the /DYNAMICBASE linker operation to enable ASLR when the application is compiled. The TOE does not allocate any memory region with both write and execute permissions. The Unisys Stealth Windows Endpoint can run successfully with Windows Defender Exploit Guard configured and enabled.

The Unisys Stealth Windows Endpoint does not write user-modifiable files to directories that contain executable files unless explicitly directed by the user to do so. The Unisys Stealth Windows Endpoint application is compiled with the Enable Security Check (/GS) complier option for stack-based buffer overflow protection.

Linux

The Unisys Stealth Linux Endpoint application enables address space layout randomization (ASLR) using the –fpie compiler option and the –pie linker option. The TOE does not allocate any memory region with both write and execute permissions.

The Unisys Stealth Linux Endpoint can run on a Red Hat Enterprise Linux (RHEL) 7.4 or a Red Hat Enterprise Linux (RHEL) 7.5 system with SELinux enabled. The Linux Unisys Stealth application is compiled using GCC with the –fstack-protector-strong flag compiler option for stacked-based buffer overflow protection.

6.6.2 FPT_API_EXT.1

The Unisys Stealth Windows Endpoint application and the Unisys Stealth Linux Endpoint application use only documented platform APIs. See Appendix 9 for the listing of platform APIs.

6.6.3 FPT_IDV_EXT.1

The Unisys Stealth Windows Endpoint software version information is displayed in the lower right-hand corner of the Stealth Dashboard window. All components of a Unisys Stealth Windows Endpoint release have the same build number.

The Unisys Stealth Linux Endpoint software version information is displayed by the execution of the command stconfig –V. The Unisys Stealth Linux Endpoint includes 3rd party libraries that are versioned by the supporting third party organization. These 3rd party libraries are included in the TOE where the number behind the hyphen "-" in the third party library is incremented by Unisys when Unisys makes a change to the component as per Linux convention.

The Unisys Stealth versioning is broken down as follows:

Version: Major Release.Minor Release.Cycle.Patch

Example: 4.0.026.0

- Major Release: indicates a new release where significant new feature content is added to Stealth and is used for Marketing Launch.

- Minor Release: indicates a new release where minor new feature content is added to Stealth and there will not be a Marketing Launch.
- Cycle: indicates a software build number. All components of a Stealth release have the same build number and a complete test cycle is performed.
- Patch: the Patch number indicates that a Stealth component has changed in the event Unisys needs to provide a fix quickly without going through a complete build test cycle. This allows Unisys to deliver fixes more quickly. Targeted testing of the fix plus limited regression testing is performed. Hotfixes start at .01 and go upward. A value of .0 means that the component does not have any hotfixes.

The documentation version is referenced by the Major.Minor format (e.g. 4.0).

6.6.4 FPT_LIB_EXT.1

The Unisys Stealth Windows Endpoint application and the Unisys Stealth Linux Endpoint application use only the third-party libraries listed in the security functional requirements.

6.6.5 FPT_TST_EXT.1

The underlying Windows platform performs a set of self-tests to verify that Windows is operating correctly.

The relevant kernel-mode startup self-tests are:

- AES encrypt/decrypt Known Answer Test for CBC and GCM modes
- RSA Known Answer Test
- ECDSA sign/verify tests on supported NIST curves
- ECDH secret agreement Known Answer Test on supported NIST curves
- HMAC-SHA-256 Known Answer Tests
- SP800-56A concatenation KDF Known Answer Tests (same as Diffie-Hellman KAT)
- SP800-90 AES-256 counter mode DRBG Known Answer Tests (instantiate, generate and reseed).

The Windows kernel-mode cryptographic module (the Kernel Mode Cryptographic Primitives Library) also performs pair-wise consistency checks upon each invocation of RSA, ECDH, and ECDSA key-pair generation and import. SP 800-56A conditional self-tests are also performed. A continuous RNG test (CRNGT) is used for the random number generators of this cryptographic module. A pair-wise consistency test is done for Diffie-Hellman.

The Kernel Mode Cryptographic Primitives Library is loaded into the kernel's memory early during the boot process. If there is a failure in any startup self-test, the Kernel Mode Cryptographic Primitives Library DriverEntry function will fail to return the STATUS_SUCCESS status to its caller. The only way to recover from the failure of a startup self-test is to attempt to invoke DriverEntry again, which will rerun the self-tests, and will only succeed if the self-tests pass.

By thoroughly exercising the cryptography that is the basis for IPsec, Windows will avoid situations where data is transmitted through a non-trusted channel in cases where the authorized administrator has deemed that a trusted channel is necessary.

The underlying Windows platform verifies the integrity of Windows program code using its Code Integrity capability. Kernel-mode code signing (KMCS) prevents kernel-mode device drivers from loading unless they are digitally signed by trusted developers. KMCS, using public-key cryptography technologies, requires that kernel-mode code include a digital signature generated by one of the trusted certificate authorities. The TOE includes two kernel-mode device drivers (`mlstpgw.sys` and `stealthii.sys`) that must be digitally signed in this fashion. When either of these kernel device drivers tries to load, Windows decrypts the hash included with the driver using the public key stored in the certificate, then verifies that the hash matches the one computed with the code. The authenticity of the certificate is checked in the same way, using the certificate authority's public key, which is trusted by Windows. The root public key of the certificate chain used to verify the signature must match one of Microsoft's root public keys. These Microsoft root public keys are hardcoded in the Windows Boot Manager.

When each of the TOE's user-mode services is started, the integrity of the service is verified by invoking services of the Windows platform and using the digital signature of the executable. If the digital signature of a service cannot be verified, the following occurs:

- The service enters the paused state
- An error is written to the Windows System Event log.

In addition, the TOE cannot be enabled, and the Stealth Dashboard Stealth Shield icon appears yellow in the Windows taskbar.

In contrast, if the integrity verification is successful, the services start. The Stealth Dashboard Stealth Shield icon appears blue if the TOE is installed as Always On, or red if the TOE is installed as On Demand (when the TOE is enabled, the icon then turns blue).

The underlying Linux platform performs a set of self-tests to verify that Linux is operating correctly.

The RHEL 7.4 and RHEL 7.5 Linux kernel performs the following tests at power up:

- HMAC SHA-512 integrity test
- AES encrypt/decrypt Known Answer Test (tested separately)
- RSA signature verification performed as part of the integrity test (considered a KAT).
- DRBG (CTR) Known Answer Test
- HMAC SHA-256, HMAC SHA-384 Known Answer Test
- SHA-256, SHA-384, SHA-512 Known Answer Test

The following self-tests are performed by OpenSSL when the Stealth service is started.

- AES encryption/decryption Known Answer Test (tested separately)
- DSA pairwise consistency test, sign and verify
- RSA signature generation and verification Known Answer Test (tested separately).
- ECDSA pairwise consistency test, sign and verify
- Diffie-Hellman primitive "Z" Computation Known Answer Test
- EC Diffie-Hellman primitive "Z" Computation Known Answer Test
- SP 800-90A CTR_DRBG Known Answer Test
- HMAC SHA-256, HMAC SHA-384, HMAC SHA-512 Known Answer Test
- SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 Known Answer Test
- CMAC Known Answer Test
- HMAC-SHA-256 integrity test

The Linux operating system does not have capabilities similar to the Windows operating system for checking the integrity of an executable prior to loading it. Instead, the Unisys Stealth Linux Endpoint fulfills this requirement with the following:

Digital signatures for all the Stealth executables (binary and scripts) are generated and released with Stealth. The signatures are generated using SHA256 and a RSA 4096-bit key-pair. The public key is also released with Stealth.

The `stealthd` service is controlled by the platform provided `systemd`. `systemd` is configured to verify the signature for `stealthd` against the `stealthd` file prior to launching `stealthd`. The `systemd` configuration for the `stealthd` and `stconfig` services are files released as part of the Stealth Linux Endpoint software. The files to configure `systemd` are installed during the Stealth Endpoint installation. No user interaction is required to verify the verification. If the verification fails, `stealthd` is not launched, an error message is written to the system log, and the service fails.

The `stealthd` service starts the `stconfig` child service which is also controlled by `systemd`. `systemd` is configured to verify the signature for `stconfig` prior to launching `stconfig`. If the verification fails, stconfig is not launched, an error message is written to the system log, and the `stconfig` service fails. However, the `stealthd` service continues to run.

The `stealthd` process also launches other executables. For these executables, `stealthd` verifies the signature against the executable prior to launching the executable. If the verification fails, the executable is not launched, an error is written to the system log, and the operation fails. The `stealthd` service, however, continues to run. Once the integrity checks of the `stealthd`, `stconfig` and the other executables launched by `stealthd` have been verified, the services will continue to load and execute.

Standalone utilities (`stconfig`, `stuser`, and `collectdiags.sh`) are launched from the command line. These utilities are verified and executed using the `stverify.sh` utility. To execute `stconfig`, `stuser`, or `collectdiags.sh`, pass the name of the utility and any arguments to the `stverify.sh` utility on the command line. For example:

```
sudo stverify.sh stconfig -S
```

The first thing that `stverify.sh` does is verify itself against its own signature. If verification fails, `stverify.sh` displays a message and terminates. If verification is successful, `stverify.sh` then verifies the utility against the utility's signature. Again, if verification fails, a message is displayed and `stverify.sh` terminates; otherwise, the utility is executed.

The signatures for the utilities are generated in the manner described above. The `stverify.sh` script verifies the utility using the openssl CLI running in FIPS mode. The OPENSSL_FIPS environment variable is set.

The `stverify.sh` utility uses the following command for verification:

```
openssl dgst -sha256 -verify /opt/unisys/stealth/signatures/public.pem -
signature /opt/Unisys/stealth/signatures/<executable-name>.sig <full-path-including-
executable-file-name>
```

6.6.6 FPT_TUD_EXT.1

An administrator accesses the Unisys Product Support website (https://www.support.unisys.com) to download a new version of the Stealth .iso file to the Management Server and standalone Authorization Servers.

The Management Server software is upgraded (including the base endpoint packages and Enterprise Manager) using the new version of the Unisys Stealth Release .iso file.

The protectionprofile.xml file is included as part of the Unisys Stealth release 4.0. This file contains the settings that must be configured to enable Stealth to conform to the Protection Profile for Application Software as well as the PP-Module for VPN Clients.

Once the protectionprofile.xml file is modified, it is used in the process of creating endpoint packages. During the endpoint package generation process, Enterprise Manager reads the protectionprofile.xml file, validates it, and inserts the contents into the crypto.xml file (which is also included as part of the Stealth 4.0). Enterprise Manager signs the resulting crypto.xml file using a signing certificate.

A signing certificate is used to protect the integrity of the settings.xml and crypto.xml files, which are included in the endpoint software package. The endpoint package must be signed by the signing certificate since the endpoint package contains the IP addresses and certificates of the Authorization Services. If that information is corrupted, security of the COI information could be compromised. The signing certificate private key resides on the Management Server.

The signing certificate must be authorized by a trusted root certificate (and optionally by intermediate certificates) to create the trust chain. The XML files in the endpoint software package are signed. The signature is verified by the signing certificate, and in turn, the signing certificate is verified by the trust chain.

The signing certificate and associated trusted root certificate are automatically generated by the Stealth software when the Enterprise Manager software is initialized. In order to use a CA to verify the certificate, the automatically generated signing certificate should be replaced by a trusted root certificate with a CA-generated signing certificate and trust chain.

The signing certificate is automatically included in the endpoint package, which means that the signing certificate does not have to be imported on any endpoints.

After an endpoint package is installed, and changes to the protectionprofile.xml file are desired, such as adding base profiles to the profile priority list or reordering the profile priority list, the endpoint package is recreated and installed on all affected endpoints for them to use the changes.

Windows

The Stealth Management Server provides a mechanism for creating new endpoint packages and ability to distribute the Unisys Stealth Windows Endpoint package to targeted endpoints. When an endpoint receives notification that an endpoint package is available, the package is downloaded using IPsec from the Stealth-enabled Management Server and installed on the Stealth endpoint.

The endpoint base packages are installed with the Enterprise Manager software. The TOE software is updated by generating a new endpoint installation package on the Enterprise Manager and installing the new installation packages on the Windows endpoint. A signing certificate is used to protect the integrity of the endpoint software package. The XML files in the endpoint software package are signed, and that signature is verified by the signing certificate. Prior to installation of the endpoint package, the signing certificate's trust chain is installed on the endpoint, including the trusted root certificate (stored in the Windows Local Computer/Trusted Root Certification Authorities store) and any intermediate certificates (stored in the Windows Local Computer/Intermediate Certification Authorities store). The signing certificate itself is automatically included in the endpoint package.

The endpoint user is able to use the capabilities of the underlying Windows platform to verify the integrity of the endpoint <Endpoint package name>.exe installation package by verifying a SHA-512 hash calculated over the entire installation package and by verifying the validity of the digital signatures on the executables within the installation package. The steps to be performed by administrators and users are described in *Unisys Stealth Solution Common Criteria Evaluation Guidance Document*.

During the installation process, Windows verifies the installation package using the digital signature. If the digital signature cannot be verified, you see an error message indicating that the file does not contain digital signature information. The underlying Windows platform reports to the user if the digital signature is OK (for successful signature verification) or invalid (for unsuccessful signature verification).

The Stealth Dashboard or the 'stealthstatus' utility can be used to obtain status information on the current Stealth configuration, including the software version number. The Programs and Features option from the Windows Control Panel can also be used to obtain status information on the current Stealth configuration, including the software version number.

Linux

The endpoint base packages are installed with the Enterprise Manager software. The Stealth Management Server provides a mechanism for creating new endpoint packages and ability to distribute the Unisys Stealth Linux Endpoint package to targeted endpoints. During the endpoint package generation process, Enterprise Manager reads the protectionprofile.xml file, validates it, and inserts the contents into the crypto.xml file (which is also included as part of the Stealth 4.0 and later .iso files). Enterprise Manager signs the resulting crypto.xml file using a signing certificate and includes it in the endpoint package.

When an endpoint receives notification that an endpoint package is available, the package is downloaded using IPsec from the Stealth-enabled Management Server and installed on the Stealth endpoint.

When the endpoint software is installed on the Linux endpoint for the first time, the location of the trust chain for the signing certificate is also configured. (A signing certificate is a standard Stealth requirement; it is used to protect the integrity of the settings.xml and crypto.xml files, which are included in the endpoint software package.)

The TOE software is updated by generating a new endpoint installation package on the Enterprise Manager and installing the new installation packages on the Linux endpoint. Each of the component RPMs in a Unisys Stealth

Linux Endpoint package are signed. The Unisys Stealth Linux Endpoint package is not signed because it is created from a base package and provisioning information from the Enterprise Manager. For this reason the user is instructed to generate a SHA-512 hash after the package is created.

After an endpoint package is installed, any changes to the protectionprofile.xml file, such as adding base profiles to the profile priority list or reordering the profile priority list, the endpoint package must be recreated and installed on all affected endpoints for them to use the changes.

To install Stealth; copy an endpoint package to a FIPS-validated Red Hat system and execute the `/<Linux endpoint installation file name>.sh -f` command. The install shell contains binary RPM packages for each supported operating system.

The XML files in the endpoint software package are signed, and that signature is verified by the signing certificate. The signing certificate must be authorized by a trusted root certificate (and optionally by intermediate certificates). The signature is verified by the signing certificate, and in turn, the signing certificate is verified by the trust chain. The trust chain for the signing certificate (which is the trusted root certificate for the Certificate Authority and any intermediate certificates) must be saved in the CA certificates store on the endpoint.

If the Linux endpoint already includes the Stealth endpoint software, the following command is entered to remove this software: `rpm –e $(rpm –qa 'unisys*').` Only the Stealth endpoint software is removed; all configuration information is retained.

The Stealth endpoint cannot download, modify, replace, or update its own binary code. Software can be updated as described above or manually installed.

The command `stconfig –V` is used to obtain the installed version of Stealth endpoint software. The `rpm –qi $(rpm –qa 'unisys*')` command is used to query details on the installed Stealth packages.

6.6.7 FPT_TUD_EXT.2

The Unisys Stealth Endpoint applications are distributed using the format of the platform-supported package manager.

The Unisys Stealth Linux Endpoint packages are delivered to a FIPS-validated Red Hat system via RPM packages for each supported operating system. The Unisys Stealth Windows Endpoint packages are delivered via an .EXE installation package.

6.7 Trusted Channel/Path

6.7.1 FTP_DIT_EXT.1

The Windows Unisys Stealth Endpoint and the Linux Unisys Stealth Endpoint encrypt all transmitted sensitive data with IPsec between other Stealth Endpoints and the Management Server.

See Section 6.1.7 for a description of how the TOE establishes IPsec VPN connections with configured VPN endpoints and the Authentication Server. The resulting VPNs ensure that both ends of the channel are authenticated and the channel protects data from disclosure and modification.

All Stealth endpoints must authenticate with a Stealth Authorization Server. Endpoint authentication is the mechanism used by the Stealth Authorization Server to determine which Communities of Interest (COIs) a Stealth endpoint is authorized to use. There are three authentication schemes supported by the Stealth Authorization Server, Integrated Windows Authentication, LDAP Authentication, and Certificate-Based Authorization.

The LDAP Authentication scheme does pass user credentials to the Stealth Authorization Server. These credentials are encrypted using the 3072-bit public key from the Authorization Server certificate. The resulting encrypted credentials are passed to the Stealth Authorization Server through an IPSec tunnel. Note: The LDAP Authentication scheme should only be used for Stealth Linux Endpoints.

The Unisys Stealth Windows Endpoint and the Unisys Stealth Linux Endpoint are configured to use FIPS Approved algorithms. The CAVP certificate numbers are provided in the table below.

	RHEL 7.4 Kernel on Intel Xeon E5	**RHEL 7.4 OpenSSL on Intel Xeon E5**	**RHEL 7.5 Kernel on Intel Xeon E5**	**RHEL 7.5 OpenSSL on Intel Xeon E5**	**Windows 10 on Intel Core i3, Intel Core i5, Intel Core i7, Intel Pentium**	**Server 2016 on Intel Xeon, Intel Core i5, Intel Core i7, AMD A4**
FCS_CKM.2 Cryptographic Key Establishment						
ECC key pair generation (NIST curves P-256, P-384)		#1144, #1150		#1347, #1350	#C348	#911
Key Agreement					#C211	#92
FCS_COP.1(1) AES Data Encryption/Decryption						
AES CBC 128, 256	#4780, #4781 #4782, #4783 #4784, #4785 #4786, #4787	#4644, #4666 #4695, #4698 #4700	#5545, #5546, #5547, #5548, #5549, #5550, #5551, #5561	#5203, #5205 #5209, #5210 #5212	#C211	#4063, #4074
AES GCM 128 256	#4780, #4781 #4782, #4783 #4784, #4785 #4786, #4787	#4644, #4666 #4695, #4698 #4700	#5545, #5546, #5547, #5548, #5549, #5550, #5551, #5561	#5203, #5205 #5209, #5210 #5212	#C211	#4064
FCS_COP.1(3) Signature Generation and Verification						
RSA 2048 bit	#2619, #2620, #2626	#2535, #2546	#2977, #2978 #2979, #2980	#2786, #2789	#C348	#2192
ECC key pair (NIST curves P-256, P-384)		#1144, #1150		#1347, #1350	#C348	#911
FCS_COP.1(2) Cryptographic hashing						
SHA 1, 256, 384, 512	#3924, #3925 #3930, #3931, #3884, #3885	#3807, #3823 #3842, #3845 #3847	#4451, #4452 #4453, #4454, #4300, #4314	#4193, #4195 #4199, #4200 #4202	#C211	#3347

	RHEL 7.4 Kernel on Intel Xeon E5	**RHEL 7.4 OpenSSL on Intel Xeon E5**	**RHEL 7.5 Kernel on Intel Xeon E5**	**RHEL 7.5 OpenSSL on Intel Xeon E5**	**Windows 10 on Intel Core i3, Intel Core i5, Intel Core i7, Intel Pentium**	**Server 2016 on Intel Xeon, Intel Core i5, Intel Core i7, AMD A4**
FCS_COP.1(4) Keyed-hash message authentication						
HMAC-SHA-256, 384	#3156, #3157 #3188, #3189 #3193, #3194	#3076, #3090 #3107, #3110 #3112	#3546, #3562 #3696, #3697 #3698, #3699	#3445, #3447 #3451, #3452 #3454	#C211	#2651

Table 7: Cryptographic Algorithm Validation Program Certifications

6.8 Timely Security Update (ALC_TSU_EXT.1)

The Unisys Stealth development team follows Unisys software development, productization and release processes to manage the creation and deployment of updates, including security updates, for the Unisys Stealth software.

Unisys has a support services portal at https://www.support.unisys.com that clients use to submit potential security vulnerabilities and other service incidents relating to the products they have licensed.

Unisys uses information derived from client-reported service incidents, Unisys code reviews, publicly available resources and notifications it may receive from third parties to identify, assess, manage and mitigate security vulnerabilities in its products. Unisys works with its third party suppliers to assess the impact of vulnerabilities the suppliers report in their products that may be included in Unisys Stealth. Security advisories, when published by Unisys, may be posted by corporate communication at https://www.unisys.com/about-us/support/security-advisory and pushed to registered clients via a Client Technical Bulletin (CTB).

Unisys strives to provide fixes or workarounds to mitigate security vulnerabilities in supported versions of its Unisys Stealth software as soon as possible, but generally in less than 90 days of becoming aware of public disclosure of a vulnerability impacting the software, depending on the complexity of the issue, its impact and any third party dependencies. Fixes for security vulnerabilities that Unisys has classified as low risk may be deferred until the next regularly scheduled product release. Unisys Stealth software updates are made available for download from secure software repositories accessible by eligible clients entitled to receive Unisys software updates through the Unisys support portal at https://www.support.unisys.com.

7. Protection Profile Claims

This ST is conformant to the *Protection Profile for Application Software,* Version 1.3, 1 March 2019 and the *PP-Module for Virtual Private Network (VPN) Clients,* Version 2.1, 5 October 2017. As explained in Section 3, the Security Problem Definition and the Security Problem Definition of [PP_APP_V1.3] have been included by reference into this ST.

As explained in Section 4, Security Objectives, the Security Objectives of [PP_APP_V1.3] have been included by reference into this ST. The SFRs in this ST are reproduced from [PP_APP_V1.3] and [MOD_VPNC_V2.1].

SFR Iterations

- FCS_STO_EXT.1 is iterated to capture the contrasting behaviors of the Windows and Linux platforms with regards to the storage of credentials.
- FDP_DAR_EXT.1 is iterated to capture the contrasting behaviors of the Windows and Linux platforms with regards to encryption of sensitive application data.
- FDP_RIP.2 is iterated to capture the contrasting behaviors of the TOE, and the TOE platforms with regards to the full residual information protection.
- FPT_LIB_EXT.1 is iterated to capture the different third party libraries used by the Unisys Stealth Windows Endpoint and the Unisys Stealth Linux Endpoint.

8. Rationale

This ST includes by reference Security Problem Definition, Security Objectives, and Security Assurance Requirements from *Protection Profile for Application Software,* Version 1.3, 1 March 2019 and the *PP-Module for Virtual Private Network (VPN) Clients,* Version 2.1, 5 October 2017. The ST makes no additions to the PP assumptions. The PP security functional requirements have been reproduced with the PP operations completed. Operations on the security requirements follow the PP application notes and assurance activities. Consequently, the PP rationale applies, but is incomplete. The TOE Summary Specification rationale below serves to complete the rationale required for the ST.

8.1 TOE Summary Specification Rationale

Section 6, the TOE Summary Specification, describes how the security functions of the TOE meet the claimed SFRs. The following table provides a mapping of the SFRs to the security function descriptions to support the TOE Summary Specification.

	Cryptographic Support	User Data Protection	Identification and Authentication	Security Management	Privacy	TSF Protection	Trusted Channel/Path
FCS_CKM.1(1)	X						
FCS_CKM.1/VPN	X						
FCS_CKM_EXT.1	X						
FCS_CKM.2	X						
FCS_CKM_EXT.2	X						
FCS_CKM_EXT.4	X						
FCS_COP.1(1)	X						
FCS_COP.1(2)	X						
FCS_COP.1(3)	X						
FCS_COP.1(4)	X						
FCS_IPSEC_EXT.1	X						
FCS_RBG_EXT.1	X						
FCS_STO_EXT.1(1)	X						
FCS_STO_EXT.1(2)	X						
FDP_DAR_EXT.1(1)		X					
FDP_DAR_EXT.1(2)		X					
FDP_DEC_EXT.1		X					
FDP_NET_EXT.1		X					
FDP_RIP.2(1)		X					
FDP_RIP.2(2)		X					
FIA_X509_EXT.1			X				
FIA_X509_EXT.2			X				
FMT_CFG_EXT.1				X			
FMT_MEC_EXT.1				X			
FMT_SMF.1/VPN				X			
FMT_SMF.1				X			

	Cryptographic Support	User Data Protection	Identification and Authentication	Security Management	Privacy	TSF Protection	Trusted Channel/Path
FPR_ANO_EXT.1					X		
FPT_AEX_EXT.1						X	
FPT_API_EXT.1						X	
FPT_IDV_EXT.1						X	
FPT_LIB_EXT.1(1)						X	
FPT_LIB_EXT.1(2)						X	
FPT_TST_EXT.1						X	
FPT_TUD_EXT.1						X	
FPT_TUD_EXT.2						X	
FTP_DIT_EXT.1							X

Table 8: Security Functions vs. Requirements Mapping

9. Appendix A - Platform APIs

9.1 Windows Platform APIs

This Section identifies the Windows platform APIs used by the Unisys Stealth Windows Endpoint application.

