Fortra's GoAnywhere Managed File Transfer v6.8 Security Target

Document Version: 1.1

intertek acumen security

2400 Research Blvd
Suite 395
Rockville, MD 20850

Table Of Contents

Revision History

Version	Date	Description
0.1	August 2019	Initial Draft
0.2	December 2019	Updated based on Fortra review
0.3	January 2020	Updated based on Fortra review
0.4	February 2020	Updated based on internal review
0.5	July 2020	Updated based TDs and Fortra review
0.6	October 2020	Updated based on ECR comments and operational testing
0.7	March 15, 2021	TD and other minor updates
0.8	August 3, 2021	Updated TDs and addressed ECR comments
0.9	December 7, 2022	Updated TDs
1.0	December 29, 2022	Final version
1.1	March 29, 2023	Minor update

1 Security Target Introduction

1.1 Security Target and TOE Reference

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

Category	Identifier
ST Title	Fortra’s GoAnywhere Managed File Transfer v6.8 Security Target
ST Version	1.1
ST Date	March 29, 2023
ST Author	Intertek Acumen Security
TOE Identifier	Fortra’s GoAnywhere Managed File Transfer v6.8
TOE Software Version	6.8
TOE Developer	Fortra, LLC
Key Words	Application, Linux, Windows, TLS, SSH, File Transfer

Table 1 TOE/ST Identification

1.2 TOE Overview

The Target of Evaluation (TOE) is the Fortra’s GoAnywhere Managed File Transfer v6.8 (MFT). The TOE is a software application that provides secure file transfer services over HTTPS, TLS, and SSH. GoAnywhere MFT is a secure managed file transfer solution that streamlines the exchange of data between systems, employees, customers, and trading partners. It provides centralized control with extensive security settings, detailed audit trails, and helps process information from files into XML, CSV, and JSON databases.

1.3 TOE Description

1.3.1 Evaluated Configuration

The TOE has been evaluated on the following host platforms:

- CentOS 7 on ESXi 6.7 with Intel Xeon E5-4620v4 (Broadwell)
- Windows Server 2016 on ESXi 6.7 with Intel Xeon E5-4620v4 (Broadwell)

Note: The TOE is the application software only. The host platforms are not part of the evaluation.

The TOE supports (sometimes optionally) secure connectivity with several other IT environment devices as described below.

Environment Component	Required	Usage/Purpose Description
Web Browser	Yes	Remote administration and User file access over HTTPS/TLSv1.2.
Database Server	Yes	MySQL, PostgreSQL, MS SQL Server, Oracle, or DB2/400 for storing settings. The server must support TLSv1.2 to enable secure access by the TOE.
LDAP/AD Server	No	Remote authentication server supporting TLSv1.2.
Mail Server	No	Mail server supporting SMTP over TLSv1.2 for sending notifications.
File Server	No	Remote file server for storing user files: • AS2, AS4, or WebDAV servers supporting HTTPS/TLSv1.2

Environment Component	Required	Usage/Purpose Description
		• SFTP or SCP servers supporting SSHv2 • FTP/s servers supporting TLSv1.2 • Amazon S3 or Azure Blob Storage supporting HTTPS/TLSv1.2 • REST, SOAP, or generic HTTPS/TLSv1.2 server
File Transfer Client	No	Client allowing uses to store and retrieve files from the TOE: • AS2 or AS4 clients supporting HTTPS/TLSv1.2 • SFTP or SCP clients supporting SSHv2 • FTP/s client supporting TLSv1.2
Java Runtime Environment	Yes (on CentOS)	Platform-provided Java SE 8 Java Runtime Environment (JRE). Note: The Windows platform does not provide a JRE, so the Windows version of the TOE includes the required JRE.

Table 2 IT Environment Components

1.3.2 Physical Boundaries

The TOE is a software application running on a host platform (as listed above).

1.3.3 Logical Boundaries

The TOE provides the security functionality required by [SWAPP], [TLS-PKG], and [SSH-EP].