Windows Platform APIs

This Section identifies the Windows platform APIs used by the Unisys Stealth Windows Endpoint application.

mlstpgw.sys (32-bit)

- ntoskrnl.exe
  - sprintf_s
  - ObfDereferenceObject
  - RtlUnwind
  - ObReferenceObjectByHandle
  - IoReuseIrp
  - IoFreeIrp
  - IoAllocateIrp
  - PsCreateSystemThread
  - strncpy
  - PsTerminateSystemThread
  - MmUnlockPages
  - KeWaitForSingleObject
  - KeWaitForMultipleObjects
  - KeLeaveGuardedRegion
  - KeEnterGuardedRegion
  - KeSetEvent
  - wcstombs
  - ZwQueryValueKey
  - ZwOpenKey
  - ZwClose
  - InterlockedPushEntrySList
  - InterlockedPopEntrySList
  - ExInterlockedFlushSList
  - KeInitializeEvent
  - _wcsicmp
  - wcsncpy
  - wcsncat
  - IoFreeMdl
  - IofCompleteRequest
  - IoAllocateMdl
  - MmUnmapLockedPages
  - MmMapLockedPagesSpecifyCache
  - MmBuildMdlForNonPagedPool
  - ExfInterlockedRemoveHeadList
  - ExfInterlockedInsertTailList
  - KeQuerySystemTime
  - _allmul
  - _alldiv
  - KefReleaseSpinLockFromDpcLevel
  - KefAcquireSpinLockAtDpcLevel
  - KeInitializeSpinLock
  - RtlGetVersion

RtlUnicodeStringToAnsiString
KeGetCurrentThread
wcsncmp
memcpy
_vsnwprintf
IoWriteErrorLogEntry
IoAllocateErrorLogEntry
ExFreePoolWithTag
ExAllocatePoolWithTag
memset
RtlInitUnicodeString

HAL.dll
KfLowerIrql
KeRaiseIrqlToDpcLevel
KfAcquireSpinLock
KfReleaseSpinLock
KeQueryPerformanceCounter

NDIS.SYS
NdisFreeNetBufferList
NdisFreeNetBufferListPool
NdisAllocateNetBufferListPool
NdisFreeMdl
NdisAllocateMdl
NdisFCancelOidRequest
NdisFCancelSendNetBufferLists
NdisFNetPnPEvent
NdisFDevicePnPEventNotify
NdisFIndicateStatus
NdisFOidRequestComplete
NdisFOidRequest
NdisFIndicateReceiveNetBufferLists
NdisFSendNetBufferListsComplete
NdisFReturnNetBufferLists
NdisFSendNetBufferLists
NdisFSetAttributes
NdisAllocateNetBufferAndNetBufferList
NdisFRegisterFilterDriver
NdisQueueIoWorkItem
NdisAllocateIoWorkItem
NdisFreeCloneOidRequest
disAllocateCloneOidRequest
NdisAllocateMemoryWithTagPriority
NdisOpenConfigurationEx
NdisSetEvent
NdisFreeMemory
NdisCloseConfiguration
NdisReadConfiguration
NdisOpenConfigurationKeyByName
NdisFreeIoWorkItem
NdisFreeMemoryWithTag
NdisAllocateMemoryWithTag
NdisGetDeviceReservedExtension
NdisDeregisterDeviceEx
NdisRegisterDeviceEx

NdisFDeregisterFilterDriver

NETIO.SYS
WskCaptureProviderNPI
WskReleaseProviderNPI
WskRegister
WskDeregister

mlstpgw.sys (64-bit)

ntoskrnl.exe
IoFreeIrp
IoReuseIrp
IoAllocateIrp
ObfDereferenceObject
sprintf_s
PsCreateSystemThread
strncpy
PsTerminateSystemThread
MmUnlockPages
KeWaitForSingleObject
KeWaitForMultipleObjects
KeLeaveGuardedRegion
KeEnterGuardedRegion
KeSetEvent
wcstombs
ZwQueryValueKey
ZwOpenKey
ZwClose
ExpInterlockedFlushSList
ExpInterlockedPushEntrySList
ExpInterlockedPopEntrySList
InitializeSListHead
KeInitializeEvent
_wcsicmp
wcsncpy
wcsncat
__C_specific_handler
IoFreeMdl
IofCompleteRequest
IoAllocateMdl
MmUnmapLockedPages
MmMapLockedPagesSpecifyCache
MmBuildMdlForNonPagedPool
ExInterlockedRemoveHeadList
ExInterlockedInsertTailList
KeReleaseSpinLockFromDpcLevel
KeReleaseSpinLock
KeAcquireSpinLockRaiseToDpc
KeAcquireSpinLockAtDpcLevel
KeInitializeSpinLock
KfRaiseIrql
KeLowerIrql
RtlGetVersion
RtlUnicodeStringToAnsiString

wcsncmp
_vsnwprintf
IoWriteErrorLogEntry
IoAllocateErrorLogEntry
ExFreePoolWithTag
ExAllocatePoolWithTag
ObReferenceObjectByHandle
RtlInitUnicodeString

HAL.dll
KeQueryPerformanceCounter

NDIS.SYS
NdisRegisterDeviceEx
NdisDeregisterDeviceEx
NdisAllocateNetBufferAndNetBufferList
NdisFreeNetBufferList
NdisFreeNetBufferListPool
NdisAllocateNetBufferListPool
NdisFreeMdl
NdisAllocateMdl
NdisFCancelOidRequest
NdisFCancelSendNetBufferLists
NdisFNetPnPEvent
NdisFDevicePnPEventNotify
NdisFIndicateStatus
NdisFOidRequestComplete
NdisFOidRequest
NdisFIndicateReceiveNetBufferLists
NdisFSendNetBufferListsComplete
NdisFReturnNetBufferLists
NdisFSendNetBufferLists
NdisFSetAttributes
NdisFDeregisterFilterDriver
NdisFRegisterFilterDriver
NdisQueueIoWorkItem
NdisAllocateIoWorkItem
NdisFreeCloneOidRequest
NdisAllocateCloneOidRequest
NdisAllocateMemoryWithTagPriority
NdisOpenConfigurationEx
NdisSetEvent
NdisFreeMemory
NdisCloseConfiguration
NdisReadConfiguration
NdisOpenConfigurationKeyByName
NdisFreeIoWorkItem
NdisFreeMemoryWithTag
NdisAllocateMemoryWithTag
NdisGetDeviceReservedExtension

NETIO.SYS
WskCaptureProviderNPI
WskReleaseProviderNPI
WskRegister
WskDeregister

stealthii.sys (32-bit)

ntoskrnl.exe

InterlockedPopEntrySList
InterlockedPushEntrySList
mbstowcs
IoAllocateIrp
IoFreeIrp
IoReuseIrp
memmove
PsCreateSystemThread
MmGetSystemRoutineAddress
IoAllocateWorkItem
IoQueueWorkItem
ExInitializeNPagedLookasideList
ExDeleteNPagedLookasideList
ExInterlockedFlushSList
KeWaitForSingleObject
KeInitializeTimer
KeSetEvent
KeInitializeEvent
KeFlushQueuedDpcs
KeInitializeDpc
RtlGetVersion
RtlInitUnicodeString
RtlCopyUnicodeString
KeGetCurrentThread
memset
memcpy
IoFreeWorkItem
IoFreeMdl
KeSetTimer
KeCancelTimer
KeLeaveGuardedRegion
KeEnterGuardedRegion
rand
PsTerminateSystemThread
KeClearEvent
_allmul
_alldiv
ZwQueryValueKey
ZwOpenKey
ZwClose
ObfDereferenceObject
ObReferenceObjectByHandle
IoRegisterShutdownNotification
IoDeleteSymbolicLink
IoDeleteDevice
IoCreateSymbolicLink
KeInitializeSpinLock
IoCreateDevice
IofCompleteRequest
IoAllocateMdl
MmUnmapLockedPages

MmMapLockedPagesSpecifyCache
MmBuildMdlForNonPagedPool
ExFreePoolWithTag
ExAllocatePoolWithTag
RtlCompareMemory
RtlQueryRegistryValues
RtlUnicodeStringToAnsiString

HAL.dll
KfLowerIrql
KeRaiseIrqlToDpcLevel
KeAcquireInStackQueuedSpinLock
KeReleaseInStackQueuedSpinLock
KeQueryPerformanceCounter
KeGetCurrentIrql
KfAcquireSpinLock
KfReleaseSpinLock

NDIS.SYS
NdisFreeNetBufferListPool
NdisAllocateNetBufferListPool
NdisAdvanceNetBufferDataStart
NdisRetreatNetBufferDataStart

fwpkclnt.sys
FwpsGetPacketListSecurityInformation0
FwpsFreeNetBufferList0
FwpsAllocateNetBufferAndNetBufferList0
FwpsQueryPacketInjectionState0
FwpsInjectTransportReceiveAsync0
FwpsInjectTransportSendAsync1
FwpsConstructIpHeaderForTransportPacket0
FwpsFreeCloneNetBufferList0
FwpsAllocateCloneNetBufferList0
FwpsCompleteOperation0
FwpsPendOperation0
FwpsInjectionHandleDestroy0
FwpsInjectionHandleCreate0
FwpsCalloutUnregisterById0
FwpsCalloutRegister2

NETIO.SYS
WskReleaseProviderNPI
WskDeregister
WskRegister
WskCaptureProviderNPI

WDFLDR.SYS
WdfVersionUnbind
WdfVersionBindClass
WdfVersionBind
WdfVersionUnbindClass

stealthii.sys (64-bit)

```
ntoskrnl.exe
        rand
        KeEnterGuardedRegion
        KeLeaveGuardedRegion
        KeCancelTimer
        KeSetTimer
        KeAcquireSpinLockRaiseToDpc
        KeReleaseSpinLock
        InitializeSListHead
        ExpInterlockedPopEntrySList
        ExpInterlockedPushEntrySList
        ExpInterlockedFlushSList
        RtlGetVersion
        IoAllocateIrp
        IoFreeIrp
        IoReuseIrp
        PsTerminateSystemThread
        MmGetSystemRoutineAddress
        IoAllocateWorkItem
        IoQueueWorkItem
        ExQueryDepthSList
        ExInitializeNPagedLookasideList
        ExDeleteNPagedLookasideList
        RtlInitUnicodeString
        IoFreeWorkItem
        RtlCopyUnicodeString
        IoFreeMdl
        IofCompleteRequest
        IoAllocateMdl
        MmUnmapLockedPages
        MmMapLockedPagesSpecifyCache
        KeClearEvent
        ZwQueryValueKey
        ZwOpenKey
        ZwClose
        ObfDereferenceObject
        ObReferenceObjectByHandle
        IoRegisterShutdownNotification
        IoDeleteSymbolicLink
        IoDeleteDevice
        IoCreateSymbolicLink
        IoCreateDevice
        PsCreateSystemThread
        KeInitializeSpinLock
        KeWaitForSingleObject
        KeInitializeTimer
        KeSetEvent
        KeInitializeEvent
        KeFlushQueuedDpcs
        RtlQueryRegistryValues
        KeInitializeDpc
        MmBuildMdlForNonPagedPool
        ExFreePoolWithTag
        ExAllocatePoolWithTag
        KeReleaseInStackQueuedSpinLock
        KeAcquireInStackQueuedSpinLock
```

KfRaiseIrql
KeLowerIrql
RtlCompareMemory
mbstowcs
RtlUnicodeStringToAnsiString

HAL.dll
KeQueryPerformanceCounter

NDIS.SYS
NdisRetreatNetBufferDataStart
NdisAdvanceNetBufferDataStart
NdisFreeNetBufferListPool
NdisAllocateNetBufferListPool

fwpkclnt.sys
FwpsGetPacketListSecurityInformation0
FwpsFreeNetBufferList0
FwpsAllocateNetBufferAndNetBufferList0
FwpsQueryPacketInjectionState0
FwpsInjectTransportReceiveAsync0
FwpsInjectTransportSendAsync1
FwpsConstructIpHeaderForTransportPacket0
FwpsFreeCloneNetBufferList0
FwpsAllocateCloneNetBufferList0
FwpsCompleteOperation0
FwpsPendOperation0
FwpsInjectionHandleDestroy0
FwpsInjectionHandleCreate0
FwpsCalloutUnregisterById0
FwpsCalloutRegister2

NETIO.SYS
WskReleaseProviderNPI
WskCaptureProviderNPI
WskRegister
WskDeregister

WDFLDR.SYS
WdfVersionUnbind
WdfVersionBindClass
WdfVersionBind
WdfVersionUnbindClass

USSL_Logon.exe (32-bit)

USERENV.dll
ExpandEnvironmentStringsForUserW

WTSAPI32.dll
WTSQueryUserToken

WS2_32.dll
WSARecv
WSASocketW

WSAWaitForMultipleEvents
getaddrinfo
inet_ntop
FreeAddrInfoW
InetNtopW
GetAddrInfoW
WSACloseEvent
WSAAccept
WSAGetOverlappedResult
WSASend
WSACreateEvent

ADVAPI32.dll

LsaNtStatusToWinError
CryptSetProvParam
DuplicateToken
OpenProcessToken
StartServiceW
StartServiceCtrlDispatcherW
SetServiceStatus
RegisterServiceCtrlHandlerExW
QueryServiceStatus
OpenServiceW
OpenSCManagerW
DeleteService
CreateServiceW
ControlService
CloseServiceHandle
ChangeServiceConfig2W
RegQueryInfoKeyW
RegEnumKeyExW
RegDeleteKeyW
RegCreateKeyExW
GetUserNameW
DeregisterEventSource
RegisterEventSourceW
GetUserNameA
CryptAcquireContextW
CryptReleaseContext
CryptGenKey
CryptDestroyKey
CryptGetKeyParam
CryptGetHashParam
CryptGetProvParam
CryptGenRandom
CryptGetUserKey
CryptEncrypt
CryptDecrypt
CryptCreateHash
CryptHashData
CryptDestroyHash
RegCloseKey
RegOpenKeyExW
RegQueryValueExW
RegSetValueExW
RevertToSelf

RegDeleteValueW
ImpersonateLoggedOnUser
ReportEventW

Secur32.dll

DeleteSecurityContext
ApplyControlToken
QueryContextAttributesW
CompleteAuthToken
LsaFreeReturnBuffer
LsaEnumerateLogonSessions
AcquireCredentialsHandleW
LsaGetLogonSessionData
AcceptSecurityContext
FreeContextBuffer
FreeCredentialsHandle
EncryptMessage
DecryptMessage
QuerySecurityPackageInfoW

CRYPT32.dll

CryptBinaryToStringW
CertVerifyCertificateChainPolicy
CertDuplicateCertificateContext
CryptDecodeObjectEx
CertVerifyRevocation
CertNameToStrW
CryptMsgClose
CryptQueryObject
CryptMsgGetParam
CertStrToNameW
CertGetNameStringA
CertCreateSelfSignCertificate
CryptSetKeyIdentifierProperty
CertFreeCertificateChainEngine
CertGetCertificateChain
CertFreeCertificateChain
CertAddEncodedCertificateToStore
CertGetNameStringW
CryptCreateKeyIdentifierFromCSP
CryptStringToBinaryW
CryptImportPublicKeyInfoEx2
CryptAcquireCertificatePrivateKey
CryptImportPublicKeyInfo
CryptExportPublicKeyInfoEx
CertGetIntendedKeyUsage
CertFindExtension
CertVerifyTimeValidity
CryptSignAndEncodeCertificate
CertGetPublicKeyLength
CertGetEnhancedKeyUsage
CertDeleteCertificateFromStore
CertAddCertificateContextToStore
CertGetCertificateContextProperty
CertSetCertificateContextProperty
CertFreeCertificateContext

CryptEncodeObjectEx
CryptEncodeObject
CryptDecodeObject
CertOpenStore
CertCloseStore
CertFindCertificateInStore
CertCreateCertificateContext

CRYPTUI.dll
CryptUIWizImport
CryptUIWizExport
CryptUIWizDigitalSign

WinSCard.dll
SCardFreeMemory
SCardEstablishContext
SCardReleaseContext
SCardListReadersW
SCardGetCardTypeProviderNameW
SCardGetStatusChangeW
SCardListCardsW

IPHLPAPI.DLL
GetUnicastIpAddressTable
FreeMibTable
GetIpAddrTable

RPCRT4.dll
RpcBindingUnbind
RpcStringFreeW
RpcBindingFromStringBindingW
UuidCreate
RpcStringBindingComposeW
NdrClientCall2

ntdll.dll
RtlIpv6StringToAddressW
RtlIpv4AddressToStringExA
RtlIpv6AddressToStringExA
RtlIpv6StringToAddressA
RtlIpv4StringToAddressW
RtlIpv4StringToAddressA

KERNEL32.dll
GetSystemTimeAsFileTime
SetUnhandledExceptionFilter
IsProcessorFeaturePresent
WaitForSingleObjectEx
GetStartupInfoW
VirtualQuery
QueryPerformanceCounter
GetCurrentProcessId
UnhandledExceptionFilter
IsDebuggerPresent
DeleteTimerQueueTimer
CreateTimerQueueTimer

InterlockedPushEntrySList
InterlockedFlushSList
WriteFile
CompareFileTime
VerifyVersionInfoW
VerSetConditionMask
GetSystemInfo
GetTimeZoneInformation
TerminateProcess
InitializeSListHead
GetComputerNameExW
FindClose
FindNextFileW
FindFirstFileW
GetFileSize
LocalAlloc
MoveFileW
CreateDirectoryW
LocalFree
CreateFileW
ReadFile
CloseHandle
GetLastError
GetSystemTime
FileTimeToSystemTime
SystemTimeToFileTime
MultiByteToWideChar
DecodePointer
RaiseException
InitializeCriticalSectionEx
DeleteCriticalSection
OutputDebugStringW
HeapAlloc
HeapFree
GetProcessHeap
EnterCriticalSection
LeaveCriticalSection
Sleep
GetModuleHandleW
GetProcAddress
SetHandleInformation
CreatePipe
PeekNamedPipe
SetEvent
ResetEvent
WaitForSingleObject
CreateEventW
WaitForMultipleObjects
GetExitCodeProcess
CreateProcessW
FreeLibrary
LoadLibraryW
GetCommandLineW
HeapDestroy
HeapReAlloc
HeapSize

GetCurrentThreadId
FindResourceExW
GetModuleFileNameW
LoadLibraryExW
LoadResource
LockResource
SizeofResource
FindResourceW
lstrcmpiW
GetCurrentProcess
WTSGetActiveConsoleSessionId
lstrlenW
WideCharToMultiByte
OpenEventW
CreateTimerQueue
DeleteTimerQueueEx
DebugBreak
GetLocalTime
DeleteFileW

USER32.dll
LoadStringW
wsprintfW
PostThreadMessageW
CharNextW
MessageBoxW
UnregisterClassW

ole32.dll
CoTaskMemFree
CoFreeUnusedLibraries
CoCreateGuid
CoUninitialize
CoInitializeEx
CoAddRefServerProcess
CoReleaseServerProcess
CoCreateInstance
CoTaskMemAlloc
CoTaskMemRealloc

SHELL32.dll
SHGetKnownFolderPath

SHLWAPI.dll
PathStripToRootW
PathRemoveFileSpecW

ncrypt.dll
BCryptGenerateKeyPair
NCryptImportKey
BCryptDecrypt
BCryptGetProperty
NCryptExportKey
NCryptFinalizeKey
BCryptEncrypt
BCryptDestroyKey

- NCryptFreeObject
- NCryptGetProperty
- BCryptFinalizeKeyPair
- BCryptGenerateSymmetricKey
- NCryptDecrypt
- BCryptCreateHash
- NCryptSetProperty
- BCryptSetProperty
- BCryptHashData
- NCryptEncrypt
- BCryptImportKeyPair
- NCryptCreatePersistedKey
- BCryptDestroyHash
- BCryptCloseAlgorithmProvider
- BCryptFinishHash
- BCryptExportKey
- BCryptGenRandom
- NCryptOpenKey
- NCryptOpenStorageProvider
- BCryptOpenAlgorithmProvider
- BCryptImportKey

fwpuclnt.dll
- FwpmEngineOpen0
- IPsecGetStatistics1
- IPsecSaContextDestroyEnumHandle0
- IPsecSaContextCreateEnumHandle0
- IPsecSaCreateEnumHandle0
- IPsecSaDestroyEnumHandle0
- FwpmEngineClose0
- IPsecSaEnum1
- IPsecSaContextEnum1

CRYPTXML.dll
- CryptXmlClose
- CryptXmlVerifySignature
- CryptXmlGetDocContext
- CryptXmlGetStatus
- CryptXmlOpenToDecode
- CryptXmlEncode
- CryptXmlSign
- CryptXmlCreateReference
- CryptXmlOpenToEncode

WINTRUST.dll
- WinVerifyTrust

WINHTTP.dll
- WinHttpSetTimeouts
- WinHttpSetStatusCallback
- WinHttpSetCredentials
- WinHttpReceiveResponse
- WinHttpOpen
- WinHttpAddRequestHeaders
- WinHttpQueryHeaders
- WinHttpReadData

WinHttpOpenRequest
WinHttpSetOption
WinHttpCloseHandle
WinHttpCreateUrl
WinHttpQueryAuthSchemes
WinHttpSendRequest
WinHttpConnect
WinHttpCrackUrl
WinHttpQueryDataAvailable

USSL_Logon.exe (64-bit)

WTSAPI32.dll
WTSQueryUserToken

WS2_32.dll
WSAWaitForMultipleEvents
WSACloseEvent
WSASend
WSAAccept
getaddrinfo
inet_ntop
InetNtopW
FreeAddrInfoW
GetAddrInfoW
WSASocketW
WSACreateEvent
WSAGetOverlappedResult
WSARecv

ADVAPI32.dll
LsaNtStatusToWinError
CryptSetProvParam
DuplicateToken
OpenProcessToken
SetServiceStatus
RegisterServiceCtrlHandlerExW
StartServiceCtrlDispatcherW
DeleteService
ControlService
DeregisterEventSource
ChangeServiceConfig2W
CreateServiceW
CloseServiceHandle
OpenServiceW
OpenSCManagerW
RegCreateKeyExW
RegEnumKeyExW
RegQueryInfoKeyW
RegDeleteKeyW
ReportEventW
RegisterEventSourceW
CryptReleaseContext
CryptAcquireContextW
CryptGetUserKey

```
          CryptGenKey
          CryptGenRandom
          CryptDestroyKey
          GetUserNameA
          CryptGetKeyParam
          RegOpenKeyExW
          RegCloseKey
          RegQueryValueExW
          RegSetValueExW
          RevertToSelf
          RegDeleteValueW
          ImpersonateLoggedOnUser
          GetUserNameW

CRYPT32.dll
          CryptImportPublicKeyInfoEx2
          CertVerifyCertificateChainPolicy
          CryptBinaryToStringW
          CryptDecodeObjectEx
          CertDuplicateCertificateContext
          CertVerifyRevocation
          CertNameToStrW
          CryptMsgClose
          CryptMsgGetParam
          CryptQueryObject
          CryptStringToBinaryW
          CertGetNameStringW
          CertAddEncodedCertificateToStore
          CertFreeCertificateChain
          CertGetCertificateChain
          CertDeleteCertificateFromStore
          CertAddCertificateContextToStore
          CertFreeCertificateContext
          CertGetNameStringA
          CertGetCertificateContextProperty
          CertOpenStore
          CertFindCertificateInStore
          CertCloseStore
          CryptAcquireCertificatePrivateKey
          CertStrToNameW
          CryptEncodeObject
          CryptEncodeObjectEx
          CertFindExtension
          CryptDecodeObject
          CryptCreateKeyIdentifierFromCSP
          CryptSetKeyIdentifierProperty
          CertCreateSelfSignCertificate
          CryptExportPublicKeyInfoEx
          CryptSignAndEncodeCertificate
          CertCreateCertificateContext
          CertSetCertificateContextProperty
          CertVerifyTimeValidity

CRYPTUI.dll
          CryptUIWizExport
```

Secur32.dll

- AcquireCredentialsHandleW
- ApplyControlToken
- DecryptMessage
- DeleteSecurityContext
- LsaEnumerateLogonSessions
- LsaGetLogonSessionData
- AcceptSecurityContext
- LsaFreeReturnBuffer
- FreeCredentialsHandle
- EncryptMessage
- CompleteAuthToken
- QuerySecurityPackageInfoW
- FreeContextBuffer
- QueryContextAttributesW

WinSCard.dll

- SCardEstablishContext
- SCardListReadersW
- SCardFreeMemory
- SCardGetCardTypeProviderNameW
- SCardReleaseContext
- SCardListCardsW
- SCardGetStatusChangeW

IPHLPAPI.DLL

- GetUnicastIpAddressTable
- FreeMibTable

RPCRT4.dll

- RpcBindingUnbind
- RpcStringFreeW
- RpcStringBindingComposeW
- UuidCreate
- RpcBindingFromStringBindingW
- NdrClientCall2

ntdll.dll

- RtlIpv4StringToAddressW
- RtlIpv6StringToAddressW
- RtlIpv4StringToAddressA
- RtlIpv6StringToAddressA
- RtlCaptureContext
- RtlLookupFunctionEntry
- RtlVirtualUnwind

KERNEL32.dll

- VirtualQuery
- TerminateProcess
- IsProcessorFeaturePresent
- WaitForSingleObjectEx
- QueryPerformanceCounter
- GetCurrentProcessId
- GetSystemTimeAsFileTime
- IsDebuggerPresent
- SetUnhandledExceptionFilter

CreateTimerQueueTimer
WriteFile
CompareFileTime
InterlockedFlushSList
InterlockedPushEntrySList
InitializeSListHead
GetTimeZoneInformation
VerifyVersionInfoW
GetComputerNameExW
GetSystemInfo
UnhandledExceptionFilter
DeleteTimerQueueTimer
FindNextFileW
FindFirstFileW
FindClose
VerSetConditionMask
LocalAlloc
GetFileSize
MoveFileW
CreateDirectoryW
LocalFree
GetLastError
GetSystemTime
SystemTimeToFileTime
FileTimeToSystemTime
CreateFileW
ReadFile
CloseHandle
MultiByteToWideChar
InitializeCriticalSectionEx
RaiseException
DecodePointer
DeleteCriticalSection
EnterCriticalSection
LeaveCriticalSection
OutputDebugStringW
HeapFree
GetProcessHeap
HeapAlloc
GetProcAddress
GetModuleHandleW
Sleep
CreateEventW
ResetEvent
WaitForSingleObject
GetExitCodeProcess
WaitForMultipleObjects
SetEvent
CreatePipe
SetHandleInformation
PeekNamedPipe
CreateProcessW
LoadLibraryW
FreeLibrary
HeapDestroy
HeapReAlloc

```
          HeapSize
          lstrcmpiW
          GetModuleFileNameW
          SizeofResource
          LoadResource
          FindResourceW
          LoadLibraryExW
          GetCommandLineW
          GetCurrentThreadId
          LockResource
          FindResourceExW
          GetCurrentProcess
          WTSGetActiveConsoleSessionId
          lstrlenW
          WideCharToMultiByte
          OpenEventW
          CreateTimerQueue
          DeleteTimerQueueEx
          DebugBreak
          GetLocalTime
          DeleteFileW

USER32.dll
          LoadStringW
          wsprintfW
          CharNextW
          MessageBoxW
          PostThreadMessageW

ole32.dll
          CoAddRefServerProcess
          CoFreeUnusedLibraries
          CoCreateGuid
          CoUninitialize
          CoInitializeEx
          CoReleaseServerProcess
          CoTaskMemAlloc
          CoCreateInstance
          CoTaskMemRealloc
          CoTaskMemFree

SHELL32.dll
          SHGetKnownFolderPath

SHLWAPI.dll
          PathStripToRootW
          PathRemoveFileSpecW

ncrypt.dll
          BCryptDecrypt
          BCryptEncrypt
          BCryptGenerateKeyPair
          BCryptGenerateSymmetricKey
          BCryptSetProperty
          BCryptGetProperty
          BCryptOpenAlgorithmProvider
```

BCryptDestroyKey
NCryptFreeObject
NCryptGetProperty
BCryptCloseAlgorithmProvider
NCryptExportKey
NCryptImportKey
NCryptDecrypt
NCryptEncrypt
NCryptFinalizeKey
NCryptSetProperty
NCryptCreatePersistedKey
NCryptOpenKey
NCryptOpenStorageProvider
BCryptGenRandom
BCryptDestroyHash
BCryptFinishHash
BCryptHashData
BCryptCreateHash
BCryptFinalizeKeyPair
BCryptImportKeyPair
BCryptImportKey
BCryptExportKey

fwpuclnt.dll
FwpmEngineClose0
IPsecGetStatistics1
IPsecSaContextCreateEnumHandle0
IPsecSaContextEnum1
IPsecSaContextDestroyEnumHandle0
IPsecSaCreateEnumHandle0
FwpmEngineOpen0
IPsecSaEnum1
IPsecSaDestroyEnumHandle0

CRYPTXML.dll
CryptXmlGetDocContext
CryptXmlGetStatus
CryptXmlVerifySignature
CryptXmlCreateReference
CryptXmlEncode
CryptXmlSign
CryptXmlOpenToDecode
CryptXmlOpenToEncode
CryptXmlClose

WINTRUST.dll
WinVerifyTrust

WINHTTP.dll
WinHttpQueryDataAvailable
WinHttpSetTimeouts
WinHttpSetCredentials
WinHttpSetStatusCallback
WinHttpQueryHeaders
WinHttpReceiveResponse
WinHttpQueryAuthSchemes

WinHttpSendRequest
WinHttpAddRequestHeaders
WinHttpOpenRequest
WinHttpSetOption
WinHttpCreateUrl
WinHttpReadData
WinHttpConnect
WinHttpCloseHandle
WinHttpOpen
WinHttpCrackUrl

USSL_PreLogon.exe (32-bit)

RPCRT4.dll
- RpcBindingUnbind
- RpcStringFreeW
- RpcBindingFromStringBindingW
- RpcStringBindingComposeW
- UuidCreate
- NdrClientCall2

WS2_32.dll
- WSACloseEvent
- WSAAccept
- WSARecv
- WSAResetEvent
- WSAGetOverlappedResult
- WSASend
- WSACreateEvent
- WSAWaitForMultipleEvents
- WSASocketW

KERNEL32.dll
- FindResourceExW
- ResetEvent
- WaitForMultipleObjects
- CreateDirectoryW
- DeleteFileW
- GetLocalTime
- SetEvent
- LocalAlloc
- LocalFree
- GetComputerNameExW
- WideCharToMultiByte
- CreateEventW
- GetCommandLineW
- LoadLibraryExW
- FindResourceW
- LoadResource
- SizeofResource
- GetModuleHandleW
- LockResource
- FreeLibrary
- GetModuleFileNameW
- CloseHandle

```
        lstrcmpiW
        MultiByteToWideChar
        HeapSize
        HeapReAlloc
        HeapDestroy
        Sleep
        HeapAlloc
        GetProcessHeap
        HeapFree
        OutputDebugStringW
        LeaveCriticalSection
        EnterCriticalSection
        DeleteCriticalSection
        DecodePointer
        RaiseException
        GetLastError
        InitializeCriticalSectionEx
        IsDebuggerPresent
        UnhandledExceptionFilter
        SetUnhandledExceptionFilter
        GetCurrentProcess
        TerminateProcess
        IsProcessorFeaturePresent
        WaitForSingleObjectEx
        QueryPerformanceCounter
        GetCurrentProcessId
        GetSystemTimeAsFileTime
        InitializeSListHead
        GetCurrentThreadId
        GetProcAddress
        WaitForSingleObject
        MoveFileW
        VirtualQuery

USER32.dll
        wsprintfW
        MessageBoxW
        LoadStringW
        CharNextW
        PostThreadMessageW

ADVAPI32.dll
        RegisterServiceCtrlHandlerExW
        RegCloseKey
        RegDeleteKeyW
        RegQueryInfoKeyW
        RegEnumKeyExW
        RegSetValueExW
        RegCreateKeyExW
        RegDeleteValueW
        OpenSCManagerW
        OpenServiceW
        CloseServiceHandle
        CreateServiceW
        ChangeServiceConfig2W
        ControlService
```

DeleteService
RegisterEventSourceW
ReportEventW
RegOpenKeyExW
StartServiceCtrlDispatcherW
RegQueryValueExW
SetServiceStatus

ole32.dll
CoTaskMemAlloc
CoTaskMemFree
CoTaskMemRealloc
CoInitializeEx
CoReleaseServerProcess
CoAddRefServerProcess
CoFreeUnusedLibraries
CoUninitialize
CoCreateInstance

SHLWAPI.dll
PathRemoveFileSpecW

fwpuclnt.dll
IPsecSaContextCreateEnumHandle0
IPsecGetStatistics1
IPsecSaContextDestroyEnumHandle0
IPsecSaContextEnum1
FwpmEngineClose0
FwpmEngineOpen0

ncrypt.dll
BCryptDestroyKey
BCryptCloseAlgorithmProvider

WINTRUST.dll
WinVerifyTrust

CRYPT32.dll
CertCloseStore
CryptQueryObject
CryptMsgClose
CertFreeCertificateContext
CertFindCertificateInStore
CertGetNameStringW
CryptMsgGetParam

USSL_PreLogon.exe (64-bit)

RPCRT4.dll
RpcBindingUnbind
RpcStringFreeW
RpcStringBindingComposeW
RpcBindingFromStringBindingW
UuidCreate

```
          NdrClientCall2

WS2_32.dll
          FreeAddrInfoW
          InetNtopW
          WSASocketW
          WSAWaitForMultipleEvents
          WSASend
          WSAResetEvent
          WSARecv
          WSAGetOverlappedResult
          WSACreateEvent
          WSACloseEvent
          WSAAccept
          GetAddrInfoW

KERNEL32.dll
          MultiByteToWideChar
          ResetEvent
          WaitForMultipleObjects
          CreateDirectoryW
          DeleteFileW
          GetLocalTime
          LockResource
          CreateFileW
          GetFileSize
          ReadFile
          LocalAlloc
          LocalFree
          VerSetConditionMask
          FindClose
          FindFirstFileW
          FindNextFileW
          GetSystemInfo
          GetComputerNameExW
          VerifyVersionInfoW
          GetTimeZoneInformation
          WideCharToMultiByte
          WriteFile
          CompareFileTime
          LoadResource
          LoadLibraryExW
          GetProcAddress
          GetModuleHandleW
          GetModuleFileNameW
          FreeLibrary
          FindResourceExW
          lstrcmpiW
          CreateEventW
          WaitForSingleObject
          SetEvent
          HeapSize
          HeapReAlloc
          HeapDestroy
          CloseHandle
          GetCommandLineW
```

```
        Sleep
        LeaveCriticalSection
        EnterCriticalSection
        GetProcessHeap
        HeapFree
        HeapAlloc
        OutputDebugStringW
        DeleteCriticalSection
        InitializeCriticalSectionEx
        GetLastError
        RaiseException
        DecodePointer
        IsDebuggerPresent
        UnhandledExceptionFilter
        SetUnhandledExceptionFilter
        GetCurrentProcess
        TerminateProcess
        IsProcessorFeaturePresent
        WaitForSingleObjectEx
        QueryPerformanceCounter
        GetCurrentProcessId
        GetSystemTimeAsFileTime
        InitializeSListHead
        FindResourceW
        GetCurrentThreadId
        SizeofResource
        MoveFileW

USER32.dll
        LoadStringW
        PostThreadMessageW
        wsprintfW
        MessageBoxW
        CharNextW

ADVAPI32.dll
        StartServiceCtrlDispatcherW
        RegOpenKeyExW
        RegCreateKeyExW
        RegDeleteKeyW
        RegDeleteValueW
        RegEnumKeyExW
        RegQueryInfoKeyW
        RegSetValueExW
        ChangeServiceConfig2W
        CloseServiceHandle
        ControlService
        CreateServiceW
        DeleteService
        OpenSCManagerW
        OpenServiceW
        ImpersonateLoggedOnUser
        RegQueryValueExW
        ReportEventW
        RegisterEventSourceW
        RegisterServiceCtrlHandlerExW
        RegCloseKey
        SetServiceStatus
        DeregisterEventSource

SHELL32.dll
        SHGetKnownFolderPath

ole32.dll
        CoTaskMemRealloc
        CoTaskMemAlloc
        CoFreeUnusedLibraries
        CoCreateInstance
        CoReleaseServerProcess
        CoAddRefServerProcess
        CoInitializeEx
        CoTaskMemFree
        CoUninitialize

SHLWAPI.dll
        PathStripToRootW
        PathRemoveFileSpecW

fwpuclnt.dll
        IPsecSaContextDestroyEnumHandle0
        IPsecSaEnum1
        IPsecSaContextEnum1
        IPsecSaContextCreateEnumHandle0
        IPsecGetStatistics1
        IPsecSaDestroyEnumHandle0
        FwpmEngineClose0
        FwpmEngineOpen0
        IPsecSaCreateEnumHandle0

ncrypt.dll
        NCryptGetProperty
        BCryptDestroyKey
        BCryptOpenAlgorithmProvider
        BCryptGetProperty
        BCryptSetProperty
        BCryptCloseAlgorithmProvider
        BCryptGenerateSymmetricKey
        NCryptFreeObject
        BCryptGenerateKeyPair
        BCryptEncrypt
        BCryptDecrypt
        BCryptExportKey
        BCryptImportKey
        BCryptImportKeyPair
        BCryptFinalizeKeyPair
        BCryptCreateHash
        BCryptHashData
        NCryptExportKey
        NCryptDecrypt
        NCryptEncrypt
        NCryptFinalizeKey
        NCryptSetProperty
        NCryptImportKey
        NCryptCreatePersistedKey
        NCryptOpenKey
        NCryptOpenStorageProvider
        BCryptGenRandom
        BCryptDestroyHash
        BCryptFinishHash

CRYPTXML.dll
        CryptXmlGetDocContext
        CryptXmlVerifySignature
        CryptXmlOpenToDecode
        CryptXmlGetStatus
        CryptXmlClose

WINHTTP.dll
        WinHttpSetCredentials
        WinHttpQueryHeaders
        WinHttpReceiveResponse
        WinHttpQueryAuthSchemes
        WinHttpSendRequest
        WinHttpAddRequestHeaders
        WinHttpOpenRequest
        WinHttpSetOption
        WinHttpQueryDataAvailable
        WinHttpReadData
        WinHttpConnect
        WinHttpCloseHandle
        WinHttpCrackUrl
        WinHttpCreateUrl
        WinHttpOpen

IPHLPAPI.DLL
        GetUnicastIpAddressTable
        FreeMibTable

ntdll.dll
        RtlLookupFunctionEntry
        RtlCaptureContext
        RtlIpv6StringToAddressW
        RtlIpv4StringToAddressW
        RtlVirtualUnwind

WINTRUST.dll
        WinVerifyTrust

CRYPT32.dll
        CertCreateCertificateContext
        CertVerifyRevocation
        CryptMsgGetParam
        CryptMsgClose
        CryptBinaryToStringW
        CryptStringToBinaryW
        CertVerifyCertificateChainPolicy
        CertFreeCertificateChain
        CertGetCertificateChain
```

CertVerifyTimeValidity
CertFreeCertificateContext
CryptImportPublicKeyInfoEx2
CertFindCertificateInStore
CertGetNameStringW
CryptQueryObject
CertOpenStore
CertDuplicateCertificateContext
CertNameToStrW
CertCloseStore
CryptDecodeObjectEx

USSL_Protocol.exe (32-bit)

CRYPT32.dll
CryptBinaryToStringW
CertVerifyCertificateChainPolicy
CryptMsgClose
CryptQueryObject
CryptMsgGetParam
CertCreateCertificateContext
CertGetNameStringW
CertFindCertificateInStore
CryptImportPublicKeyInfoEx2
CertFreeCertificateChain
CertCloseStore
CertDuplicateCertificateContext
CertNameToStrW
CertVerifyRevocation
CertVerifyTimeValidity
CertGetCertificateChain
CryptDecodeObjectEx
CryptStringToBinaryW
CertOpenStore
CertFreeCertificateContext

IPHLPAPI.DLL
FreeMibTable
CancelMibChangeNotify2
NotifyUnicastIpAddressChange
GetUnicastIpAddressTable
NotifyIpInterfaceChange
GetIpInterfaceEntry
GetIfEntry2
GetAdaptersAddresses
GetUnicastIpAddressEntry

KERNEL32.dll
InterlockedPushEntrySList
InitializeSListHead
ReadFile
LocalAlloc
GetFileSize
GetTimeZoneInformation
GetSystemInfo

VerSetConditionMask
WideCharToMultiByte
VerifyVersionInfoW
CreateTimerQueue
InterlockedFlushSList
WriteFile
GetSystemTimeAsFileTime
GetCurrentProcessId
QueryPerformanceCounter
GetThreadId
GetExitCodeThread
DebugBreak
LocalFree
MultiByteToWideChar
lstrcmpiW
MoveFileW
SizeofResource
LockResource
LoadResource
LoadLibraryExW
GetProcAddress
GetModuleHandleW
GetModuleFileNameW
FreeLibrary
FindResourceExW
GetCurrentThreadId
WaitForSingleObject
HeapSize
HeapReAlloc
HeapDestroy
CloseHandle
GetCommandLineW
SetEvent
GetOverlappedResult
WaitForSingleObjectEx
IsProcessorFeaturePresent
TerminateProcess
GetCurrentProcess
SetUnhandledExceptionFilter
UnhandledExceptionFilter
IsDebuggerPresent
DeviceIoControl
CreateFileW
Sleep
HeapAlloc
WaitForMultipleObjects
GetLocalTime
DeleteFileW
GetComputerNameExW
CompareFileTime
DeleteTimerQueueTimer
CreateTimerQueueTimer
CreateDirectoryW
FindResourceW
DeleteTimerQueueEx
DecodePointer

```
        CreateEventW
        ResetEvent
        LeaveCriticalSection
        RaiseException
        GetLastError
        InitializeCriticalSectionEx
        DeleteCriticalSection
        EnterCriticalSection
        GetProcessHeap
        HeapFree
        OutputDebugStringW

USER32.dll
        PostThreadMessageW
        LoadStringW
        CharNextW
        MessageBoxW
        wsprintfW

ADVAPI32.dll
        RegCloseKey
        CreateServiceW
        DeleteService
        RegCreateKeyExW
        RegDeleteKeyW
        RegDeleteValueW
        RegEnumKeyExW
        RegOpenKeyExW
        RegQueryInfoKeyW
        RegSetValueExW
        ChangeServiceConfig2W
        CloseServiceHandle
        OpenSCManagerW
        OpenServiceW
        RegisterServiceCtrlHandlerExW
        SetServiceStatus
        StartServiceCtrlDispatcherW
        ReportEventW
        RegisterEventSourceW
        RegQueryValueExW
        ControlService

SHELL32.dll
        SHGetKnownFolderPath

ole32.dll
        CoTaskMemAlloc
        CoTaskMemRealloc
        CoUninitialize
        CoFreeUnusedLibraries
        CoInitializeEx
        CoAddRefServerProcess
        CoReleaseServerProcess
        CoTaskMemFree
        CoCreateInstance
```

```
SHLWAPI.dll
        PathStripToRootW
        PathRemoveFileSpecW

fwpuclnt.dll
        IPsecSaContextAddInbound1
        IPsecSaContextAddOutbound1
        IPsecSaContextGetSpi1
        IPsecSaContextDeleteById0
        FwpmProviderAdd0
        IPsecSaContextCreate1
        FwpmIPsecTunnelDeleteByKey0
        FwpmIPsecTunnelAdd2
        FwpmSubLayerDeleteByKey0
        FwpmProviderContextDeleteByKey0
        FwpmProviderContextAdd2
        FwpmProviderDeleteByKey0
        FwpmTransactionCommit0
        FwpmTransactionBegin0
        FwpmFilterDeleteById0
        FwpmFilterAdd0
        FwpmEngineClose0
        FwpmEngineOpen0
        FwpmSubLayerAdd0
        IPsecSaContextEnum1
        IPsecGetStatistics1
        IPsecSaContextCreateEnumHandle0
        FwpmNetEventUnsubscribe0
        FwpmCalloutDeleteByKey0
        FwpmTransactionAbort0
        FwpmCalloutAdd0
        FwpmFilterUnsubscribeChanges0

WS2_32.dll
        GetAddrInfoW
        FreeAddrInfoW
        WSASocketW
        WSAIoctl

RPCRT4.dll
        RpcServerUnregisterIf
        RpcServerUseProtseqIfW
        RpcServerRegisterIfEx
        RpcServerInqBindings
        RpcBindingVectorFree
        UuidIsNil
        UuidCreateNil
        UuidToStringW
        RpcStringFreeW
        UuidCreate
        NdrServerCall2
        RpcEpRegisterW

ncrypt.dll
        BCryptSignHash
        BCryptVerifySignature
```

BCryptDestroyKey
BCryptDestroySecret
BCryptGenerateSymmetricKey
NCryptDecrypt
BCryptCreateHash
NCryptSetProperty
BCryptSetProperty
BCryptHashData
BCryptImportKeyPair
NCryptFreeObject
NCryptCreatePersistedKey
BCryptDestroyHash
BCryptCloseAlgorithmProvider
NCryptGetProperty
BCryptFinishHash
BCryptExportKey
NCryptOpenKey
NCryptOpenStorageProvider
BCryptImportKey
BCryptGenerateKeyPair
BCryptSecretAgreement
BCryptDecrypt
BCryptFinalizeKeyPair
BCryptGetProperty
BCryptDeriveKey
BCryptOpenAlgorithmProvider
BCryptGenRandom
BCryptEncrypt
NCryptFinalizeKey

CRYPTXML.dll
CryptXmlOpenToEncode
CryptXmlCreateReference
CryptXmlSign
CryptXmlEncode
CryptXmlOpenToDecode
CryptXmlGetStatus
CryptXmlGetDocContext
CryptXmlVerifySignature
CryptXmlClose

WINHTTP.dll
WinHttpSetCredentials
WinHttpReceiveResponse
WinHttpOpen
WinHttpAddRequestHeaders
WinHttpQueryHeaders
WinHttpQueryDataAvailable
WinHttpCrackUrl
WinHttpConnect
WinHttpReadData
WinHttpSendRequest
WinHttpQueryAuthSchemes
WinHttpCloseHandle
WinHttpSetOption
WinHttpOpenRequest

ntdll.dll
RtlIpv4StringToAddressA
RtlIpv4StringToAddressW
RtlIpv6StringToAddressW
RtlIpv6StringToAddressA

WINTRUST.dll
WinVerifyTrust

USSL_Protocol.exe (64-bit)

CRYPT32.dll
CryptAcquireCertificatePrivateKey
CertVerifyCertificateChainPolicy
CryptBinaryToStringW
CryptMsgClose
CryptMsgGetParam
CryptQueryObject
CertCreateCertificateContext
CertGetCertificateChain
CertGetNameStringW
CertNameToStrW
CryptImportPublicKeyInfoEx2
CertVerifyTimeValidity
CertVerifyRevocation
CertDuplicateCertificateContext
CertFindCertificateInStore
CertCloseStore
CertOpenStore
CryptDecodeObjectEx
CryptStringToBinaryW
CertFreeCertificateChain
CertFreeCertificateContext

IPHLPAPI.DLL
FreeMibTable
CancelMibChangeNotify2
NotifyUnicastIpAddressChange
GetUnicastIpAddressTable
NotifyIpInterfaceChange
GetIpInterfaceEntry
GetIfEntry2
GetAdaptersAddresses
GetUnicastIpAddressEntry

KERNEL32.dll
InterlockedPushEntrySList
InterlockedFlushSList
GetFileSize
ReadFile
LocalAlloc
VerSetConditionMask
FindClose
FindFirstFileW

FindNextFileW
GetSystemInfo
VerifyVersionInfoW
GetTimeZoneInformation
WideCharToMultiByte
CreateTimerQueue
InitializeSListHead
WriteFile
GetSystemTimeAsFileTime
GetCurrentProcessId
QueryPerformanceCounter
GetThreadId
GetExitCodeThread
DebugBreak
LocalFree
MultiByteToWideChar
lstrcmpiW
MoveFileW
SizeofResource
LockResource
LoadResource
LoadLibraryExW
GetProcAddress
GetModuleHandleW
GetModuleFileNameW
FreeLibrary
FindResourceExW
GetCurrentThreadId
WaitForSingleObject
HeapSize
HeapReAlloc
HeapDestroy
CloseHandle
GetCommandLineW
SetEvent
GetOverlappedResult
WaitForSingleObjectEx
IsProcessorFeaturePresent
TerminateProcess
GetCurrentProcess
SetUnhandledExceptionFilter
UnhandledExceptionFilter
IsDebuggerPresent
DeviceIoControl
CreateFileW
Sleep
HeapAlloc
WaitForMultipleObjects
GetLocalTime
DeleteFileW
GetComputerNameExW
CompareFileTime
DeleteTimerQueueTimer
CreateTimerQueueTimer
CreateDirectoryW
FindResourceW

- DeleteTimerQueueEx
- DecodePointer
- CreateEventW
- ResetEvent
- LeaveCriticalSection
- RaiseException
- GetLastError
- InitializeCriticalSectionEx
- DeleteCriticalSection
- EnterCriticalSection
- GetProcessHeap
- HeapFree
- OutputDebugStringW

USER32.dll

- CharNextW
- PostThreadMessageW
- LoadStringW
- MessageBoxW
- wsprintfW

ADVAPI32.dll

- RegCloseKey
- CreateServiceW
- DeleteService
- RegCreateKeyExW
- RegDeleteKeyW
- RegDeleteValueW
- RegEnumKeyExW
- RegOpenKeyExW
- RegQueryInfoKeyW
- RegSetValueExW
- ChangeServiceConfig2W
- CloseServiceHandle
- OpenSCManagerW
- OpenServiceW
- RegisterServiceCtrlHandlerExW
- SetServiceStatus
- StartServiceCtrlDispatcherW
- DeregisterEventSource
- RegisterEventSourceW
- ReportEventW
- CryptReleaseContext
- ImpersonateLoggedOnUser
- RegQueryValueExW
- ControlService

SHELL32.dll

- SHGetKnownFolderPath

ole32.dll

- CoUninitialize
- CoInitializeEx
- CoAddRefServerProcess
- CoReleaseServerProcess
- CoCreateInstance

```
        CoTaskMemAlloc
        CoTaskMemRealloc
        CoFreeUnusedLibraries
        CoTaskMemFree

SHLWAPI.dll
        PathStripToRootW
        PathRemoveFileSpecW

fwpuclnt.dll
        IPsecSaContextDestroyEnumHandle0
        IPsecSaCreateEnumHandle0
        IPsecSaEnum1
        IPsecSaDestroyEnumHandle0
        FwpmProviderContextDeleteByKey0
        IPsecSaContextEnum1
        FwpmSubLayerDeleteByKey0
        FwpmTransactionBegin0
        FwpmNetEventUnsubscribe0
        FwpmCalloutDeleteByKey0
        FwpmSubLayerAdd0
        FwpmEngineOpen0
        FwpmEngineClose0
        FwpmTransactionCommit0
        FwpmFilterUnsubscribeChanges0
        IPsecSaContextCreateEnumHandle0
        IPsecGetStatistics1
        FwpmProviderContextAdd2
        FwpmFilterAdd0
        FwpmFilterDeleteById0
        FwpmProviderDeleteByKey0
        FwpmProviderAdd0
        FwpmCalloutAdd0
        FwpmTransactionAbort0
        IPsecSaContextAddOutbound1
        IPsecSaContextAddInbound1
        IPsecSaContextGetSpi1
        IPsecSaContextDeleteById0
        IPsecSaContextCreate1
        FwpmIPsecTunnelDeleteByKey0
        FwpmIPsecTunnelAdd2

WS2_32.dll
        InetNtopW
        FreeAddrInfoW
        GetAddrInfoW
        WSASocketW
        WSAIoctl

RPCRT4.dll
        RpcEpRegisterW
        UuidCreate
        RpcStringFreeW
        UuidToStringW
        UuidCreateNil
        UuidIsNil
```

RpcBindingVectorFree
RpcServerInqBindings
RpcServerRegisterIfEx
RpcServerUnregisterIf
RpcServerUseProtseqIfW
NdrServerCall2

ncrypt.dll
BCryptImportKey
BCryptSignHash
BCryptVerifySignature
BCryptDestroyKey
BCryptDestroySecret
BCryptSecretAgreement
BCryptOpenAlgorithmProvider
BCryptGenRandom
NCryptFreeObject
NCryptExportKey
NCryptImportKey
NCryptDecrypt
NCryptEncrypt
NCryptFinalizeKey
NCryptSetProperty
NCryptGetProperty
NCryptCreatePersistedKey
NCryptOpenKey
NCryptOpenStorageProvider
BCryptDestroyHash
BCryptFinishHash
BCryptHashData
BCryptCreateHash
BCryptFinalizeKeyPair
BCryptImportKeyPair
BCryptDeriveKey
BCryptExportKey
BCryptDecrypt
BCryptEncrypt
BCryptGenerateKeyPair
BCryptGenerateSymmetricKey
BCryptCloseAlgorithmProvider
BCryptGetProperty
BCryptSetProperty

CRYPTXML.dll
CryptXmlEncode
CryptXmlSign
CryptXmlCreateReference
CryptXmlOpenToEncode
CryptXmlGetStatus
CryptXmlGetDocContext
CryptXmlVerifySignature
CryptXmlOpenToDecode
CryptXmlClose

WINHTTP.dll
WinHttpQueryHeaders

WinHttpSetCredentials
WinHttpQueryAuthSchemes
WinHttpSendRequest
WinHttpCrackUrl
WinHttpCreateUrl
WinHttpOpen
WinHttpCloseHandle
WinHttpAddRequestHeaders
WinHttpConnect
WinHttpReadData
WinHttpQueryDataAvailable
WinHttpSetOption
WinHttpOpenRequest
WinHttpReceiveResponse

ntdll.dll

RtlLookupFunctionEntry
RtlCaptureContext
RtlIpv6StringToAddressA
RtlIpv4StringToAddressA
RtlIpv6StringToAddressW
RtlIpv4StringToAddressW
RtlVirtualUnwind

WINTRUST.dll

WinVerifyTrust

9.2 Linux Platform APIs

This Section identifies the Linux platform APIs used by the Unisys Stealth Linux Endpoint application.

charon (strongSwan daemon)

GLIBC

abort@GLIBC_2.2.5 (2)
asprintf@GLIBC_2.2.5 (2)
__errno_location@GLIBC_2.2.5 (3)
exit@GLIBC_2.2.5 (2)
fchown@GLIBC_2.2.5 (2)
fclose@GLIBC_2.2.5 (2)
fcntl@GLIBC_2.2.5 (3)
fflush@GLIBC_2.2.5 (2)
fileno@GLIBC_2.2.5 (2)
fopen@GLIBC_2.2.5 (2)
__fprintf_chk@GLIBC_2.3.4 (4)
fputc@GLIBC_2.2.5 (2)
fread@GLIBC_2.2.5 (2)
ftruncate@GLIBC_2.2.5 (2)
fwrite@GLIBC_2.2.5 (2)
getenv@GLIBC_2.2.5 (2)
getopt_long@GLIBC_2.2.5 (2)
getpid@GLIBC_2.2.5 (2)
kill@GLIBC_2.2.5 (2)
__libc_start_main@GLIBC_2.2.5 (2)

__printf_chk@GLIBC_2.3.4 (4)
pthread_sigmask@GLIBC_2.2.5 (3)
sigaction@GLIBC_2.2.5 (3)
sigaddset@GLIBC_2.2.5 (2)
sigemptyset@GLIBC_2.2.5 (2)
sigprocmask@GLIBC_2.2.5 (2)
sigwaitinfo@GLIBC_2.2.5 (2)
__stack_chk_fail@GLIBC_2.4 (5)
strtol@GLIBC_2.2.5 (2)
uname@GLIBC_2.2.5 (2)
unlink@GLIBC_2.2.5 (2)
__vfprintf_chk@GLIBC_2.3.4 (4)
__xstat@GLIBC_2.2.5 (2)

libcharon.so (strongSwan Charon library)

GLIBC
calloc@GLIBC_2.2.5 (2)
closelog@GLIBC_2.2.5 (2)
__ctype_b_loc@GLIBC_2.3 (8)
__cxa_finalize@GLIBC_2.2.5 (2)
__errno_location@GLIBC_2.2.5 (4)
fclose@GLIBC_2.2.5 (2)
ferror@GLIBC_2.2.5 (2)
fgets@GLIBC_2.2.5 (2)
fopen@GLIBC_2.2.5 (2)
__fprintf_chk@GLIBC_2.3.4 (3)
free@GLIBC_2.2.5 (2)
fwrite@GLIBC_2.2.5 (2)
getenv@GLIBC_2.2.5 (2)
getpid@GLIBC_2.2.5 (2)
gettimeofday@GLIBC_2.2.5 (2)
localtime_r@GLIBC_2.2.5 (2)
malloc@GLIBC_2.2.5 (2)
memcmp@GLIBC_2.2.5 (2)
memcpy@GLIBC_2.14 (7)
memset@GLIBC_2.2.5 (2)
openlog@GLIBC_2.2.5 (2)
pclose@GLIBC_2.2.5 (2)
popen@GLIBC_2.2.5 (2)
pow@GLIBC_2.2.5 (6)
random@GLIBC_2.2.5 (2)
realloc@GLIBC_2.2.5 (2)
setlinebuf@GLIBC_2.2.5 (2)
setlogmask@GLIBC_2.2.5 (2)
__snprintf_chk@GLIBC_2.3.4 (3)
snprintf@GLIBC_2.2.5 (2)
srandom@GLIBC_2.2.5 (2)
__stack_chk_fail@GLIBC_2.4 (5)
strchr@GLIBC_2.2.5 (2)
strcmp@GLIBC_2.2.5 (2)
__strdup@GLIBC_2.2.5 (2)
strftime@GLIBC_2.2.5 (2)
strlen@GLIBC_2.2.5 (2)

__strndup@GLIBC_2.2.5 (2)
__syslog_chk@GLIBC_2.4 (5)
time@GLIBC_2.2.5 (2)
usleep@GLIBC_2.2.5 (2)
__vsnprintf_chk@GLIBC_2.3.4 (3)

strongSwan Plugins (libstrongswan-*.so)

GLIBC
__asprintf_chk@GLIBC_2.8 (5)
asprintf@GLIBC_2.2.5 (2)
bind@GLIBC_2.2.5 (2)
calloc@GLIBC_2.2.5 (2)
close@GLIBC_2.2.5 (3)
__cxa_finalize@GLIBC_2.2.5 (4)
dlsym@GLIBC_2.2.5 (8)
__errno_location@GLIBC_2.2.5 (3)
fclose@GLIBC_2.2.5 (3)
fcntl@GLIBC_2.2.5 (2)
__fdelt_chk@GLIBC_2.15 (5)
fdopen@GLIBC_2.2.5 (3)
ferror@GLIBC_2.2.5 (3)
fflush@GLIBC_2.2.5 (2)
fgets@GLIBC_2.2.5 (3)
fmemopen@GLIBC_2.2.5 (3)
fopen@GLIBC_2.2.5 (3)
__fprintf_chk@GLIBC_2.3.4 (3)
fputc@GLIBC_2.2.5 (2)
fputs@GLIBC_2.2.5 (3)
fread@GLIBC_2.2.5 (3)
free@GLIBC_2.2.5 (4)
fwrite@GLIBC_2.2.5 (3)
getenv@GLIBC_2.2.5 (2)
getpid@GLIBC_2.2.5 (3)
getprotobyname@GLIBC_2.2.5 (3)
getservbyname@GLIBC_2.2.5 (3)
getsockname@GLIBC_2.2.5 (3)
getsockopt@GLIBC_2.2.5 (3)
globfree@GLIBC_2.2.5 (2)
glob@GLIBC_2.2.5 (2)
if_indextoname@GLIBC_2.2.5 (3)
if_nametoindex@GLIBC_2.2.5 (3)
ioctl@GLIBC_2.2.5 (3)
__isoc99_sscanf@GLIBC_2.7 (7)
mallinfo@GLIBC_2.2.5 (3)
malloc@GLIBC_2.2.5 (4)
memchr@GLIBC_2.2.5 (3)
memcmp@GLIBC_2.2.5 (4)
__memcpy_chk@GLIBC_2.3.4 (4)
memcpy@GLIBC_2.14 (8)
memset@GLIBC_2.2.5 (4)
open@GLIBC_2.2.5 (2)
poll@GLIBC_2.2.5 (2)
read@GLIBC_2.2.5 (2)

```
realloc@GLIBC_2.2.5 (3)
recvfrom@GLIBC_2.2.5 (3)
recv@GLIBC_2.2.5 (3)
recvmsg@GLIBC_2.2.5 (2)
select@GLIBC_2.2.5 (3)
sendmsg@GLIBC_2.2.5 (2)
sendto@GLIBC_2.2.5 (3)
setsockopt@GLIBC_2.2.5 (3)
SHA1_Init@libcrypto.so.10 (2)
SHA1_Update@libcrypto.so.10 (2)
sk_free@libcrypto.so.10 (2)
sk_num@libcrypto.so.10 (2)
sk_pop_free@libcrypto.so.10 (2)
sk_pop@libcrypto.so.10 (2)
sk_value@libcrypto.so.10 (2)
sleep@GLIBC_2.2.5 (3)
__snprintf_chk@GLIBC_2.3.4 (7)
snprintf@GLIBC_2.2.5 (2)
socket@GLIBC_2.2.5 (3)
__sprintf_chk@GLIBC_2.3.4 (3)
__stack_chk_fail@GLIBC_2.4 (6)
strcasecmp@GLIBC_2.2.5 (3)
strcmp@GLIBC_2.2.5 (3)
__strdup@GLIBC_2.2.5 (4)
strlen@GLIBC_2.2.5 (3)
strncasecmp@GLIBC_2.2.5 (3)
strncat@GLIBC_2.2.5 (3)
strncmp@GLIBC_2.2.5 (3)
strncpy@GLIBC_2.2.5 (3)
__strndup@GLIBC_2.2.5 (3)
strndup@GLIBC_2.2.5 (3)
strrchr@GLIBC_2.2.5 (3)
strstr@GLIBC_2.2.5 (3)
strtol@GLIBC_2.2.5 (3)
strtoul@GLIBC_2.2.5 (3)
strtoull@GLIBC_2.2.5 (3)
sysconf@GLIBC_2.2.5 (3)
time@GLIBC_2.2.5 (4)
uname@GLIBC_2.2.5 (3)
unlink@GLIBC_2.2.5 (3)
__vasprintf_chk@GLIBC_2.8 (5)
__vsnprintf_chk@GLIBC_2.3.4 (2)
__xstat@GLIBC_2.2.5 (3)
```

libcrypto and OPENSSL (OpenSSL)

```
ACCESS_DESCRIPTION_free@libcrypto.so.10 (2)
ASN1_BIT_STRING_free@libcrypto.so.10 (2)
ASN1_INTEGER_get@libcrypto.so.10 (2)
ASN1_OBJECT_free@libcrypto.so.10 (2)
ASN1_STRING_data@libcrypto.so.10 (2)
ASN1_STRING_free@libcrypto.so.10 (2)
ASN1_STRING_length@libcrypto.so.10 (2)
ASN1_STRING_type@libcrypto.so.10 (2)
AUTHORITY_KEYID_free@libcrypto.so.10 (2)
BASIC_CONSTRAINTS_free@libcrypto.so.10 (2)
```

BIO_free@libcrypto.so.10 (2)
BIO_new_mem_buf@libcrypto.so.10 (2)
BN_bin2bn@libcrypto.so.10 (2)
BN_bn2bin@libcrypto.so.10 (2)
BN_clear_free@libcrypto.so.10 (2)
BN_cmp@libcrypto.so.10 (2)
BN_copy@libcrypto.so.10 (2)
BN_CTX_end@libcrypto.so.10 (2)
BN_CTX_free@libcrypto.so.10 (2)
BN_CTX_get@libcrypto.so.10 (2)
BN_CTX_new@libcrypto.so.10 (2)
BN_CTX_start@libcrypto.so.10 (2)
BN_div@libcrypto.so.10 (2)
BN_free@libcrypto.so.10 (2)
BN_gcd@libcrypto.so.10 (2)
BN_mod_exp@libcrypto.so.10 (2)
BN_mod_inverse@libcrypto.so.10 (2)
BN_mod_sqr@libcrypto.so.10 (2)
BN_mul@libcrypto.so.10 (2)
BN_new@libcrypto.so.10 (2)
BN_num_bits@libcrypto.so.10 (2)
BN_pseudo_rand_range@libcrypto.so.10 (2)
BN_rshift@libcrypto.so.10 (2)
BN_set_word@libcrypto.so.10 (2)
BN_sub@libcrypto.so.10 (2)
BN_value_one@libcrypto.so.10 (2)
CMS_ContentInfo_free@libcrypto.so.10 (2)
CMS_get0_content@libcrypto.so.10 (2)
CMS_get0_RecipientInfos@libcrypto.so.10 (2)
CMS_get0_SignerInfos@libcrypto.so.10 (2)
CMS_get0_type@libcrypto.so.10 (2)
CMS_get1_certs@libcrypto.so.10 (2)
CMS_RecipientInfo_ktri_get0_algs@libcrypto.so.10 (2)
CMS_RecipientInfo_ktri_get0_signer_id@libcrypto.so.10 (2)
CMS_RecipientInfo_type@libcrypto.so.10 (2)
CMS_signed_get0_data_by_OBJ@libcrypto.so.10 (2)
CMS_signed_get_attr_count@libcrypto.so.10 (2)
CMS_signed_get_attr@libcrypto.so.10 (2)
CMS_SignerInfo_get0_algs@libcrypto.so.10 (2)
CMS_SignerInfo_get0_signer_id@libcrypto.so.10 (2)
CONF_modules_free@libcrypto.so.10 (2)
CRYPTO_cleanup_all_ex_data@libcrypto.so.10 (2)
CRYPTO_num_locks@libcrypto.so.10 (2)
CRYPTO_set_dynlock_create_callback@libcrypto.so.10 (2)
CRYPTO_set_dynlock_destroy_callback@libcrypto.so.10 (2)
CRYPTO_set_dynlock_lock_callback@libcrypto.so.10 (2)
CRYPTO_set_locking_callback@libcrypto.so.10 (2)
CRYPTO_THREADID_set_callback@libcrypto.so.10 (2)
CRYPTO_THREADID_set_numeric@libcrypto.so.10 (2)
d2i_AutoPrivateKey@libcrypto.so.10 (2)
d2i_CMS_bio@libcrypto.so.10 (2)
d2i_ECParameters@libcrypto.so.10 (2)
d2i_ECPrivateKey@libcrypto.so.10 (2)
d2i_EC_PUBKEY@libcrypto.so.10 (2)
d2i_PKCS12_bio@libcrypto.so.10 (2)
d2i_RSAPrivateKey@libcrypto.so.10 (2)

```
d2i_RSA_PUBKEY@libcrypto.so.10 (2)
d2i_RSAPublicKey@libcrypto.so.10 (2)
d2i_X509_CRL@libcrypto.so.10 (2)
d2i_X509@libcrypto.so.10 (2)
DH_compute_key@libcrypto.so.10 (2)
DH_free@libcrypto.so.10 (2)
DH_generate_key@libcrypto.so.10 (2)
DH_new@libcrypto.so.10 (2)
DH_size@libcrypto.so.10 (2)
DIST_POINT_free@libcrypto.so.10 (2)
ECDSA_do_sign@OPENSSL_1.0.1_EC (3)
ECDSA_do_verify@OPENSSL_1.0.1_EC (3)
ECDSA_SIG_free@OPENSSL_1.0.1_EC (3)
ECDSA_SIG_new@OPENSSL_1.0.1_EC (3)
ECDSA_sign@OPENSSL_1.0.1_EC (3)
ECDSA_size@OPENSSL_1.0.1_EC (3)
ECDSA_verify@OPENSSL_1.0.1_EC (3)
EC_GROUP_cmp@OPENSSL_1.0.1_EC (3)
EC_GROUP_free@OPENSSL_1.0.1_EC (3)
EC_GROUP_get_degree@OPENSSL_1.0.1_EC (3)
EC_GROUP_new_by_curve_name@OPENSSL_1.0.1_EC (3)
EC_KEY_check_key@OPENSSL_1.0.1_EC (3)
EC_KEY_free@OPENSSL_1.0.1_EC (3)
EC_KEY_generate_key@OPENSSL_1.0.1_EC (3)
EC_KEY_get0_group@OPENSSL_1.0.1_EC (3)
EC_KEY_get0_private_key@OPENSSL_1.0.1_EC (3)
EC_KEY_get0_public_key@OPENSSL_1.0.1_EC (3)
EC_KEY_new_by_curve_name@OPENSSL_1.0.1_EC (3)
EC_KEY_set_asn1_flag@OPENSSL_1.0.1_EC (3)
EC_KEY_set_conv_form@OPENSSL_1.0.1_EC (3)
EC_KEY_set_private_key@OPENSSL_1.0.1_EC (3)
EC_KEY_set_public_key@OPENSSL_1.0.1_EC (3)
EC_POINT_clear_free@OPENSSL_1.0.1_EC (3)
EC_POINT_free@OPENSSL_1.0.1_EC (3)
EC_POINT_get_affine_coordinates_GFp@OPENSSL_1.0.1_EC (3)
EC_POINT_is_on_curve@OPENSSL_1.0.1_EC (3)
EC_POINT_mul@OPENSSL_1.0.1_EC (3)
EC_POINT_new@OPENSSL_1.0.1_EC (3)
EC_POINT_set_affine_coordinates_GFp@OPENSSL_1.0.1_EC (3)
ENGINE_by_id@libcrypto.so.10 (2)
ENGINE_cleanup@libcrypto.so.10 (2)
ENGINE_ctrl_cmd_string@libcrypto.so.10 (2)
ENGINE_free@libcrypto.so.10 (2)
ENGINE_init@libcrypto.so.10 (2)
ENGINE_load_builtin_engines@libcrypto.so.10 (2)
ENGINE_load_private_key@libcrypto.so.10 (2)
ENGINE_register_all_complete@libcrypto.so.10 (2)
ERR_free_strings@libcrypto.so.10 (2)
ERR_remove_thread_state@libcrypto.so.10 (2)
EVP_aes_128_gcm@libcrypto.so.10 (2)
EVP_aes_192_gcm@libcrypto.so.10 (2)
EVP_aes_256_gcm@libcrypto.so.10 (2)
EVP_CIPHER_block_size@libcrypto.so.10 (2)
EVP_CIPHER_CTX_ctrl@libcrypto.so.10 (2)
EVP_CIPHER_CTX_free@libcrypto.so.10 (2)
EVP_CIPHER_CTX_new@libcrypto.so.10 (2)
```

```
EVP_CIPHER_CTX_set_key_length@libcrypto.so.10 (2)
EVP_CIPHER_CTX_set_padding@libcrypto.so.10 (2)
EVP_CipherFinal_ex@libcrypto.so.10 (2)
EVP_CipherInit_ex@libcrypto.so.10 (2)
EVP_CIPHER_iv_length@libcrypto.so.10 (2)
EVP_CipherUpdate@libcrypto.so.10 (2)
EVP_cleanup@libcrypto.so.10 (2)
EVP_des_ecb@libcrypto.so.10 (2)
EVP_DigestFinal_ex@libcrypto.so.10 (2)
EVP_DigestInit_ex@libcrypto.so.10 (2)
EVP_DigestSignFinal@libcrypto.so.10 (2)
EVP_DigestSignInit@libcrypto.so.10 (2)
EVP_DigestUpdate@libcrypto.so.10 (2)
EVP_DigestVerifyFinal@libcrypto.so.10 (2)
EVP_DigestVerifyInit@libcrypto.so.10 (2)
EVP_enc_null@libcrypto.so.10 (2)
EVP_get_cipherbyname@libcrypto.so.10 (2)
EVP_get_digestbyname@libcrypto.so.10 (2)
EVP_MD_CTX_create@libcrypto.so.10 (2)
EVP_MD_CTX_destroy@libcrypto.so.10 (2)
EVP_MD_size@libcrypto.so.10 (2)
EVP_PKEY_base_id@libcrypto.so.10 (2)
EVP_PKEY_CTX_ctrl@libcrypto.so.10 (2)
EVP_PKEY_free@libcrypto.so.10 (2)
EVP_PKEY_get1_EC_KEY@libcrypto.so.10 (2)
EVP_PKEY_get1_RSA@libcrypto.so.10 (2)
EVP_PKEY_new@libcrypto.so.10 (2)
EVP_PKEY_set1_RSA@libcrypto.so.10 (2)
FIPS_mode@libcrypto.so.10 (2)
FIPS_mode_set@libcrypto.so.10 (2)
GENERAL_NAME_free@libcrypto.so.10 (2)
HMAC_CTX_cleanup@libcrypto.so.10 (2)
HMAC_CTX_init@libcrypto.so.10 (2)
HMAC_Final@libcrypto.so.10 (2)
HMAC_Init_ex@libcrypto.so.10 (2)
HMAC_Update@libcrypto.so.10 (2)
i2d_ASN1_TYPE@libcrypto.so.10 (2)
i2d_ECPrivateKey@libcrypto.so.10 (2)
i2d_EC_PUBKEY@libcrypto.so.10 (2)
i2d_PrivateKey@libcrypto.so.10 (2)
i2d_RSAPrivateKey@libcrypto.so.10 (2)
i2d_RSA_PUBKEY@libcrypto.so.10 (2)
i2d_RSAPublicKey@libcrypto.so.10 (2)
i2d_X509_ALGOR@libcrypto.so.10 (2)
i2d_X509_ATTRIBUTE@libcrypto.so.10 (2)
i2d_X509_CINF@libcrypto.so.10 (2)
i2d_X509_CRL_INFO@libcrypto.so.10 (2)
i2d_X509@libcrypto.so.10 (2)
i2d_X509_NAME@libcrypto.so.10 (2)
i2d_X509_PUBKEY@libcrypto.so.10 (2)
i2o_ECPublicKey@libcrypto.so.10 (2)
IPAddressFamily_free@libcrypto.so.10 (2)
OBJ_cleanup@libcrypto.so.10 (2)
OBJ_nid2obj@libcrypto.so.10 (2)
OBJ_nid2sn@libcrypto.so.10 (2)
OBJ_obj2nid@libcrypto.so.10 (2)
```

OBJ_obj2txt@libcrypto.so.10 (2)
OPENSSL_add_all_algorithms_noconf@libcrypto.so.10 (2)
OPENSSL_config@libcrypto.so.10 (2)
PKCS12_free@libcrypto.so.10 (2)
PKCS12_parse@libcrypto.so.10 (2)
RAND_bytes@libcrypto.so.10 (2)
RAND_seed@libcrypto.so.10 (2)
RAND_status@libcrypto.so.10 (2)
RSA_check_key@libcrypto.so.10 (2)
RSA_free@libcrypto.so.10 (2)
RSA_generate_key_ex@libcrypto.so.10 (2)
RSA_new@libcrypto.so.10 (2)
RSA_private_decrypt@libcrypto.so.10 (2)
RSA_private_encrypt@libcrypto.so.10 (2)
RSA_public_decrypt@libcrypto.so.10 (2)
RSA_public_encrypt@libcrypto.so.10 (2)
RSA_size@libcrypto.so.10 (2)
v3_addr_get_afi@libcrypto.so.10 (2)
v3_addr_get_range@libcrypto.so.10 (2)
v3_addr_is_canonical@libcrypto.so.10 (2)
X509_ATTRIBUTE_count@libcrypto.so.10 (2)
X509_ATTRIBUTE_get0_type@libcrypto.so.10 (2)
X509_CRL_free@libcrypto.so.10 (2)
X509_EXTENSION_get_critical@libcrypto.so.10 (2)
X509_EXTENSION_get_data@libcrypto.so.10 (2)
X509_EXTENSION_get_object@libcrypto.so.10 (2)
X509_free@libcrypto.so.10 (2)
X509_get0_pubkey_bitstr@libcrypto.so.10 (2)
X509_get0_signature@libcrypto.so.10 (2)
X509_get_issuer_name@libcrypto.so.10 (2)
X509_get_serialNumber@libcrypto.so.10 (2)
X509_get_subject_name@libcrypto.so.10 (2)
X509_REVOKED_get_ext_d2i@libcrypto.so.10 (2)
X509V3_EXT_d2i@libcrypto.so.10 (2)

libcurl (cURL packaged with Stealth)
curl_easy_cleanup
curl_easy_getinfo
curl_easy_init
curl_easy_perform
curl_easy_setopt
curl_easy_strerror
curl_global_cleanup
curl_global_init
curl_slist_append
curl_slist_free_all
curl_version_info

libiptc
iptc_commit
iptc_delete_entry
iptc_free
iptc_init
iptc_insert_entry

iptc_strerror

libstrongswan.so

GLIBC

accept@GLIBC_2.2.5 (4)
access@GLIBC_2.2.5 (2)
__asprintf_chk@GLIBC_2.8 (11)
backtrace@GLIBC_2.2.5 (2)
backtrace_symbols@GLIBC_2.2.5 (2)
bind@GLIBC_2.2.5 (2)
bsearch@GLIBC_2.2.5 (2)
calloc@GLIBC_2.2.5 (2)
chown@GLIBC_2.2.5 (2)
clearerr@GLIBC_2.2.5 (2)
clock_gettime@GLIBC_2.17 (6)
closedir@GLIBC_2.2.5 (2)
close@GLIBC_2.2.5 (4)
connect@GLIBC_2.2.5 (4)
__ctype_b_loc@GLIBC_2.3 (16)
__cxa_finalize@GLIBC_2.2.5 (2)
dladdr@GLIBC_2.2.5 (10)
dlclose@GLIBC_2.2.5 (10)
dlerror@GLIBC_2.2.5 (10)
dlopen@GLIBC_2.2.5 (10)
dlsym@GLIBC_2.2.5 (10)
dup2@GLIBC_2.2.5 (2)
dup@GLIBC_2.2.5 (2)
__errno_location@GLIBC_2.2.5 (4)
execve@GLIBC_2.2.5 (2)
exit@GLIBC_2.2.5 (2)
fclose@GLIBC_2.2.5 (2)
fcntl@GLIBC_2.2.5 (4)
fdopen@GLIBC_2.2.5 (2)
ferror@GLIBC_2.2.5 (2)
fileno@GLIBC_2.2.5 (2)
fopen@GLIBC_2.2.5 (2)
fork@GLIBC_2.2.5 (4)
__fprintf_chk@GLIBC_2.3.4 (3)
fputc@GLIBC_2.2.5 (2)
fputs@GLIBC_2.2.5 (2)
fread@GLIBC_2.2.5 (2)
freeaddrinfo@GLIBC_2.2.5 (2)
free@GLIBC_2.2.5 (2)
fseek@GLIBC_2.2.5 (2)
ftell@GLIBC_2.2.5 (2)
fwrite@GLIBC_2.2.5 (2)
__fxstat@GLIBC_2.2.5 (2)
gai_strerror@GLIBC_2.2.5 (2)
__gcc_personality_v0@GCC_3.3.1 (13)
getaddrinfo@GLIBC_2.2.5 (2)
getegid@GLIBC_2.2.5 (2)
getenv@GLIBC_2.2.5 (2)
geteuid@GLIBC_2.2.5 (2)
getgrnam_r@GLIBC_2.2.5 (2)
getgroups@GLIBC_2.2.5 (2)

```
getpid@GLIBC_2.2.5 (2)
getprotobynumber@GLIBC_2.2.5 (2)
getpwnam_r@GLIBC_2.2.5 (2)
getpwuid_r@GLIBC_2.2.5 (2)
getservbyport@GLIBC_2.2.5 (2)
gettimeofday@GLIBC_2.2.5 (2)
globfree@GLIBC_2.2.5 (2)
glob@GLIBC_2.2.5 (2)
gmtime_r@GLIBC_2.2.5 (2)
inet_ntop@GLIBC_2.2.5 (2)
inet_pton@GLIBC_2.2.5 (2)
initgroups@GLIBC_2.2.5 (2)
ioctl@GLIBC_2.2.5 (2)
_IO_getc@GLIBC_2.2.5 (2)
isatty@GLIBC_2.2.5 (2)
__isoc99_sscanf@GLIBC_2.7 (8)
listen@GLIBC_2.2.5 (2)
localtime_r@GLIBC_2.2.5 (2)
malloc@GLIBC_2.2.5 (2)
memchr@GLIBC_2.2.5 (2)
memcmp@GLIBC_2.2.5 (2)
__memcpy_chk@GLIBC_2.3.4 (3)
memcpy@GLIBC_2.14 (12)
memmove@GLIBC_2.2.5 (2)
memrchr@GLIBC_2.2.5 (2)
memset@GLIBC_2.2.5 (2)
mkdir@GLIBC_2.2.5 (2)
mmap@GLIBC_2.2.5 (2)
munmap@GLIBC_2.2.5 (2)
__open_2@GLIBC_2.7 (8)
opendir@GLIBC_2.2.5 (2)
open@GLIBC_2.2.5 (4)
pclose@GLIBC_2.2.5 (2)
pipe@GLIBC_2.2.5 (2)
poll@GLIBC_2.2.5 (2)
popen@GLIBC_2.2.5 (2)
prctl@GLIBC_2.2.5 (2)
pthread_cancel@GLIBC_2.2.5 (4)
pthread_condattr_destroy@GLIBC_2.2.5 (4)
pthread_condattr_init@GLIBC_2.2.5 (4)
pthread_condattr_setclock@GLIBC_2.3.3 (14)
pthread_cond_broadcast@GLIBC_2.3.2 (5)
pthread_cond_destroy@GLIBC_2.3.2 (5)
pthread_cond_init@GLIBC_2.3.2 (5)
pthread_cond_signal@GLIBC_2.3.2 (5)
pthread_cond_timedwait@GLIBC_2.3.2 (5)
pthread_cond_wait@GLIBC_2.3.2 (5)
pthread_create@GLIBC_2.2.5 (4)
pthread_detach@GLIBC_2.2.5 (4)
pthread_exit@GLIBC_2.2.5 (4)
pthread_getspecific@GLIBC_2.2.5 (4)
pthread_join@GLIBC_2.2.5 (4)
pthread_key_create@GLIBC_2.2.5 (4)
pthread_key_delete@GLIBC_2.2.5 (4)
pthread_kill@GLIBC_2.2.5 (4)
pthread_mutex_destroy@GLIBC_2.2.5 (4)
```

```
pthread_mutex_init@GLIBC_2.2.5 (4)
pthread_mutex_lock@GLIBC_2.2.5 (4)
pthread_mutex_unlock@GLIBC_2.2.5 (4)
pthread_rwlock_destroy@GLIBC_2.2.5 (4)
pthread_rwlock_init@GLIBC_2.2.5 (4)
pthread_rwlock_rdlock@GLIBC_2.2.5 (4)
pthread_rwlock_trywrlock@GLIBC_2.2.5 (4)
pthread_rwlock_unlock@GLIBC_2.2.5 (4)
pthread_rwlock_wrlock@GLIBC_2.2.5 (4)
pthread_self@GLIBC_2.2.5 (4)
pthread_setcancelstate@GLIBC_2.2.5 (4)
pthread_setspecific@GLIBC_2.2.5 (4)
pthread_spin_destroy@GLIBC_2.2.5 (4)
pthread_spin_init@GLIBC_2.2.5 (4)
pthread_spin_lock@GLIBC_2.2.5 (4)
pthread_spin_unlock@GLIBC_2.2.5 (4)
pthread_testcancel@GLIBC_2.2.5 (4)
qsort_r@GLIBC_2.8 (11)
raise@GLIBC_2.2.5 (4)
random@GLIBC_2.2.5 (2)
__rawmemchr@GLIBC_2.2.5 (2)
readdir@GLIBC_2.2.5 (2)
read@GLIBC_2.2.5 (4)
realloc@GLIBC_2.2.5 (2)
recv@GLIBC_2.2.5 (4)
register_printf_specifier@GLIBC_2.10 (7)
rewind@GLIBC_2.2.5 (2)
sched_yield@GLIBC_2.2.5 (2)
send@GLIBC_2.2.5 (4)
setgid@GLIBC_2.2.5 (2)
setsockopt@GLIBC_2.2.5 (2)
setuid@GLIBC_2.2.5 (2)
sigaddset@GLIBC_2.2.5 (2)
sigemptyset@GLIBC_2.2.5 (2)
sigprocmask@GLIBC_2.2.5 (2)
sigwaitinfo@GLIBC_2.2.5 (2)
__snprintf_chk@GLIBC_2.3.4 (3)
snprintf@GLIBC_2.2.5 (2)
socket@GLIBC_2.2.5 (2)
srandom@GLIBC_2.2.5 (2)
__stack_chk_fail@GLIBC_2.4 (9)
stpcpy@GLIBC_2.2.5 (2)
strcasecmp@GLIBC_2.2.5 (2)
strchr@GLIBC_2.2.5 (2)
strcmp@GLIBC_2.2.5 (2)
strcpy@GLIBC_2.2.5 (2)
__strdup@GLIBC_2.2.5 (2)
strerror@GLIBC_2.2.5 (2)
strlen@GLIBC_2.2.5 (2)
strncasecmp@GLIBC_2.2.5 (2)
strncmp@GLIBC_2.2.5 (2)
__strncpy_chk@GLIBC_2.3.4 (3)
strncpy@GLIBC_2.2.5 (2)
__strndup@GLIBC_2.2.5 (2)
strrchr@GLIBC_2.2.5 (2)
strstr@GLIBC_2.2.5 (2)
```

strtod@GLIBC_2.2.5 (2)
strtol@GLIBC_2.2.5 (2)
strtoul@GLIBC_2.2.5 (2)
strtoull@GLIBC_2.2.5 (2)
syscall@GLIBC_2.2.5 (2)
sysconf@GLIBC_2.2.5 (2)
time@GLIBC_2.2.5 (2)
umask@GLIBC_2.2.5 (2)
unlink@GLIBC_2.2.5 (2)
_Unwind_Resume@GCC_3.0 (15)
__vasprintf_chk@GLIBC_2.8 (11)
__vfprintf_chk@GLIBC_2.3.4 (3)
__vsnprintf_chk@GLIBC_2.3.4 (3)
waitpid@GLIBC_2.2.5 (4)
write@GLIBC_2.2.5 (4)
__xpg_strerror_r@GLIBC_2.3.4 (3)
__xstat@GLIBC_2.2.5 (2)

libdavici.so (strongSwan Interface Library)

GLIBC

calloc@GLIBC_2.2.5 (3)
close@GLIBC_2.2.5 (3)
connect@GLIBC_2.2.5 (3)
__ctype_b_loc@GLIBC_2.3 (6)
__cxa_finalize@GLIBC_2.2.5 (3)
__errno_location@GLIBC_2.2.5 (3)
__fprintf_chk@GLIBC_2.3.4 (2)
free@GLIBC_2.2.5 (3)
malloc@GLIBC_2.2.5 (3)
memcpy@GLIBC_2.14 (5)
realloc@GLIBC_2.2.5 (3)
recv@GLIBC_2.2.5 (3)
send@GLIBC_2.2.5 (3)
__snprintf_chk@GLIBC_2.3.4 (2)
snprintf@GLIBC_2.2.5 (3)
socket@GLIBC_2.2.5 (3)
__stack_chk_fail@GLIBC_2.4 (4)
strcmp@GLIBC_2.2.5 (3)
__strdup@GLIBC_2.2.5 (3)
strlen@GLIBC_2.2.5 (3)
strncmp@GLIBC_2.2.5 (3)
__vsnprintf_chk@GLIBC_2.3.4 (2)

libvici.so (strongSwan Interface Library)

GLIBC

__cxa_finalize@GLIBC_2.2.5 (3)
__errno_location@GLIBC_2.2.5 (4)
__fprintf_chk@GLIBC_2.3.4 (2)
free@GLIBC_2.2.5 (3)

fwrite@GLIBC_2.2.5 (3)
malloc@GLIBC_2.2.5 (3)
__snprintf_chk@GLIBC_2.3.4 (2)
__stack_chk_fail@GLIBC_2.4 (5)
strcasecmp@GLIBC_2.2.5 (3)
strchr@GLIBC_2.2.5 (3)
strcmp@GLIBC_2.2.5 (3)
__strdup@GLIBC_2.2.5 (3)
strlen@GLIBC_2.2.5 (3)
strndup@GLIBC_2.2.5 (3)
strtol@GLIBC_2.2.5 (3)
__vsnprintf_chk@GLIBC_2.3.4 (2)

libcurl.so

GLIBC

accept@GLIBC_2.2.5 (4)
access@GLIBC_2.2.5 (4)
alarm@GLIBC_2.2.5 (4)
bind@GLIBC_2.2.5 (4)
calloc@GLIBC_2.2.5 (4)
clock_gettime@GLIBC_2.17 (6)
close@GLIBC_2.2.5 (4)
connect@GLIBC_2.2.5 (4)
__ctype_b_loc@GLIBC_2.3 (11)
__ctype_tolower_loc@GLIBC_2.3 (11)
__ctype_toupper_loc@GLIBC_2.3 (11)
__cxa_finalize@GLIBC_2.2.5 (4)
dup2@GLIBC_2.2.5 (4)
__errno_location@GLIBC_2.2.5 (4)
execl@GLIBC_2.2.5 (4)
exit@GLIBC_2.2.5 (4)
fclose@GLIBC_2.2.5 (4)
fcntl@GLIBC_2.2.5 (4)
__fdelt_chk@GLIBC_2.15 (14)
fflush@GLIBC_2.2.5 (4)
fgets@GLIBC_2.2.5 (4)
fileno@GLIBC_2.2.5 (4)
fopen@GLIBC_2.2.5 (4)
fork@GLIBC_2.2.5 (4)
fputc@GLIBC_2.2.5 (4)
fread@GLIBC_2.2.5 (4)
freeaddrinfo@GLIBC_2.2.5 (4)
free@GLIBC_2.2.5 (4)
freeifaddrs@GLIBC_2.3 (11)
fseek@GLIBC_2.2.5 (4)
ftell@GLIBC_2.2.5 (4)
fwrite@GLIBC_2.2.5 (4)
__fxstat@GLIBC_2.2.5 (4)
getaddrinfo@GLIBC_2.2.5 (4)
getenv@GLIBC_2.2.5 (4)
geteuid@GLIBC_2.2.5 (4)
gethostname@GLIBC_2.2.5 (4)
getifaddrs@GLIBC_2.3 (11)
getpeername@GLIBC_2.2.5 (4)
getpid@GLIBC_2.2.5 (4)

```
getpwuid_r@GLIBC_2.2.5 (4)
getsockname@GLIBC_2.2.5 (4)
getsockopt@GLIBC_2.2.5 (4)
gettimeofday@GLIBC_2.2.5 (4)
gmtime_r@GLIBC_2.2.5 (4)
if_nametoindex@GLIBC_2.2.5 (4)
inet_ntop@GLIBC_2.2.5 (4)
inet_pton@GLIBC_2.2.5 (4)
__isoc99_sscanf@GLIBC_2.7 (7)
kill@GLIBC_2.2.5 (4)
listen@GLIBC_2.2.5 (4)
__longjmp_chk@GLIBC_2.11 (5)
lseek@GLIBC_2.2.5 (4)
malloc@GLIBC_2.2.5 (4)
memchr@GLIBC_2.2.5 (4)
memcmp@GLIBC_2.2.5 (4)
__memcpy_chk@GLIBC_2.3.4 (9)
memcpy@GLIBC_2.14 (13)
memmove@GLIBC_2.2.5 (4)
memset@GLIBC_2.2.5 (4)
open@GLIBC_2.2.5 (4)
__poll_chk@GLIBC_2.16 (8)
poll@GLIBC_2.2.5 (4)
qsort@GLIBC_2.2.5 (4)
__rawmemchr@GLIBC_2.2.5 (4)
read@GLIBC_2.2.5 (4)
realloc@GLIBC_2.2.5 (4)
recvfrom@GLIBC_2.2.5 (4)
recv@GLIBC_2.2.5 (4)
send@GLIBC_2.2.5 (4)
sendto@GLIBC_2.2.5 (4)
setsockopt@GLIBC_2.2.5 (4)
sigaction@GLIBC_2.2.5 (4)
__sigsetjmp@GLIBC_2.2.5 (4)
socket@GLIBC_2.2.5 (4)
socketpair@GLIBC_2.2.5 (4)
__sprintf_chk@GLIBC_2.3.4 (9)
__stack_chk_fail@GLIBC_2.4 (12)
strcasecmp@GLIBC_2.2.5 (4)
strchr@GLIBC_2.2.5 (4)
strcmp@GLIBC_2.2.5 (4)
__strcpy_chk@GLIBC_2.3.4 (9)
strcpy@GLIBC_2.2.5 (4)
strdup@GLIBC_2.2.5 (4)
strerror@GLIBC_2.2.5 (4)
strlen@GLIBC_2.2.5 (4)
strncasecmp@GLIBC_2.2.5 (4)
strncmp@GLIBC_2.2.5 (4)
strncpy@GLIBC_2.2.5 (4)
strrchr@GLIBC_2.2.5 (4)
strstr@GLIBC_2.2.5 (4)
__strtok_r@GLIBC_2.2.5 (4)
strtol@GLIBC_2.2.5 (4)
strtoul@GLIBC_2.2.5 (4)
time@GLIBC_2.2.5 (4)
waitpid@GLIBC_2.2.5 (4)
```

write@GLIBC_2.2.5 (4)
__xpg_basename@GLIBC_2.2.5 (4)
__xpg_strerror_r@GLIBC_2.3.4 (9)
__xstat@GLIBC_2.2.5 (4)

libcrypto (OpenSSL)
ASN1_INTEGER_get@libcrypto.so.10 (2)
ASN1_STRING_data@libcrypto.so.10 (2)
ASN1_STRING_length@libcrypto.so.10 (2)
ASN1_STRING_print@libcrypto.so.10 (2)
ASN1_STRING_to_UTF8@libcrypto.so.10 (2)
ASN1_STRING_type@libcrypto.so.10 (2)
ASN1_TIME_print@libcrypto.so.10 (2)
BIO_ctrl@libcrypto.so.10 (2)
BIO_free@libcrypto.so.10 (2)
BIO_new@libcrypto.so.10 (2)
BIO_printf@libcrypto.so.10 (2)
BIO_puts@libcrypto.so.10 (2)
BIO_s_mem@libcrypto.so.10 (2)
BN_num_bits@libcrypto.so.10 (2)
BN_print@libcrypto.so.10 (2)
CONF_modules_free@libcrypto.so.10 (2)
CONF_modules_load_file@libcrypto.so.10 (2)
CRYPTO_cleanup_all_ex_data@libcrypto.so.10 (2)
CRYPTO_free@libcrypto.so.10 (2)
CRYPTO_malloc@libcrypto.so.10 (2)
d2i_OCSP_RESPONSE@libcrypto.so.10 (2)
d2i_PKCS12_fp@libcrypto.so.10 (2)
DES_ecb_encrypt@libcrypto.so.10 (2)
DES_set_key@libcrypto.so.10 (2)
DES_set_odd_parity@libcrypto.so.10 (2)
ENGINE_by_id@libcrypto.so.10 (2)
ENGINE_cleanup@libcrypto.so.10 (2)
ENGINE_ctrl_cmd@libcrypto.so.10 (2)
ENGINE_ctrl@libcrypto.so.10 (2)
ENGINE_finish@libcrypto.so.10 (2)
ENGINE_free@libcrypto.so.10 (2)
ENGINE_get_first@libcrypto.so.10 (2)
ENGINE_get_id@libcrypto.so.10 (2)
ENGINE_get_next@libcrypto.so.10 (2)
ENGINE_init@libcrypto.so.10 (2)
ENGINE_load_builtin_engines@libcrypto.so.10 (2)
ENGINE_load_private_key@libcrypto.so.10 (2)
ENGINE_set_default@libcrypto.so.10 (2)
ERR_clear_error@libcrypto.so.10 (2)
ERR_error_string@libcrypto.so.10 (2)
ERR_error_string_n@libcrypto.so.10 (2)
ERR_free_strings@libcrypto.so.10 (2)
ERR_get_error@libcrypto.so.10 (2)
ERR_peek_error@libcrypto.so.10 (2)
ERR_remove_thread_state@libcrypto.so.10 (2)
EVP_cleanup@libcrypto.so.10 (2)
EVP_PKEY_copy_parameters@libcrypto.so.10 (2)
EVP_PKEY_free@libcrypto.so.10 (2)
GENERAL_NAMES_free@libcrypto.so.10 (2)

```
i2a_ASN1_OBJECT@libcrypto.so.10 (2)
i2d_X509_PUBKEY@libcrypto.so.10 (2)
i2t_ASN1_OBJECT@libcrypto.so.10 (2)
MD4_Final@libcrypto.so.10 (2)
MD4_Init@libcrypto.so.10 (2)
MD4_Update@libcrypto.so.10 (2)
MD5_Final@libcrypto.so.10 (2)
MD5_Init@libcrypto.so.10 (2)
MD5_Update@libcrypto.so.10 (2)
OCSP_BASICRESP_free@libcrypto.so.10 (2)
OCSP_basic_verify@libcrypto.so.10 (2)
OCSP_cert_status_str@libcrypto.so.10 (2)
OCSP_check_validity@libcrypto.so.10 (2)
OCSP_crl_reason_str@libcrypto.so.10 (2)
OCSP_resp_count@libcrypto.so.10 (2)
OCSP_resp_get0@libcrypto.so.10 (2)
OCSP_RESPONSE_free@libcrypto.so.10 (2)
OCSP_response_get1_basic@libcrypto.so.10 (2)
OCSP_response_status@libcrypto.so.10 (2)
OCSP_response_status_str@libcrypto.so.10 (2)
OCSP_single_get0_status@libcrypto.so.10 (2)
OPENSSL_add_all_algorithms_noconf@libcrypto.so.10 (2)
OPENSSL_load_builtin_modules@libcrypto.so.10 (2)
PEM_read_RSA_PUBKEY@libcrypto.so.10 (3)
PEM_read_X509@libcrypto.so.10 (2)
PEM_write_bio_X509@libcrypto.so.10 (2)
PKCS12_free@libcrypto.so.10 (2)
PKCS12_parse@libcrypto.so.10 (2)
PKCS12_PBE_add@libcrypto.so.10 (2)
RAND_add@libcrypto.so.10 (2)
RAND_bytes@libcrypto.so.10 (2)
RAND_egd@libcrypto.so.10 (2)
RAND_file_name@libcrypto.so.10 (2)
RAND_load_file@libcrypto.so.10 (2)
RAND_status@libcrypto.so.10 (2)
SHA256_Final@libcrypto.so.10 (2)
SHA256_Init@libcrypto.so.10 (2)
SHA256_Update@libcrypto.so.10 (2)
sk_num@libcrypto.so.10 (2)
sk_pop_free@libcrypto.so.10 (2)
sk_pop@libcrypto.so.10 (2)
sk_value@libcrypto.so.10 (2)
SSL_CIPHER_get_name@libssl.so.10 (3)
SSL_connect@libssl.so.10 (3)
SSL_ctrl@libssl.so.10 (3)
SSL_CTX_add_client_CA@libssl.so.10 (3)
SSL_CTX_check_private_key@libssl.so.10 (3)
SSL_CTX_ctrl@libssl.so.10 (3)
SSL_CTX_free@libssl.so.10 (3)
SSL_CTX_get_cert_store@libssl.so.10 (3)
SSL_CTX_load_verify_locations@libssl.so.10 (3)
SSL_CTX_new@libssl.so.10 (3)
SSL_CTX_set_alpn_protos@libssl.so.10 (3)
SSL_CTX_set_cipher_list@libssl.so.10 (3)
SSL_CTX_set_default_passwd_cb@libssl.so.10 (3)
SSL_CTX_set_default_passwd_cb_userdata@libssl.so.10 (3)
```

```
SSL_CTX_set_msg_callback@libssl.so.10 (3)
SSL_CTX_set_next_proto_select_cb@libssl.so.10 (3)
SSL_CTX_set_verify@libssl.so.10 (3)
SSL_CTX_use_certificate_chain_file@libssl.so.10 (3)
SSL_CTX_use_certificate_file@libssl.so.10 (3)
SSL_CTX_use_certificate@libssl.so.10 (3)
SSL_CTX_use_PrivateKey_file@libssl.so.10 (3)
SSL_CTX_use_PrivateKey@libssl.so.10 (3)
SSLeay@OPENSSL_1.0.2 (10)
SSL_free@libssl.so.10 (3)
SSL_get0_alpn_selected@libssl.so.10 (3)
SSL_get1_session@libssl.so.10 (3)
SSL_get_certificate@libssl.so.10 (3)
SSL_get_current_cipher@libssl.so.10 (3)
SSL_get_error@libssl.so.10 (3)
SSL_get_peer_cert_chain@libssl.so.10 (3)
SSL_get_peer_certificate@libssl.so.10 (3)
SSL_get_privatekey@libssl.so.10 (3)
SSL_get_shutdown@libssl.so.10 (3)
SSL_get_verify_result@libssl.so.10 (3)
SSL_library_init@libssl.so.10 (3)
SSL_load_error_strings@libssl.so.10 (3)
SSL_new@libssl.so.10 (3)
SSL_peek@libssl.so.10 (3)
SSL_pending@libssl.so.10 (3)
SSL_read@libssl.so.10 (3)
SSL_SESSION_free@libssl.so.10 (3)
SSL_set_connect_state@libssl.so.10 (3)
SSL_set_fd@libssl.so.10 (3)
SSL_set_session@libssl.so.10 (3)
SSL_shutdown@libssl.so.10 (3)
SSLv23_client_method@libssl.so.10 (3)
SSLv3_client_method@libssl.so.10 (3)
SSL_version@libssl.so.10 (3)
SSL_write@libssl.so.10 (3)
UI_create_method@libcrypto.so.10 (2)
UI_destroy_method@libcrypto.so.10 (2)
UI_get0_user_data@libcrypto.so.10 (2)
UI_get_input_flags@libcrypto.so.10 (2)
UI_get_string_type@libcrypto.so.10 (2)
UI_method_get_closer@libcrypto.so.10 (2)
UI_method_get_opener@libcrypto.so.10 (2)
UI_method_get_reader@libcrypto.so.10 (2)
UI_method_get_writer@libcrypto.so.10 (2)
UI_method_set_closer@libcrypto.so.10 (2)
UI_method_set_opener@libcrypto.so.10 (2)
UI_method_set_reader@libcrypto.so.10 (2)
UI_method_set_writer@libcrypto.so.10 (2)
UI_OpenSSL@libcrypto.so.10 (2)
UI_set_result@libcrypto.so.10 (2)
X509_check_issued@libcrypto.so.10 (2)
X509_EXTENSION_get_data@libcrypto.so.10 (2)
X509_EXTENSION_get_object@libcrypto.so.10 (2)
X509_free@libcrypto.so.10 (2)
X509_get_ext_d2i@libcrypto.so.10 (2)
X509_get_issuer_name@libcrypto.so.10 (2)
```

X509_get_pubkey@libcrypto.so.10 (2)
X509_get_serialNumber@libcrypto.so.10 (2)
X509_get_subject_name@libcrypto.so.10 (2)
X509_load_crl_file@libcrypto.so.10 (2)
X509_LOOKUP_file@libcrypto.so.10 (2)
X509_NAME_ENTRY_get_data@libcrypto.so.10 (2)
X509_NAME_get_entry@libcrypto.so.10 (2)
X509_NAME_get_index_by_NID@libcrypto.so.10 (2)
X509_NAME_print_ex@libcrypto.so.10 (2)
X509_STORE_add_lookup@libcrypto.so.10 (2)
X509_STORE_set_flags@libcrypto.so.10 (2)
X509V3_EXT_print@libcrypto.so.10 (2)
X509_verify_cert_error_string@libcrypto.so.10 (2)

libnetfilter_conntrack.so

GLIBC
__assert_fail@GLIBC_2.2.5 (2)
calloc@GLIBC_2.2.5 (2)
ctime@GLIBC_2.2.5 (2)
__errno_location@GLIBC_2.2.5 (2)
fclose@GLIBC_2.2.5 (2)
fgets@GLIBC_2.2.5 (2)
fopen@GLIBC_2.2.5 (2)
free@GLIBC_2.2.5 (2)
inet_ntoa@GLIBC_2.2.5 (2)
inet_ntop@GLIBC_2.2.5 (2)
localtime_r@GLIBC_2.2.5 (2)
malloc@GLIBC_2.2.5 (2)
memcmp@GLIBC_2.2.5 (2)
memcpy@GLIBC_2.14 (7)
memset@GLIBC_2.2.5 (2)
__rawmemchr@GLIBC_2.2.5 (2)
setsockopt@GLIBC_2.2.5 (2)
__snprintf_chk@GLIBC_2.3.4 (3)
snprintf@GLIBC_2.2.5 (2)
__stack_chk_fail@GLIBC_2.4 (5)
strchr@GLIBC_2.2.5 (2)
strcmp@GLIBC_2.2.5 (2)
__strcpy_chk@GLIBC_2.3.4 (3)
__strdup@GLIBC_2.2.5 (2)
strlen@GLIBC_2.2.5 (2)
__strncpy_chk@GLIBC_2.3.4 (3)
strncpy@GLIBC_2.2.5 (2)
strtoul@GLIBC_2.2.5 (2)
time@GLIBC_2.2.5 (2)

libmnl (netfilter)
mnl_attr_get_payload_len@LIBMNL_1.0 (4)
mnl_attr_get_payload@LIBMNL_1.0 (4)
mnl_attr_get_str@LIBMNL_1.0 (4)
mnl_attr_get_type@LIBMNL_1.0 (4)
mnl_attr_get_u16@LIBMNL_1.0 (4)
mnl_attr_get_u32@LIBMNL_1.0 (4)
mnl_attr_get_u64@LIBMNL_1.0 (4)

mnl_attr_get_u8@LIBMNL_1.0 (4)
mnl_attr_nest_cancel@LIBMNL_1.0 (4)
mnl_attr_nest_end@LIBMNL_1.0 (4)
mnl_attr_nest_start@LIBMNL_1.0 (4)
mnl_attr_parse@LIBMNL_1.0 (4)
mnl_attr_parse_nested@LIBMNL_1.0 (4)
mnl_attr_parse_payload@LIBMNL_1.1 (6)
mnl_attr_put@LIBMNL_1.0 (4)
mnl_attr_put_strz@LIBMNL_1.0 (4)
mnl_attr_put_u16@LIBMNL_1.0 (4)
mnl_attr_put_u32@LIBMNL_1.0 (4)
mnl_attr_put_u64@LIBMNL_1.0 (4)
mnl_attr_put_u8@LIBMNL_1.0 (4)
mnl_attr_type_valid@LIBMNL_1.0 (4)
mnl_attr_validate2@LIBMNL_1.0 (4)
mnl_attr_validate@LIBMNL_1.0 (4)
mnl_nlmsg_get_payload_len@LIBMNL_1.0 (4)
mnl_nlmsg_get_payload@LIBMNL_1.0 (4)

libnfnetlink (netfilter)
nfnl_addattr16
nfnl_addattr32
nfnl_addattr_l
nfnl_callback_register
nfnl_callback_unregister
nfnl_catch
nfnl_close
nfnl_fd
nfnl_fill_hdr
nfnl_open
nfnl_parse_attr
nfnl_query
nfnl_send
nfnl_subsys_close
nfnl_subsys_open

libxmlsec.so

GLIBC
__ctype_b_loc@GLIBC_2.3 (5)
__cxa_finalize@GLIBC_2.2.5 (3)
__errno_location@GLIBC_2.2.5 (3)
fclose@GLIBC_2.2.5 (3)
fopen@GLIBC_2.2.5 (3)
__fprintf_chk@GLIBC_2.3.4 (4)
fputc@GLIBC_2.2.5 (3)
fread@GLIBC_2.2.5 (3)
fwrite@GLIBC_2.2.5 (3)
memcmp@GLIBC_2.2.5 (3)
memcpy@GLIBC_2.14 (10)
memmove@GLIBC_2.2.5 (3)
memset@GLIBC_2.2.5 (3)
rand@GLIBC_2.2.5 (3)
__sprintf_chk@GLIBC_2.3.4 (4)
srand@GLIBC_2.2.5 (3)

```
__stack_chk_fail@GLIBC_2.4 (6)
time@GLIBC_2.2.5 (3)
```

Libxml2

```
inputPush@LIBXML2_2.4.30 (2)
valuePush@LIBXML2_2.4.30 (2)
xmlAddChild@LIBXML2_2.4.30 (2)
xmlAddID@LIBXML2_2.4.30 (2)
xmlAddNextSibling@LIBXML2_2.4.30 (2)
xmlAddPrevSibling@LIBXML2_2.4.30 (2)
xmlC14NExecute@LIBXML2_2.4.30 (2)
xmlCreateFileParserCtxt@LIBXML2_2.4.30 (2)
xmlCreateMemoryParserCtxt@LIBXML2_2.4.30 (2)
xmlCreatePushParserCtxt@LIBXML2_2.4.30 (2)
xmlDocGetRootElement@LIBXML2_2.4.30 (2)
xmlDocSetRootElement@LIBXML2_2.4.30 (2)
xmlEncodeSpecialChars@LIBXML2_2.4.30 (2)
xmlFileClose@LIBXML2_2.4.30 (2)
xmlFileMatch@LIBXML2_2.4.30 (2)
xmlFileOpen@LIBXML2_2.4.30 (2)
xmlFileRead@LIBXML2_2.4.30 (2)
xmlFreeDoc@LIBXML2_2.4.30 (2)
xmlFreeInputStream@LIBXML2_2.4.30 (2)
xmlFreeNode@LIBXML2_2.4.30 (2)
xmlFreeNodeList@LIBXML2_2.4.30 (2)
xmlFreeParserCtxt@LIBXML2_2.4.30 (2)
xmlFreeParserInputBuffer@LIBXML2_2.4.30 (2)
__xmlGenericError
__xmlGenericErrorContext
xmlGetID@LIBXML2_2.4.30 (2)
xmlGetProp@LIBXML2_2.4.30 (2)
xmlInitParser@LIBXML2_2.4.30 (2)
xmlIOFTPClose@LIBXML2_2.4.30 (2)
xmlIOFTPMatch@LIBXML2_2.4.30 (2)
xmlIOFTPOpen@LIBXML2_2.4.30 (2)
xmlIOFTPRead@LIBXML2_2.4.30 (2)
xmlIOHTTPClose@LIBXML2_2.4.30 (2)
xmlIOHTTPMatch@LIBXML2_2.4.30 (2)
xmlIOHTTPOpen@LIBXML2_2.4.30 (2)
xmlIOHTTPRead@LIBXML2_2.4.30 (2)
xmlLinkGetData@LIBXML2_2.4.30 (2)
xmlListCreate@LIBXML2_2.4.30 (2)
xmlListDelete@LIBXML2_2.4.30 (2)
xmlListEmpty@LIBXML2_2.4.30 (2)
xmlListFront@LIBXML2_2.4.30 (2)
xmlListInsert@LIBXML2_2.4.30 (2)
xmlListPopFront@LIBXML2_2.4.30 (2)
xmlListSort@LIBXML2_2.4.30 (2)
xmlNanoFTPCleanup@LIBXML2_2.4.30 (2)
xmlNanoFTPInit@LIBXML2_2.4.30 (2)
xmlNanoHTTPCleanup@LIBXML2_2.4.30 (2)
xmlNanoHTTPInit@LIBXML2_2.4.30 (2)
xmlNewChild@LIBXML2_2.4.30 (2)
xmlNewDoc@LIBXML2_2.4.30 (2)
xmlNewDocNode@LIBXML2_2.4.30 (2)
xmlNewIOInputStream@LIBXML2_2.4.30 (2)
```

xmlNewNode@LIBXML2_2.4.30 (2)
xmlNewNs@LIBXML2_2.4.30 (2)
xmlNewParserCtxt@LIBXML2_2.4.30 (2)
xmlNewText@LIBXML2_2.4.30 (2)
xmlNodeAddContent@LIBXML2_2.4.30 (2)
xmlNodeDumpOutput@LIBXML2_2.4.30 (2)
xmlNodeGetContent@LIBXML2_2.4.30 (2)
xmlNodeListGetString@LIBXML2_2.4.30 (2)
xmlNodeSetContentLen@LIBXML2_2.4.30 (2)
xmlNodeSetContent@LIBXML2_2.4.30 (2)
xmlNodeSetLang@LIBXML2_2.4.30 (2)
xmlOutputBufferClose@LIBXML2_2.4.30 (2)
xmlOutputBufferCreateIO@LIBXML2_2.4.30 (2)
xmlOutputBufferWriteString@LIBXML2_2.4.30 (2)
xmlParseChunk@LIBXML2_2.4.30 (2)
xmlParseDocument@LIBXML2_2.4.30 (2)
xmlParseFile@LIBXML2_2.4.30 (2)
xmlParseInNodeContext@LIBXML2_2.6.12 (9)
xmlParseMemory@LIBXML2_2.4.30 (2)
xmlParserGetDirectory@LIBXML2_2.4.30 (2)
xmlParserInputBufferCreateIO@LIBXML2_2.4.30 (2)
xmlRecoverMemory@LIBXML2_2.4.30 (2)
xmlReplaceNode@LIBXML2_2.4.30 (2)
xmlSaveFormatFile@LIBXML2_2.4.30 (2)
xmlSearchNsByHref@LIBXML2_2.4.30 (2)
xmlSearchNs@LIBXML2_2.4.30 (2)
xmlSetNs@LIBXML2_2.4.30 (2)
xmlSetProp@LIBXML2_2.4.30 (2)
xmlSetTreeDoc@LIBXML2_2.4.30 (2)
xmlStrchr@LIBXML2_2.4.30 (2)
xmlStrcmp@LIBXML2_2.4.30 (2)
xmlStrdup@LIBXML2_2.4.30 (2)
xmlStrEqual@LIBXML2_2.4.30 (2)
xmlStrlen@LIBXML2_2.4.30 (2)
xmlStrncmp@LIBXML2_2.4.30 (2)
xmlStrndup@LIBXML2_2.4.30 (2)
xmlStrPrintf@LIBXML2_2.6.0 (8)
xmlStrVPrintf@LIBXML2_2.6.2 (7)
xmlUnlinkNode@LIBXML2_2.4.30 (2)
xmlURIUnescapeString@LIBXML2_2.4.30 (2)
xmlXPathErr@LIBXML2_2.6.0 (8)
xmlXPathEvalExpression@LIBXML2_2.4.30 (2)
xmlXPathFreeContext@LIBXML2_2.4.30 (2)
xmlXPathFreeNodeSet@LIBXML2_2.4.30 (2)
xmlXPathFreeObject@LIBXML2_2.4.30 (2)
xmlXPathNewContext@LIBXML2_2.4.30 (2)
xmlXPathNewNodeSet@LIBXML2_2.4.30 (2)
xmlXPathNodeSetAdd@LIBXML2_2.4.30 (2)
xmlXPathNodeSetContains@LIBXML2_2.4.30 (2)
xmlXPathNodeSetCreate@LIBXML2_2.4.30 (2)
xmlXPathNsLookup@LIBXML2_2.4.30 (2)
xmlXPathRegisterFunc@LIBXML2_2.4.30 (2)
xmlXPathRegisterNs@LIBXML2_2.4.30 (2)
xmlXPtrEval@LIBXML2_2.4.30 (2)
xmlXPtrNewContext@LIBXML2_2.4.30 (2)

libxmlsec1-openssl.so

OpenSSL

AES_decrypt@libcrypto.so.10 (2)
AES_encrypt@libcrypto.so.10 (2)
AES_set_decrypt_key@libcrypto.so.10 (2)
AES_set_encrypt_key@libcrypto.so.10 (2)
ASN1_INTEGER_cmp@libcrypto.so.10 (2)
ASN1_INTEGER_free@libcrypto.so.10 (2)
ASN1_INTEGER_to_BN@libcrypto.so.10 (2)
ASN1_OCTET_STRING_free@libcrypto.so.10 (2)
ASN1_STRING_data@libcrypto.so.10 (2)
ASN1_STRING_length@libcrypto.so.10 (2)
ASN1_TIME_check@libcrypto.so.10 (2)
BIO_ctrl@libcrypto.so.10 (2)
BIO_free_all@libcrypto.so.10 (2)
BIO_free@libcrypto.so.10 (2)
BIO_new_file@libcrypto.so.10 (2)
BIO_new@libcrypto.so.10 (2)
BIO_new_mem_buf@libcrypto.so.10 (2)
BIO_read@libcrypto.so.10 (2)
BIO_s_mem@libcrypto.so.10 (2)
BIO_write@libcrypto.so.10 (2)
BN_bin2bn@libcrypto.so.10 (2)
BN_bn2bin@libcrypto.so.10 (2)
BN_bn2dec@libcrypto.so.10 (2)
BN_clear_free@libcrypto.so.10 (2)
BN_dec2bn@libcrypto.so.10 (2)
BN_free@libcrypto.so.10 (2)
BN_new@libcrypto.so.10 (2)
BN_num_bits@libcrypto.so.10 (2)
BN_print_fp@libcrypto.so.10 (2)
BN_set_word@libcrypto.so.10 (2)
BN_to_ASN1_INTEGER@libcrypto.so.10 (2)
CONF_modules_unload@libcrypto.so.10 (2)
CRYPTO_add_lock@libcrypto.so.10 (2)
CRYPTO_cleanup_all_ex_data@libcrypto.so.10 (2)
CRYPTO_free@libcrypto.so.10 (2)
d2i_PKCS12_bio@libcrypto.so.10 (2)
d2i_PKCS8PrivateKey_bio@libcrypto.so.10 (2)
d2i_PrivateKey_bio@libcrypto.so.10 (2)
d2i_PUBKEY_bio@libcrypto.so.10 (2)
d2i_X509_bio@libcrypto.so.10 (2)
d2i_X509_CRL_bio@libcrypto.so.10 (2)
DSA_do_sign@libcrypto.so.10 (2)
DSA_do_verify@libcrypto.so.10 (2)
DSA_free@libcrypto.so.10 (2)
DSA_generate_key@libcrypto.so.10 (2)
DSA_generate_parameters_ex@libcrypto.so.10 (2)
DSA_new@libcrypto.so.10 (2)
DSA_SIG_free@libcrypto.so.10 (2)
DSA_SIG_new@libcrypto.so.10 (2)
DSA_size@libcrypto.so.10 (2)
ECDSA_do_sign@OPENSSL_1.0.1_EC (5)
ECDSA_do_verify@OPENSSL_1.0.1_EC (5)

```
ECDSA_SIG_free@OPENSSL_1.0.1_EC (5)
ECDSA_SIG_new@OPENSSL_1.0.1_EC (5)
EC_GROUP_get_order@OPENSSL_1.0.1_EC (5)
EC_KEY_free@OPENSSL_1.0.1_EC (5)
EC_KEY_get0_group@OPENSSL_1.0.1_EC (5)
ENGINE_cleanup@libcrypto.so.10 (2)
ERR_free_strings@libcrypto.so.10 (2)
ERR_load_crypto_strings@libcrypto.so.10 (2)
ERR_load_strings@libcrypto.so.10 (2)
ERR_put_error@libcrypto.so.10 (2)
ERR_remove_thread_state@libcrypto.so.10 (2)
EVP_aes_128_cbc@libcrypto.so.10 (2)
EVP_aes_192_cbc@libcrypto.so.10 (2)
EVP_aes_256_cbc@libcrypto.so.10 (2)
EVP_CIPHER_block_size@libcrypto.so.10 (2)
EVP_CIPHER_CTX_free@libcrypto.so.10 (2)
EVP_CIPHER_CTX_new@libcrypto.so.10 (2)
EVP_CIPHER_CTX_set_padding@libcrypto.so.10 (2)
EVP_CipherFinal@libcrypto.so.10 (2)
EVP_CipherInit@libcrypto.so.10 (2)
EVP_CIPHER_iv_length@libcrypto.so.10 (2)
EVP_CIPHER_key_length@libcrypto.so.10 (2)
EVP_CipherUpdate@libcrypto.so.10 (2)
EVP_cleanup@libcrypto.so.10 (2)
EVP_des_ede3_cbc@libcrypto.so.10 (2)
EVP_DigestFinal@libcrypto.so.10 (2)
EVP_DigestInit@libcrypto.so.10 (2)
EVP_DigestUpdate@libcrypto.so.10 (2)
EVP_md5@libcrypto.so.10 (2)
EVP_MD_CTX_create@libcrypto.so.10 (2)
EVP_MD_CTX_destroy@libcrypto.so.10 (2)
EVP_MD_size@libcrypto.so.10 (2)
EVP_PKEY_assign@libcrypto.so.10 (2)
EVP_PKEY_free@libcrypto.so.10 (2)
EVP_PKEY_get1_DSA@libcrypto.so.10 (2)
EVP_PKEY_get1_EC_KEY@libcrypto.so.10 (2)
EVP_PKEY_new@libcrypto.so.10 (2)
EVP_PKEY_size@libcrypto.so.10 (2)
EVP_read_pw_string@libcrypto.so.10 (2)
EVP_ripemd160@libcrypto.so.10 (2)
EVP_sha1@libcrypto.so.10 (2)
EVP_sha224@libcrypto.so.10 (2)
EVP_sha256@libcrypto.so.10 (2)
EVP_sha384@libcrypto.so.10 (2)
EVP_sha512@libcrypto.so.10 (2)
EVP_SignFinal@libcrypto.so.10 (2)
EVP_VerifyFinal@libcrypto.so.10 (2)
HMAC_CTX_cleanup@libcrypto.so.10 (2)
HMAC_Final@libcrypto.so.10 (2)
HMAC_Init_ex@libcrypto.so.10 (2)
HMAC_Update@libcrypto.so.10 (2)
i2d_X509_bio@libcrypto.so.10 (2)
i2d_X509_CRL_bio@libcrypto.so.10 (2)
OBJ_cmp@libcrypto.so.10 (2)
OPENSSL_add_all_algorithms_noconf@libcrypto.so.10 (2)
OPENSSL_config@libcrypto.so.10 (2)
```

```
PEM_read_bio_PrivateKey@libcrypto.so.10 (2)
PEM_read_bio_PUBKEY@libcrypto.so.10 (2)
PEM_read_bio_X509_AUX@libcrypto.so.10 (2)
PKCS12_free@libcrypto.so.10 (2)
PKCS12_parse@libcrypto.so.10 (2)
PKCS12_verify_mac@libcrypto.so.10 (2)
RAND_bytes@libcrypto.so.10 (2)
RAND_cleanup@libcrypto.so.10 (2)
RAND_file_name@libcrypto.so.10 (2)
RAND_load_file@libcrypto.so.10 (2)
RAND_status@libcrypto.so.10 (2)
RAND_write_file@libcrypto.so.10 (2)
RSA_free@libcrypto.so.10 (2)
RSA_generate_key_ex@libcrypto.so.10 (2)
RSA_new@libcrypto.so.10 (2)
RSA_padding_add_PKCS1_OAEP@libcrypto.so.10 (2)
RSA_padding_check_PKCS1_OAEP@libcrypto.so.10 (2)
RSA_private_decrypt@libcrypto.so.10 (2)
RSA_public_encrypt@libcrypto.so.10 (2)
RSA_size@libcrypto.so.10 (2)
SHA1@libcrypto.so.10 (2)
sk_delete@libcrypto.so.10 (2)
sk_dup@libcrypto.so.10 (2)
sk_free@libcrypto.so.10 (2)
sk_new@libcrypto.so.10 (2)
sk_new_null@libcrypto.so.10 (2)
sk_num@libcrypto.so.10 (2)
sk_pop_free@libcrypto.so.10 (2)
sk_push@libcrypto.so.10 (2)
sk_set_cmp_func@libcrypto.so.10 (2)
sk_sort@libcrypto.so.10 (2)
sk_value@libcrypto.so.10 (2)
X509_cmp_current_time@libcrypto.so.10 (2)
X509_cmp@libcrypto.so.10 (2)
X509_CRL_dup@libcrypto.so.10 (2)
X509_CRL_free@libcrypto.so.10 (2)
X509_CRL_verify@libcrypto.so.10 (2)
X509_dup@libcrypto.so.10 (2)
X509_free@libcrypto.so.10 (2)
X509_get_ext_by_NID@libcrypto.so.10 (2)
X509_get_ext@libcrypto.so.10 (2)
X509_get_issuer_name@libcrypto.so.10 (2)
X509_get_pubkey@libcrypto.so.10 (2)
X509_get_serialNumber@libcrypto.so.10 (2)
X509_get_subject_name@libcrypto.so.10 (2)
X509_issuer_name_hash@libcrypto.so.10 (2)
X509_LOOKUP_ctrl@libcrypto.so.10 (2)
X509_LOOKUP_file@libcrypto.so.10 (2)
X509_LOOKUP_hash_dir@libcrypto.so.10 (2)
X509_NAME_add_entry_by_txt@libcrypto.so.10 (2)
X509_NAME_entry_count@libcrypto.so.10 (2)
X509_NAME_ENTRY_get_data@libcrypto.so.10 (2)
X509_NAME_ENTRY_get_object@libcrypto.so.10 (2)
X509_NAME_free@libcrypto.so.10 (2)
X509_NAME_get_entry@libcrypto.so.10 (2)
X509_NAME_new@libcrypto.so.10 (2)
```

```
X509_NAME_oneline@libcrypto.so.10 (2)
X509_NAME_print_ex@libcrypto.so.10 (2)
X509_OBJECT_free_contents@libcrypto.so.10 (2)
X509_STORE_add_cert@libcrypto.so.10 (2)
X509_STORE_add_lookup@libcrypto.so.10 (2)
X509_STORE_CTX_cleanup@libcrypto.so.10 (2)
X509_STORE_CTX_get_current_cert@libcrypto.so.10 (2)
X509_STORE_CTX_get_error@libcrypto.so.10 (2)
X509_STORE_CTX_init@libcrypto.so.10 (2)
X509_STORE_CTX_set0_param@libcrypto.so.10 (2)
X509_STORE_CTX_set_time@libcrypto.so.10 (2)
X509_STORE_free@libcrypto.so.10 (2)
X509_STORE_get_by_subject@libcrypto.so.10 (2)
X509_STORE_new@libcrypto.so.10 (2)
X509_STORE_set1_param@libcrypto.so.10 (2)
X509_STORE_set_default_paths@libcrypto.so.10 (2)
X509_subject_name_hash@libcrypto.so.10 (2)
X509_TRUST_cleanup@libcrypto.so.10 (2)
X509V3_EXT_d2i@libcrypto.so.10 (2)
X509_verify_cert_error_string@libcrypto.so.10 (2)
X509_verify_cert@libcrypto.so.10 (2)
X509_VERIFY_PARAM_free@libcrypto.so.10 (2)
X509_VERIFY_PARAM_get_flags@libcrypto.so.10 (2)
X509_VERIFY_PARAM_new@libcrypto.so.10 (2)
X509_VERIFY_PARAM_set_depth@libcrypto.so.10 (2)
X509_VERIFY_PARAM_set_flags@libcrypto.so.10 (2)
X509_VERIFY_PARAM_set_time@libcrypto.so.10 (2)
```

GLIBC

```
calloc@GLIBC_2.2.5 (3)
__ctype_b_loc@GLIBC_2.3 (8)
__cxa_finalize@GLIBC_2.2.5 (3)
__errno_location@GLIBC_2.2.5 (3)
__fprintf_chk@GLIBC_2.3.4 (10)
fputc@GLIBC_2.2.5 (3)
free@GLIBC_2.2.5 (3)
fwrite@GLIBC_2.2.5 (3)
memcmp@GLIBC_2.2.5 (3)
memcpy@GLIBC_2.14 (7)
memset@GLIBC_2.2.5 (3)
__stack_chk_fail@GLIBC_2.4 (9)
strcmp@GLIBC_2.2.5 (3)
strlen@GLIBC_2.2.5 (3)
strtol@GLIBC_2.2.5 (3)
timegm@GLIBC_2.2.5 (3)
```

Libxml2

```
xmlCharStrdup@LIBXML2_2.4.30 (4)
xmlGetProp@LIBXML2_2.4.30 (4)
xmlNodeAddContent@LIBXML2_2.4.30 (4)
xmlNodeGetContent@LIBXML2_2.4.30 (4)
xmlNodeSetContent@LIBXML2_2.4.30 (4)
xmlStrcmp@LIBXML2_2.4.30 (4)
xmlStrdup@LIBXML2_2.4.30 (4)
```

xmlStrlen@LIBXML2_2.4.30 (4)
xmlStrPrintf@LIBXML2_2.6.0 (6)

starter (strongSwan starter program)

GLIBC

abort@GLIBC_2.2.5 (2)
alarm@GLIBC_2.2.5 (2)
__asprintf_chk@GLIBC_2.8 (6)
asprintf@GLIBC_2.2.5 (2)
clearerr@GLIBC_2.2.5 (2)
close@GLIBC_2.2.5 (4)
closelog@GLIBC_2.2.5 (2)
__cxa_atexit@GLIBC_2.2.5 (2)
dup2@GLIBC_2.2.5 (2)
__errno_location@GLIBC_2.2.5 (4)
execv@GLIBC_2.2.5 (2)
exit@GLIBC_2.2.5 (2)
fclose@GLIBC_2.2.5 (2)
ferror@GLIBC_2.2.5 (2)
fileno@GLIBC_2.2.5 (2)
fopen@GLIBC_2.2.5 (2)
fork@GLIBC_2.2.5 (4)
__fprintf_chk@GLIBC_2.3.4 (3)
fputc@GLIBC_2.2.5 (2)
fread@GLIBC_2.2.5 (2)
free@GLIBC_2.2.5 (2)
fwrite@GLIBC_2.2.5 (2)
getenv@GLIBC_2.2.5 (2)
getpid@GLIBC_2.2.5 (2)
getprotobyname@GLIBC_2.2.5 (2)
getservbyname@GLIBC_2.2.5 (2)
getuid@GLIBC_2.2.5 (2)
_IO_getc@GLIBC_2.2.5 (2)
isatty@GLIBC_2.2.5 (2)
__isoc99_sscanf@GLIBC_2.7 (8)
kill@GLIBC_2.2.5 (2)
__libc_start_main@GLIBC_2.2.5 (2)
malloc@GLIBC_2.2.5 (2)
memcpy@GLIBC_2.14 (7)
open@GLIBC_2.2.5 (4)
openlog@GLIBC_2.2.5 (2)
pselect@GLIBC_2.2.5 (2)
pthread_sigmask@GLIBC_2.2.5 (4)
realloc@GLIBC_2.2.5 (2)
setbuf@GLIBC_2.2.5 (2)
setenv@GLIBC_2.2.5 (2)
setsid@GLIBC_2.2.5 (2)
sigaction@GLIBC_2.2.5 (4)
sigaddset@GLIBC_2.2.5 (2)
sigemptyset@GLIBC_2.2.5 (2)
sigprocmask@GLIBC_2.2.5 (2)
__snprintf_chk@GLIBC_2.3.4 (3)
__sprintf_chk@GLIBC_2.3.4 (3)
__stack_chk_fail@GLIBC_2.4 (5)

stpcpy@GLIBC_2.2.5 (2)
strchr@GLIBC_2.2.5 (2)
strcmp@GLIBC_2.2.5 (2)
strcpy@GLIBC_2.2.5 (2)
__strdup@GLIBC_2.2.5 (2)
strlen@GLIBC_2.2.5 (2)
strtol@GLIBC_2.2.5 (2)
strtoul@GLIBC_2.2.5 (2)
strtoull@GLIBC_2.2.5 (2)
__syslog_chk@GLIBC_2.4 (5)
system@GLIBC_2.2.5 (4)
unlink@GLIBC_2.2.5 (2)
usleep@GLIBC_2.2.5 (2)
__vfprintf_chk@GLIBC_2.3.4 (3)
__vsnprintf_chk@GLIBC_2.3.4 (3)
waitpid@GLIBC_2.2.5 (4)
__xstat@GLIBC_2.2.5 (2)

stconfig (Stealth program)

GLIBC
accept@GLIBC_2.2.5 (5)
ASN1_INTEGER_get@libcrypto.so.10 (4)
ASN1_STRING_data@libcrypto.so.10 (4)
atoi@GLIBC_2.2.5 (2)
bind@GLIBC_2.2.5 (2)
calloc@GLIBC_2.2.5 (2)
close@GLIBC_2.2.5 (5)
closelog@GLIBC_2.2.5 (2)
connect@GLIBC_2.2.5 (5)
dup2@GLIBC_2.2.5 (2)
__errno_location@GLIBC_2.2.5 (5)
exit@GLIBC_2.2.5 (2)
fclose@GLIBC_2.2.5 (2)
fcntl@GLIBC_2.2.5 (5)
fflush@GLIBC_2.2.5 (2)
fgets@GLIBC_2.2.5 (2)
fileno@GLIBC_2.2.5 (2)
fopen@GLIBC_2.2.5 (2)
fork@GLIBC_2.2.5 (5)
fprintf@GLIBC_2.2.5 (2)
fread@GLIBC_2.2.5 (2)
free@GLIBC_2.2.5 (2)
freeifaddrs@GLIBC_2.3 (8)
fscanf@GLIBC_2.2.5 (2)
fseek@GLIBC_2.2.5 (2)
ftell@GLIBC_2.2.5 (2)
ftruncate@GLIBC_2.2.5 (2)
fwrite@GLIBC_2.2.5 (2)
getgrnam@GLIBC_2.2.5 (2)
gethostbyname@GLIBC_2.2.5 (2)
gethostname@GLIBC_2.2.5 (2)
getifaddrs@GLIBC_2.3 (8)
getpid@GLIBC_2.2.5 (2)

```
getpwuid@GLIBC_2.2.5 (2)
getsockopt@GLIBC_2.2.5 (2)
gettext@GLIBC_2.2.5 (2)
gettimeofday@GLIBC_2.2.5 (2)
__h_errno_location@GLIBC_2.2.5 (5)
hstrerror@GLIBC_2.2.5 (2)
htonl@GLIBC_2.2.5 (2)
htons@GLIBC_2.2.5 (2)
inet_ntop@GLIBC_2.2.5 (2)
inet_pton@GLIBC_2.2.5 (2)
isatty@GLIBC_2.2.5 (2)
__libc_start_main@GLIBC_2.2.5 (2)
listen@GLIBC_2.2.5 (2)
localtime@GLIBC_2.2.5 (2)
lockf@GLIBC_2.2.5 (2)
malloc@GLIBC_2.2.5 (2)
memcmp@GLIBC_2.2.5 (2)
memcpy@GLIBC_2.14 (7)
memset@GLIBC_2.2.5 (2)
ntohl@GLIBC_2.2.5 (2)
open@GLIBC_2.2.5 (5)
openlog@GLIBC_2.2.5 (2)
pipe@GLIBC_2.2.5 (2)
poll@GLIBC_2.2.5 (2)
printf@GLIBC_2.2.5 (2)
pthread_self@GLIBC_2.2.5 (5)
puts@GLIBC_2.2.5 (2)
read@GLIBC_2.2.5 (5)
rewind@GLIBC_2.2.5 (2)
scanf@GLIBC_2.2.5 (2)
setsid@GLIBC_2.2.5 (2)
setsockopt@GLIBC_2.2.5 (2)
sigaction@GLIBC_2.2.5 (5)
sigemptyset@GLIBC_2.2.5 (2)
signal@GLIBC_2.2.5 (2)
sleep@GLIBC_2.2.5 (2)
snprintf@GLIBC_2.2.5 (2)
socket@GLIBC_2.2.5 (2)
sprintf@GLIBC_2.2.5 (2)
sscanf@GLIBC_2.2.5 (2)
strcasecmp@GLIBC_2.2.5 (2)
strchr@GLIBC_2.2.5 (2)
strcmp@GLIBC_2.2.5 (2)
strcpy@GLIBC_2.2.5 (2)
strftime@GLIBC_2.2.5 (2)
strlen@GLIBC_2.2.5 (2)
strncmp@GLIBC_2.2.5 (2)
strncpy@GLIBC_2.2.5 (2)
strstr@GLIBC_2.2.5 (2)
strtoull@GLIBC_2.2.5 (2)
syslog@GLIBC_2.2.5 (2)
tcgetattr@GLIBC_2.2.5 (2)
tcsetattr@GLIBC_2.2.5 (2)
time@GLIBC_2.2.5 (2)
umask@GLIBC_2.2.5 (2)
uname@GLIBC_2.2.5 (2)
```

unlink@GLIBC_2.2.5 (2)
usleep@GLIBC_2.2.5 (2)
write@GLIBC_2.2.5 (5)
writev@GLIBC_2.2.5 (2)
__xstat@GLIBC_2.2.5 (2)

libcrypto (OpenSSL)
BASIC_CONSTRAINTS_free@libcrypto.so.10 (4)
BIO_ctrl@libcrypto.so.10 (4)
BIO_free@libcrypto.so.10 (4)
BIO_new@libcrypto.so.10 (4)
BIO_new_mem_buf@libcrypto.so.10 (4)
BIO_read@libcrypto.so.10 (4)
BIO_s_mem@libcrypto.so.10 (4)
BN_free@libcrypto.so.10 (4)
BN_new@libcrypto.so.10 (4)
BN_set_word@libcrypto.so.10 (4)
CRYPTO_num_locks@libcrypto.so.10 (4)
CRYPTO_set_dynlock_create_callback@libcrypto.so.10 (4)
CRYPTO_set_dynlock_destroy_callback@libcrypto.so.10 (4)
CRYPTO_set_dynlock_lock_callback@libcrypto.so.10 (4)
CRYPTO_set_locking_callback@libcrypto.so.10 (4)
CRYPTO_THREADID_set_callback@libcrypto.so.10 (4)
CRYPTO_THREADID_set_numeric@libcrypto.so.10 (4)
d2i_GENERAL_NAMES@libcrypto.so.10 (4)
d2i_PKCS12_fp@libcrypto.so.10 (4)
d2i_X509_fp@libcrypto.so.10 (4)
d2i_X509@libcrypto.so.10 (4)
ERR_clear_error@libcrypto.so.10 (4)
ERR_error_string@libcrypto.so.10 (4)
ERR_error_string_n@libcrypto.so.10 (4)
ERR_free_strings@libcrypto.so.10 (4)
ERR_get_error@libcrypto.so.10 (4)
ERR_load_crypto_strings@libcrypto.so.10 (4)
EVP_cleanup@libcrypto.so.10 (4)
EVP_PKEY_free@libcrypto.so.10 (4)
EVP_PKEY_get1_RSA@libcrypto.so.10 (4)
EXTENDED_KEY_USAGE_free@libcrypto.so.10 (4)
FIPS_mode_set@libcrypto.so.10 (4)
GENERAL_NAMES_free@libcrypto.so.10 (4)
i2d_X509_bio@libcrypto.so.10 (4)
OBJ_create@libcrypto.so.10 (4)
OBJ_nid2sn@libcrypto.so.10 (4)
OBJ_obj2nid@libcrypto.so.10 (4)
OBJ_obj2txt@libcrypto.so.10 (4)
OPENSSL_add_all_algorithms_noconf@libcrypto.so.10 (4)
OPENSSL_config@libcrypto.so.10 (4)
PEM_read_bio_RSA_PUBKEY@libcrypto.so.10 (4)
PEM_read_X509@libcrypto.so.10 (4)
PEM_write_bio_RSAPrivateKey@libcrypto.so.10 (4)
PEM_write_bio_RSA_PUBKEY@libcrypto.so.10 (4)
PEM_write_bio_X509@libcrypto.so.10 (4)
PKCS12_free@libcrypto.so.10 (4)
PKCS12_parse@libcrypto.so.10 (4)
RAND_load_file@libcrypto.so.10 (4)

RSA_free@libcrypto.so.10 (4)
RSA_generate_key_ex@libcrypto.so.10 (4)
RSA_new@libcrypto.so.10 (4)
RSA_private_decrypt@libcrypto.so.10 (4)
RSA_public_encrypt@libcrypto.so.10 (4)
RSA_size@libcrypto.so.10 (4)
sk_free@libcrypto.so.10 (4)
sk_new_null@libcrypto.so.10 (4)
sk_num@libcrypto.so.10 (4)
sk_push@libcrypto.so.10 (4)
sk_value@libcrypto.so.10 (4)
X509_cmp_current_time@libcrypto.so.10 (4)
X509_free@libcrypto.so.10 (4)
X509_get_ext_by_NID@libcrypto.so.10 (4)
X509_get_ext_count@libcrypto.so.10 (4)
X509_get_ext_d2i@libcrypto.so.10 (4)
X509_get_ext@libcrypto.so.10 (4)
X509_get_issuer_name@libcrypto.so.10 (4)
X509_get_pubkey@libcrypto.so.10 (4)
X509_get_subject_name@libcrypto.so.10 (4)
X509_NAME_entry_count@libcrypto.so.10 (4)
X509_NAME_ENTRY_get_data@libcrypto.so.10 (4)
X509_NAME_ENTRY_get_object@libcrypto.so.10 (4)
X509_NAME_get_entry@libcrypto.so.10 (4)
X509_NAME_oneline@libcrypto.so.10 (4)
X509_new@libcrypto.so.10 (4)
X509_STORE_add_cert@libcrypto.so.10 (4)
X509_STORE_CTX_free@libcrypto.so.10 (4)
X509_STORE_CTX_get_error@libcrypto.so.10 (4)
X509_STORE_CTX_init@libcrypto.so.10 (4)
X509_STORE_CTX_new@libcrypto.so.10 (4)
X509_STORE_free@libcrypto.so.10 (4)
X509_STORE_new@libcrypto.so.10 (4)
X509V3_EXT_d2i@libcrypto.so.10 (4)
X509_verify_cert_error_string@libcrypto.so.10 (4)
X509_verify_cert@libcrypto.so.10 (4)

libcurl (cURL packaged with Stealth)
curl_easy_cleanup
curl_easy_getinfo
curl_easy_init
curl_easy_perform
curl_easy_setopt
curl_easy_strerror
curl_global_cleanup
curl_global_init
curl_slist_append
curl_slist_free_all
curl_version
curl_version_info

Glib
g_ascii_strcasecmp
g_ascii_strncasecmp

g_assertion_message_expr
g_async_queue_length
g_async_queue_new
g_async_queue_pop
g_async_queue_push
g_async_queue_timed_pop
g_async_queue_unref
g_base64_decode
g_base64_encode
g_clear_error
g_convert
g_dir_close
g_dir_open
g_dir_read_name
g_error_free
g_file_get_contents
g_file_test
g_free
g_get_current_time
g_get_home_dir
g_get_user_name
g_idle_add_full
g_io_channel_shutdown
g_key_file_free
g_key_file_get_boolean
g_key_file_get_groups
g_key_file_get_integer
g_key_file_get_keys
g_key_file_get_string
g_key_file_get_string_list
g_key_file_get_value
g_key_file_has_group
g_key_file_has_key
g_key_file_load_from_data
g_key_file_load_from_file
g_key_file_new
g_key_file_set_value
g_key_file_to_data
g_log
g_log_set_default_handler
g_logv
g_malloc
g_malloc0
g_malloc0_n
g_markup_printf_escaped
g_option_context_add_main_entries
g_option_context_free
g_option_context_new
g_option_context_parse
g_option_context_set_main_group
g_option_group_new
g_printerr
g_propagate_error
g_quark_from_string
g_realloc
g_rw_lock_clear

g_rw_lock_init
g_rw_lock_writer_lock
g_rw_lock_writer_unlock
g_set_error
g_slice_alloc
g_slice_alloc0
g_slice_free1
g_slist_append
g_slist_copy
g_slist_find_custom
g_slist_foreach
g_slist_free
g_slist_insert_sorted
g_slist_last
g_slist_length
g_slist_nth
g_slist_prepend
g_slist_reverse
g_slist_sort
g_snprintf
g_source_remove
g_stpcpy
g_strchomp
g_strchug
g_strcmp0
g_strconcat
g_strdelimit
g_strdup
g_strdup_printf
g_strdupv
g_strerror
g_strfreev
g_str_has_prefix
g_str_has_suffix
g_string_append
g_string_append_len
g_string_append_printf
g_string_assign
g_string_free
g_string_insert_c
g_string_new
g_string_new_len
g_string_prepend
g_string_printf
g_string_set_size
g_string_sized_new
g_string_truncate
g_strjoin
g_strsplit
g_strstr_len
g_thread_create
g_thread_join
g_timeout_add
g_timer_destroy
g_timer_elapsed
g_timer_new

g_timer_start
g_timer_stop
g_time_val_add
g_try_malloc0
g_try_realloc
g_utf16_to_utf8

Libxml2
xmlAddChild@LIBXML2_2.4.30 (6)
xmlCheckVersion@LIBXML2_2.4.30 (6)
xmlCleanupParser@LIBXML2_2.4.30 (6)
xmlDocDumpFormatMemoryEnc@LIBXML2_2.4.30 (6)
xmlDocDumpMemoryEnc@LIBXML2_2.4.30 (6)
xmlDocGetRootElement@LIBXML2_2.4.30 (6)
xmlFreeDoc@LIBXML2_2.4.30 (6)
xmlGetLastError@LIBXML2_2.6.0 (3)
xmlGetProp@LIBXML2_2.4.30 (6)
xmlInitParser@LIBXML2_2.4.30 (6)
xmlNodeGetContent@LIBXML2_2.4.30 (6)
xmlNodeListGetString@LIBXML2_2.4.30 (6)
xmlParseFile@LIBXML2_2.4.30 (6)
xmlReadMemory@LIBXML2_2.6.0 (3)
xmlStrcmp@LIBXML2_2.4.30 (6)
xmlXPathEvalExpression@LIBXML2_2.4.30 (6)
xmlXPathFreeContext@LIBXML2_2.4.30 (6)
xmlXPathFreeObject@LIBXML2_2.4.30 (6)
xmlXPathNewContext@LIBXML2_2.4.30 (6)
xmlXPathRegisterNs@LIBXML2_2.4.30 (6)

libxmlsec1 (XMLSec packaged with Stealth)
xmlSecCheckVersionExt
xmlSecDSigCtxCreate
xmlSecDSigCtxDestroy
xmlSecDSigCtxSign
xmlSecDSigCtxVerify
xmlSecErrorsDefaultCallbackEnableOutput
xmlSecErrorsSetCallback
xmlSecFindNode
xmlSecInit
xmlSecKeysMngrCreate
xmlSecKeysMngrDestroy
xmlSecOpenSSLAppDefaultKeysMngrInit
xmlSecOpenSSLAppInit
xmlSecOpenSSLAppKeyLoadMemory
xmlSecOpenSSLAppKeysMngrAddCertsFile
xmlSecOpenSSLAppKeysMngrAddCertsPath
xmlSecOpenSSLAppShutdown
xmlSecOpenSSLInit
xmlSecOpenSSLShutdown
xmlSecOpenSSLTransformRsaSha1GetKlass
xmlSecOpenSSLTransformSha1GetKlass
xmlSecShutdown
xmlSecTmplReferenceAddTransform
xmlSecTmplSignatureAddReference

xmlSecTmplSignatureCreate
xmlSecTransformEnvelopedGetKlass
xmlSecTransformInclC14NGetKlass

stealthd (Stealth daemon)

GLIBC
accept@GLIBC_2.2.5 (4)
__assert_fail@GLIBC_2.2.5 (5)
atoi@GLIBC_2.2.5 (5)
bind@GLIBC_2.2.5 (5)
calloc@GLIBC_2.2.5 (5)
close@GLIBC_2.2.5 (4)
closelog@GLIBC_2.2.5 (5)
__cmsg_nxthdr@GLIBC_2.2.5 (5)
connect@GLIBC_2.2.5 (4)
dup2@GLIBC_2.2.5 (5)
__errno_location@GLIBC_2.2.5 (4)
exit@GLIBC_2.2.5 (5)
fclose@GLIBC_2.2.5 (5)
fcntl@GLIBC_2.2.5 (4)
fgetc@GLIBC_2.2.5 (5)
fileno@GLIBC_2.2.5 (5)
fopen@GLIBC_2.2.5 (5)
fork@GLIBC_2.2.5 (4)
fprintf@GLIBC_2.2.5 (5)
fputc@GLIBC_2.2.5 (5)
free@GLIBC_2.2.5 (5)
freeifaddrs@GLIBC_2.3 (10)
ftruncate@GLIBC_2.2.5 (5)
fwrite@GLIBC_2.2.5 (5)
getdtablesize@GLIBC_2.2.5 (5)
getgrnam@GLIBC_2.2.5 (5)
gethostbyname@GLIBC_2.2.5 (5)
gethostname@GLIBC_2.2.5 (5)
getifaddrs@GLIBC_2.3 (10)
getpid@GLIBC_2.2.5 (5)
getpwuid@GLIBC_2.2.5 (5)
getsockname@GLIBC_2.2.5 (5)
getsockopt@GLIBC_2.2.5 (5)
gettext@GLIBC_2.2.5 (5)
gettimeofday@GLIBC_2.2.5 (5)
__h_errno_location@GLIBC_2.2.5 (4)
hstrerror@GLIBC_2.2.5 (5)
htonl@GLIBC_2.2.5 (5)
htons@GLIBC_2.2.5 (5)
if_indextoname@GLIBC_2.2.5 (5)
if_nametoindex@GLIBC_2.2.5 (5)
inet_aton@GLIBC_2.2.5 (5)
inet_ntoa@GLIBC_2.2.5 (5)
inet_ntop@GLIBC_2.2.5 (5)
inet_pton@GLIBC_2.2.5 (5)
ioctl@GLIBC_2.2.5 (5)
__libc_start_main@GLIBC_2.2.5 (5)
libnetfilter_queue

```
libxtables
listen@GLIBC_2.2.5 (5)
localtime@GLIBC_2.2.5 (5)
lockf@GLIBC_2.2.5 (5)
memcmp@GLIBC_2.2.5 (5)
memcpy@GLIBC_2.14 (9)
memset@GLIBC_2.2.5 (5)
mkdir@GLIBC_2.2.5 (5)
nice@GLIBC_2.2.5 (5)
nfq_bind_pf
nfq_close
nfq_create_queue
nfq_destroy_queue
nfq_fd
nfq_get_msg_packet_hdr
nfq_get_nfmark
nfq_get_payload
nfq_handle_packet
nfq_open
nfq_set_mode
nfq_set_verdict
nfq_set_verdict2
nfq_unbind_pf
ntohl@GLIBC_2.2.5 (5)
ntohs@GLIBC_2.2.5 (5)
open@GLIBC_2.2.5 (4)
openlog@GLIBC_2.2.5 (5)
pipe@GLIBC_2.2.5 (5)
poll@GLIBC_2.2.5 (5)
printf@GLIBC_2.2.5 (5)
pthread_self@GLIBC_2.2.5 (4)
putchar@GLIBC_2.2.5 (5)
putenv@GLIBC_2.2.5 (5)
puts@GLIBC_2.2.5 (5)
read@GLIBC_2.2.5 (4)
recvfrom@GLIBC_2.2.5 (4)
recv@GLIBC_2.2.5 (4)
recvmsg@GLIBC_2.2.5 (4)
remove@GLIBC_2.2.5 (5)
sendmsg@GLIBC_2.2.5 (4)
sendto@GLIBC_2.2.5 (4)
setsid@GLIBC_2.2.5 (5)
setsockopt@GLIBC_2.2.5 (5)
sigaction@GLIBC_2.2.5 (4)
sigemptyset@GLIBC_2.2.5 (5)
signal@GLIBC_2.2.5 (5)
sleep@GLIBC_2.2.5 (5)
snprintf@GLIBC_2.2.5 (5)
socket@GLIBC_2.2.5 (5)
sprintf@GLIBC_2.2.5 (5)
sscanf@GLIBC_2.2.5 (5)
strcasecmp@GLIBC_2.2.5 (5)
strchr@GLIBC_2.2.5 (5)
strcmp@GLIBC_2.2.5 (5)
strcpy@GLIBC_2.2.5 (5)
strerror@GLIBC_2.2.5 (5)
```

strftime@GLIBC_2.2.5 (5)
strlen@GLIBC_2.2.5 (5)
strncmp@GLIBC_2.2.5 (5)
strncpy@GLIBC_2.2.5 (5)
strstr@GLIBC_2.2.5 (5)
strtol@GLIBC_2.2.5 (5)
strtoul@GLIBC_2.2.5 (5)
strtoull@GLIBC_2.2.5 (5)
syslog@GLIBC_2.2.5 (5)
time@GLIBC_2.2.5 (5)
umask@GLIBC_2.2.5 (5)
uname@GLIBC_2.2.5 (5)
unlink@GLIBC_2.2.5 (5)
write@GLIBC_2.2.5 (4)
writev@GLIBC_2.2.5 (5)
__xstat@GLIBC_2.2.5 (5)
xtables_insmod

libcrypto and OPENSSL (OpenSSL)
ASN1_INTEGER_get@libcrypto.so.10 (3)
ASN1_STRING_data@libcrypto.so.10 (3)
BASIC_CONSTRAINTS_free@libcrypto.so.10 (3)
BIO_ctrl@libcrypto.so.10 (3)
BIO_free@libcrypto.so.10 (3)
BIO_new@libcrypto.so.10 (3)
BIO_new_mem_buf@libcrypto.so.10 (3)
BIO_read@libcrypto.so.10 (3)
BIO_s_mem@libcrypto.so.10 (3)
BN_free@libcrypto.so.10 (3)
BN_new@libcrypto.so.10 (3)
BN_set_word@libcrypto.so.10 (3)
CRYPTO_num_locks@libcrypto.so.10 (3)
CRYPTO_set_dynlock_create_callback@libcrypto.so.10 (3)
CRYPTO_set_dynlock_destroy_callback@libcrypto.so.10 (3)
CRYPTO_set_dynlock_lock_callback@libcrypto.so.10 (3)
CRYPTO_set_locking_callback@libcrypto.so.10 (3)
CRYPTO_THREADID_set_callback@libcrypto.so.10 (3)
CRYPTO_THREADID_set_numeric@libcrypto.so.10 (3)
d2i_GENERAL_NAMES@libcrypto.so.10 (3)
d2i_PKCS12_fp@libcrypto.so.10 (3)
d2i_PUBKEY@libcrypto.so.10 (3)
d2i_X509_fp@libcrypto.so.10 (3)
d2i_X509@libcrypto.so.10 (3)
EC_KEY_free@OPENSSL_1.0.1_EC (7)
EC_KEY_generate_key@OPENSSL_1.0.1_EC (7)
EC_KEY_new_by_curve_name@OPENSSL_1.0.1_EC (7)
ERR_clear_error@libcrypto.so.10 (3)
ERR_error_string@libcrypto.so.10 (3)
ERR_error_string_n@libcrypto.so.10 (3)
ERR_free_strings@libcrypto.so.10 (3)
ERR_get_error@libcrypto.so.10 (3)
ERR_load_crypto_strings@libcrypto.so.10 (3)
ERR_print_errors_fp@libcrypto.so.10 (3)
EVP_aes_128_cbc@libcrypto.so.10 (3)
EVP_aes_256_cbc@libcrypto.so.10 (3)

EVP_CIPHER_block_size@libcrypto.so.10 (3)
EVP_CIPHER_CTX_cleanup@libcrypto.so.10 (3)
EVP_CIPHER_CTX_init@libcrypto.so.10 (3)
EVP_CIPHER_CTX_set_padding@libcrypto.so.10 (3)
EVP_cleanup@libcrypto.so.10 (3)
EVP_DecryptFinal_ex@libcrypto.so.10 (3)
EVP_DecryptInit_ex@libcrypto.so.10 (3)
EVP_DecryptUpdate@libcrypto.so.10 (3)
EVP_DigestFinal_ex@libcrypto.so.10 (3)
EVP_DigestInit_ex@libcrypto.so.10 (3)
EVP_DigestSignFinal@libcrypto.so.10 (3)
EVP_DigestSignInit@libcrypto.so.10 (3)
EVP_DigestUpdate@libcrypto.so.10 (3)
EVP_DigestVerifyFinal@libcrypto.so.10 (3)
EVP_DigestVerifyInit@libcrypto.so.10 (3)
EVP_EncryptFinal_ex@libcrypto.so.10 (3)
EVP_EncryptInit_ex@libcrypto.so.10 (3)
EVP_EncryptUpdate@libcrypto.so.10 (3)
EVP_MD_CTX_cleanup@libcrypto.so.10 (3)
EVP_MD_CTX_create@libcrypto.so.10 (3)
EVP_MD_CTX_init@libcrypto.so.10 (3)
EVP_MD_size@libcrypto.so.10 (3)
EVP_PKEY_assign@libcrypto.so.10 (3)
EVP_PKEY_CTX_ctrl@libcrypto.so.10 (3)
EVP_PKEY_CTX_free@libcrypto.so.10 (3)
EVP_PKEY_CTX_new_id@libcrypto.so.10 (3)
EVP_PKEY_CTX_new@libcrypto.so.10 (3)
EVP_PKEY_derive_init@libcrypto.so.10 (3)
EVP_PKEY_derive@libcrypto.so.10 (3)
EVP_PKEY_derive_set_peer@libcrypto.so.10 (3)
EVP_PKEY_free@libcrypto.so.10 (3)
EVP_PKEY_get1_EC_KEY@libcrypto.so.10 (3)
EVP_PKEY_get1_RSA@libcrypto.so.10 (3)
EVP_PKEY_keygen_init@libcrypto.so.10 (3)
EVP_PKEY_keygen@libcrypto.so.10 (3)
EVP_PKEY_new@libcrypto.so.10 (3)
EVP_PKEY_paramgen_init@libcrypto.so.10 (3)
EVP_PKEY_paramgen@libcrypto.so.10 (3)
EVP_PKEY_set1_RSA@libcrypto.so.10 (3)
EVP_PKEY_sign_init@libcrypto.so.10 (3)
EVP_PKEY_sign@libcrypto.so.10 (3)
EVP_PKEY_verify_init@libcrypto.so.10 (3)
EVP_PKEY_verify@libcrypto.so.10 (3)
EVP_sha1@libcrypto.so.10 (3)
EVP_sha256@libcrypto.so.10 (3)
EVP_sha384@libcrypto.so.10 (3)
EXTENDED_KEY_USAGE_free@libcrypto.so.10 (3)
FIPS_mode_set@libcrypto.so.10 (3)
GENERAL_NAMES_free@libcrypto.so.10 (3)
HMAC@libcrypto.so.10 (3)
i2d_PUBKEY@libcrypto.so.10 (3)
i2d_X509_bio@libcrypto.so.10 (3)
i2o_ECPublicKey@libcrypto.so.10 (3)
o2i_ECPublicKey@libcrypto.so.10 (3)
OBJ_create@libcrypto.so.10 (3)
OBJ_nid2sn@libcrypto.so.10 (3)

OBJ_obj2nid@libcrypto.so.10 (3)
OBJ_obj2txt@libcrypto.so.10 (3)
OPENSSL_add_all_algorithms_noconf@libcrypto.so.10 (3)
OPENSSL_config@libcrypto.so.10 (3)
PEM_read_bio_RSA_PUBKEY@libcrypto.so.10 (3)
PEM_read_X509@libcrypto.so.10 (3)
PEM_write_bio_PrivateKey@libcrypto.so.10 (3)
PEM_write_bio_RSAPrivateKey@libcrypto.so.10 (3)
PEM_write_bio_RSA_PUBKEY@libcrypto.so.10 (3)
PEM_write_bio_X509@libcrypto.so.10 (3)
PKCS12_free@libcrypto.so.10 (3)
PKCS12_parse@libcrypto.so.10 (3)
RAND_bytes@libcrypto.so.10 (3)
RAND_load_file@libcrypto.so.10 (3)
RSA_free@libcrypto.so.10 (3)
RSA_generate_key_ex@libcrypto.so.10 (3)
RSA_new@libcrypto.so.10 (3)
RSA_private_decrypt@libcrypto.so.10 (3)
RSA_public_encrypt@libcrypto.so.10 (3)
RSA_size@libcrypto.so.10 (3)
sk_free@libcrypto.so.10 (3)
sk_new_null@libcrypto.so.10 (3)
sk_num@libcrypto.so.10 (3)
sk_push@libcrypto.so.10 (3)
sk_value@libcrypto.so.10 (3)
X509_cmp_current_time@libcrypto.so.10 (3)
X509_free@libcrypto.so.10 (3)
X509_get_ext_by_NID@libcrypto.so.10 (3)
X509_get_ext_count@libcrypto.so.10 (3)
X509_get_ext_d2i@libcrypto.so.10 (3)
X509_get_ext@libcrypto.so.10 (3)
X509_get_issuer_name@libcrypto.so.10 (3)
X509_get_pubkey@libcrypto.so.10 (3)
X509_get_subject_name@libcrypto.so.10 (3)
X509_NAME_entry_count@libcrypto.so.10 (3)
X509_NAME_ENTRY_get_data@libcrypto.so.10 (3)
X509_NAME_ENTRY_get_object@libcrypto.so.10 (3)
X509_NAME_get_entry@libcrypto.so.10 (3)
X509_NAME_oneline@libcrypto.so.10 (3)
X509_new@libcrypto.so.10 (3)
X509_STORE_add_cert@libcrypto.so.10 (3)
X509_STORE_CTX_free@libcrypto.so.10 (3)
X509_STORE_CTX_get_error@libcrypto.so.10 (3)
X509_STORE_CTX_init@libcrypto.so.10 (3)
X509_STORE_CTX_new@libcrypto.so.10 (3)
X509_STORE_free@libcrypto.so.10 (3)
X509_STORE_new@libcrypto.so.10 (3)
X509V3_EXT_d2i@libcrypto.so.10 (3)
X509_verify_cert_error_string@libcrypto.so.10 (3)
X509_verify_cert@libcrypto.so.10 (3)

libcurl (cURL packaged with Stealth)
curl_easy_cleanup
curl_easy_getinfo
curl_easy_init

curl_easy_perform
curl_easy_setopt
curl_easy_strerror
curl_global_cleanup
curl_global_init
curl_slist_append
curl_slist_free_all
curl_version
curl_version_info

libdavici (packaged with Stealth)
davici_cancel
davici_connect_unix
davici_disconnect
davici_get_name
davici_get_value_str
davici_kv
davici_kvf
davici_list_end
davici_list_item
davici_list_itemf
davici_list_start
davici_name_strcmp
davici_new_cmd
davici_parse
davici_queue
davici_read
davici_register
davici_section_end
davici_section_start
davici_unregister
davici_value_strcmp
davici_write

Glib
g_array_append_vals
g_array_free
g_array_new
g_array_sized_new
g_array_unref
g_ascii_strcasecmp
g_ascii_strncasecmp
g_ascii_strtoll
g_assertion_message_expr
g_async_queue_length
g_async_queue_new
g_async_queue_pop
g_async_queue_push
g_async_queue_timed_pop
g_async_queue_try_pop
g_async_queue_unref
g_base64_decode
g_base64_encode
g_build_filename
g_child_watch_add
g_clear_error

g_convert
g_dir_close
g_dir_open
g_dir_read_name
g_error_copy
g_error_free
g_file_get_contents
g_file_set_contents
g_file_test
g_free
g_get_current_time
g_getenv
g_hash_table_destroy
g_hash_table_iter_init
g_hash_table_iter_next
g_hash_table_lookup
g_hash_table_new
g_hash_table_remove
g_hash_table_replace
g_hash_table_size
g_idle_add_full
g_idle_source_new
g_io_add_watch
g_io_add_watch_full
g_io_channel_flush
g_io_channel_read_line
g_io_channel_read_line_string
g_io_channel_set_close_on_unref
g_io_channel_set_encoding
g_io_channel_shutdown
g_io_channel_unix_get_fd
g_io_channel_unix_new
g_io_channel_unref
g_io_channel_write_chars
g_io_create_watch
g_key_file_free
g_key_file_get_boolean
g_key_file_get_groups
g_key_file_get_integer
g_key_file_get_keys
g_key_file_get_string
g_key_file_get_value
g_key_file_has_group
g_key_file_has_key
g_key_file_load_from_file
g_key_file_new
g_key_file_set_value
g_list_append
g_list_find_custom
g_list_foreach
g_list_free
g_list_free_1
g_list_length
g_list_prepend
g_list_remove_link
g_log

g_log_set_default_handler
g_logv
g_main_context_new
g_main_context_unref
g_main_loop_get_context
g_main_loop_new
g_main_loop_quit
g_main_loop_run
g_main_loop_unref
g_malloc
g_malloc0
g_malloc0_n
g_markup_printf_escaped
g_option_context_add_main_entries
g_option_context_free
g_option_context_new
g_option_context_parse
g_path_get_basename
g_propagate_error
g_quark_from_static_string
g_quark_from_string
g_quark_to_string
g_queue_copy
g_queue_delete_link
g_queue_find_custom
g_queue_foreach
g_queue_free
g_queue_index
g_queue_insert_before
g_queue_is_empty
g_queue_new
g_queue_peek_head
g_queue_peek_head_link
g_queue_peek_nth
g_queue_peek_nth_link
g_queue_peek_tail_link
g_queue_pop_head
g_queue_pop_tail
g_queue_push_head
g_queue_push_tail
g_rand_int_range
g_rand_new
g_realloc
g_return_if_fail_warning
g_rw_lock_clear
g_rw_lock_init
g_rw_lock_reader_lock
g_rw_lock_reader_unlock
g_rw_lock_writer_lock
g_rw_lock_writer_unlock
g_set_error
g_shell_parse_argv
g_slice_alloc
g_slice_alloc0
g_slice_free1
g_slist_append

g_slist_find_custom
g_slist_foreach
g_slist_free
g_slist_index
g_slist_insert
g_slist_insert_sorted
g_slist_last
g_slist_length
g_slist_nth
g_slist_nth_data
g_slist_sort
g_snprintf
g_source_attach
g_source_destroy
g_source_new
g_source_remove
g_source_remove_by_user_data
g_source_set_callback
g_source_set_priority
g_source_unref
g_spawn_async
g_spawn_async_with_pipes
g_spawn_close_pid
g_spawn_command_line_sync
g_stpcpy
g_strchomp
g_strchug
g_strcmp0
g_strconcat
g_strdelimit
g_strdup
g_strdup_printf
g_strdupv
g_strdup_vprintf
g_str_equal
g_strerror
g_strfreev
g_str_hash
g_str_has_prefix
g_string_append
g_string_append_len
g_string_append_printf
g_string_assign
g_string_erase
g_string_free
g_string_insert_c
g_string_new
g_string_new_len
g_string_prepend
g_string_printf
g_string_set_size
g_string_sized_new
g_string_truncate
g_strjoin
g_strjoinv
g_strlcpy

g_strndup
g_strrstr
g_strsplit
g_strstr_len
g_thread_create
g_thread_init
g_thread_join
g_thread_pool_free
g_thread_pool_new
g_thread_pool_push
g_timeout_add
g_timeout_add_full
g_timeout_add_seconds
g_time_val_add
g_tree_destroy
g_tree_foreach
g_tree_height
g_tree_insert
g_tree_lookup
g_tree_new_full
g_tree_nnodes
g_tree_remove
g_try_malloc0
g_try_realloc
g_unix_fd_add_full
g_usleep
g_utf16_to_utf8

libiptc
ip6tc_append_entry
ip6tc_builtin
ip6tc_check_entry
ip6tc_commit
ip6tc_create_chain
ip6tc_delete_chain
ip6tc_delete_entry
ip6tc_flush_entries
ip6tc_free
ip6tc_init
ip6tc_insert_entry
ip6tc_is_chain
ip6tc_strerror
iptc_append_entry
iptc_builtin
iptc_check_entry
iptc_commit
iptc_create_chain
iptc_delete_chain
iptc_delete_entry
iptc_flush_entries
iptc_free
iptc_init
iptc_insert_entry
iptc_is_chain
iptc_strerror

libxtables
xtables_insmod

Libxml2
xmlAddChild@LIBXML2_2.4.30 (6)
xmlCheckVersion@LIBXML2_2.4.30 (6)
xmlCleanupParser@LIBXML2_2.4.30 (6)
xmlCopyNode@LIBXML2_2.4.30 (6)
xmlDocDumpFormatMemoryEnc@LIBXML2_2.4.30 (6)
xmlDocDumpMemoryEnc@LIBXML2_2.4.30 (6)
xmlDocGetRootElement@LIBXML2_2.4.30 (6)
xmlDocSetRootElement@LIBXML2_2.4.30 (6)
xmlFirstElementChild@LIBXML2_2.7.3 (8)
xmlFreeDoc@LIBXML2_2.4.30 (6)
xmlFreeNode@LIBXML2_2.4.30 (6)
xmlGetLastError@LIBXML2_2.6.0 (2)
xmlGetProp@LIBXML2_2.4.30 (6)
xmlHasProp@LIBXML2_2.4.30 (6)
xmlInitParser@LIBXML2_2.4.30 (6)
xmlIsBlankNode@LIBXML2_2.4.30 (6)
xmlNewDoc@LIBXML2_2.4.30 (6)
xmlNewNode@LIBXML2_2.4.30 (6)
xmlNewProp@LIBXML2_2.4.30 (6)
xmlNodeGetContent@LIBXML2_2.4.30 (6)
xmlNodeListGetString@LIBXML2_2.4.30 (6)
xmlParseFile@LIBXML2_2.4.30 (6)
xmlReadMemory@LIBXML2_2.6.0 (2)
xmlRemoveProp@LIBXML2_2.4.30 (6)
xmlStrcmp@LIBXML2_2.4.30 (6)
xmlStrlen@LIBXML2_2.4.30 (6)
xmlUnlinkNode@LIBXML2_2.4.30 (6)
xmlXPathEvalExpression@LIBXML2_2.4.30 (6)
xmlXPathFreeContext@LIBXML2_2.4.30 (6)
xmlXPathFreeObject@LIBXML2_2.4.30 (6)
xmlXPathNewContext@LIBXML2_2.4.30 (6)
xmlXPathRegisterNs@LIBXML2_2.4.30 (6)

libxmlsec1 (XMLSec packaged with Stealth)
xmlSecCheckVersionExt
xmlSecDSigCtxCreate
xmlSecDSigCtxDestroy
xmlSecDSigCtxSign
xmlSecDSigCtxVerify
xmlSecErrorsDefaultCallbackEnableOutput
xmlSecErrorsSetCallback
xmlSecFindNode
xmlSecInit
xmlSecKeysMngrCreate
xmlSecKeysMngrDestroy
xmlSecOpenSSLAppDefaultKeysMngrInit
xmlSecOpenSSLAppInit
xmlSecOpenSSLAppKeyLoadMemory

xmlSecOpenSSLAppKeysMngrAddCertsFile
xmlSecOpenSSLAppKeysMngrAddCertsPath
xmlSecOpenSSLAppShutdown
xmlSecOpenSSLInit
xmlSecOpenSSLShutdown
xmlSecOpenSSLTransformRsaSha1GetKlass
xmlSecOpenSSLTransformSha1GetKlass
xmlSecShutdown
xmlSecTmplReferenceAddTransform
xmlSecTmplSignatureAddReference
xmlSecTmplSignatureCreate
xmlSecTransformEnvelopedGetKlass
xmlSecTransformInclC14NGetKlass

stroke (strongSwan program)

asprintf@GLIBC_2.2.5 (2)
__cxa_atexit@GLIBC_2.2.5 (2)
__fprintf_chk@GLIBC_2.3.4 (4)
free@GLIBC_2.2.5 (2)
fwrite@GLIBC_2.2.5 (2)
getenv@GLIBC_2.2.5 (2)
getopt_long@GLIBC_2.2.5 (2)
getpass@GLIBC_2.2.5 (2)
__libc_start_main@GLIBC_2.2.5 (2)
malloc@GLIBC_2.2.5 (2)
__printf_chk@GLIBC_2.3.4 (4)
realloc@GLIBC_2.2.5 (2)
__stack_chk_fail@GLIBC_2.4 (3)
strcmp@GLIBC_2.2.5 (2)
strcpy@GLIBC_2.2.5 (2)
strlen@GLIBC_2.2.5 (2)
strrchr@GLIBC_2.2.5 (2)
strtol@GLIBC_2.2.5 (2)

stuser (Stealth program)

GLIBC
abort@GLIBC_2.2.5 (4)
chmod@GLIBC_2.2.5 (4)
clearerr@GLIBC_2.2.5 (4)
closedir@GLIBC_2.2.5 (4)
close@GLIBC_2.2.5 (3)
__errno_location@GLIBC_2.2.5 (3)
exit@GLIBC_2.2.5 (4)
fclose@GLIBC_2.2.5 (4)
ferror@GLIBC_2.2.5 (4)
fflush@GLIBC_2.2.5 (4)
fgets@GLIBC_2.2.5 (4)
fopen@GLIBC_2.2.5 (4)
fprintf@GLIBC_2.2.5 (4)
fputs@GLIBC_2.2.5 (4)
fread@GLIBC_2.2.5 (4)
fwrite@GLIBC_2.2.5 (4)
isatty@GLIBC_2.2.5 (4)
__libc_start_main@GLIBC_2.2.5 (4)

lockf@GLIBC_2.2.5 (4)
memcpy@GLIBC_2.14 (5)
memset@GLIBC_2.2.5 (4)
mkdir@GLIBC_2.2.5 (4)
mlock@GLIBC_2.2.5 (4)
munlock@GLIBC_2.2.5 (4)
opendir@GLIBC_2.2.5 (4)
open@GLIBC_2.2.5 (3)
pclose@GLIBC_2.2.5 (4)
popen@GLIBC_2.2.5 (4)
printf@GLIBC_2.2.5 (4)
putchar@GLIBC_2.2.5 (4)
puts@GLIBC_2.2.5 (4)
remove@GLIBC_2.2.5 (4)
rename@GLIBC_2.2.5 (4)
signal@GLIBC_2.2.5 (4)
sprintf@GLIBC_2.2.5 (4)
sscanf@GLIBC_2.2.5 (4)
strcasecmp@GLIBC_2.2.5 (4)
strdup@GLIBC_2.2.5 (4)
strerror@GLIBC_2.2.5 (4)
strlen@GLIBC_2.2.5 (4)
strncmp@GLIBC_2.2.5 (4)
strncpy@GLIBC_2.2.5 (4)
strpbrk@GLIBC_2.2.5 (4)
tcgetattr@GLIBC_2.2.5 (4)
tcsetattr@GLIBC_2.2.5 (4)

libcrypto (OpenSSL)
ERR_get_error@libcrypto.so.10 (2)
ERR_print_errors_cb@libcrypto.so.10 (2)
EVP_aes_256_cbc@libcrypto.so.10 (2)
EVP_CIPHER_CTX_free@libcrypto.so.10 (2)
EVP_CIPHER_CTX_new@libcrypto.so.10 (2)
EVP_cleanup@libcrypto.so.10 (2)
EVP_DecryptFinal_ex@libcrypto.so.10 (2)
EVP_DecryptInit_ex@libcrypto.so.10 (2)
EVP_DecryptUpdate@libcrypto.so.10 (2)
EVP_DigestFinal_ex@libcrypto.so.10 (2)
EVP_DigestInit_ex@libcrypto.so.10 (2)
EVP_DigestUpdate@libcrypto.so.10 (2)
EVP_EncryptFinal_ex@libcrypto.so.10 (2)
EVP_EncryptInit_ex@libcrypto.so.10 (2)
EVP_EncryptUpdate@libcrypto.so.10 (2)
EVP_MD_CTX_create@libcrypto.so.10 (2)
EVP_MD_CTX_destroy@libcrypto.so.10 (2)
EVP_sha256@libcrypto.so.10 (2)
FIPS_mode_set@libcrypto.so.10 (2)
OpenSSL_add_all_digests@libcrypto.so.10 (2)

Glib
g_ascii_strncasecmp
g_base64_decode
g_base64_encode

g_clear_error
g_free
g_log
g_malloc0
g_mutex_clear
g_mutex_init
g_mutex_lock
g_mutex_unlock
g_prefix_error
g_printf
g_propagate_prefixed_error
g_quark_from_static_string
g_return_if_fail_warning
g_set_error
g_slice_alloc0
g_slice_free1
g_spawn_error_quark
g_stpcpy
g_strcmp0
g_strconcat
g_strdup
g_strerror
g_string_free
g_strjoin