1.3.3.1 Cryptographic Support

The TOE utilizes the GoAnywhere MFT Bouncy Castle FIPS Java API cryptographic library version 1.0.2. This library implements all of the cryptographic algorithms required for SSH and TLS, drawing entropy from the platform RBG.

The cryptographic services provided by the TOE are described below.

Cryptographic Protocol	Use within the TOE
SSHv2 Client	File server transfers using SFTP or SCP
SSHv2 Server	User file transfers using SFTP or SCP
HTTPS/TLSv1.2 Client	File server transfers using AS2, AS4, WebDAV, FTP/s, Amazon S3, Azure Blob Storage, REST, SOAP, or HTTPS; Check for updates
HTTPS/TLSv1.2 Server	HTTPS Remote administration; HTTPS file access; AS2 or AS4 clients
TLSv1.2 Client	Database server; Authentication Server; Mail Server;
TLSv1.2 Server	User file transfers using FTP/s

Table 3 TOE Provided Cryptography

Each of these cryptographic algorithms have been validated for conformance to the requirements specified in their respective standards, as identified below.

SFR	Algorithm in ST	CAVP Alg.	CAVP Cert #
FCS_CKM.1	RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following:	RSA KeyGen (n = 2048, 3072)	C1876

SFR	Algorithm in ST	CAVP Alg.	CAVP Cert #
	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3		
	ECC schemes using "NIST curves" [selection: P-256, P-384, P-521] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4	ECDSA KeyGen ECDSA KeyVer (Curve = P-256, P-384, P-521)	C1876
	FFC Schemes using Diffie-Hellman group 14 that meet the following: RFC 3526, Section 3	NIAP Policy Letter #5, Addendum #2, states "No NIST CAVP, CCTL must perform all assurance/evaluation activities".	Vendor Affirmed.
FCS_CKM.2	RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1"	NIAP Policy Letter #5, Addendum #2, states "No NIST CAVP exists, must be described in TSS – See FIPS 140-2 I.G. D.4: Vendor Affirmation".	Vendor Affirmed.
	Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 2, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"	KAS-ECC (Curve = P-256, P-384, P-521)	C1876
	Key establishment scheme using Diffie-Hellman group 14 that meets the following: RFC 3526, Section 3	NIAP Policy Letter #5, Addendum #2 does not provide any guidance for this selection.	Vendor Affirmed.
FCS_COP.1/ DataEncryption	AES used in [**CBC, GCM**] mode and cryptographic key sizes [**128 bits, 256 bits**]	AES-CBC (128-bit, 256-bit) AES-GCM (128-bit, 256-bit)	C1876
FCS_COP.1/ SigGen	For RSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3	RSA SigGen RSA SigVer (n = 2048, 3072)	C1876
	For ECDSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6 and Appendix D, Implementing "NIST curves" [**P-256, P-384, P-521**]; ISO/IEC 14888-3, Section 6.4	ECDSA SigGen ECDSA SigVer (Curve = P-256, P-384, P-521)	C1876
FCS_COP.1/ Hash	[**SHA-1, SHA-256, SHA-384, SHA-512**] and message digest sizes [**160, 256, 384, 512**] bits	SHA-1 SHA2-256 SHA2-384 SHA2-512	C1876
FCS_COP.1/ KeyedHash	[**HMAC-SHA-1, HMAC-SHA- 256, HMAC-SHA-384, HMAC-SHA-512**] and cryptographic key	HMAC-SHA-1 HMAC-SHA2-256	C1876

SFR	Algorithm in ST	CAVP Alg.	CAVP Cert #
	sizes [**256-bits, 160-bits, 384-bits, 512-bits**] and message digest sizes [**160, 384, 512**] bits	HMAC-SHA2-384 HMAC-SHA2-512	
FCS_RBG_EXT.1	**CTR_DRBG (AES)**	Counter DRBG (AES)	C1876

Table 4 CAVP Algorithm Testing References

1.3.3.2 User Data Protection

The TOE relies on the underlying platform to encrypt sensitive data at rest.

1.3.3.3 Identification and Authentication

The TOE uses X.509v3 certificates as defined by RFC 5280 to authentication the TLS connection to the external TLS servers. The TOE validates the X.509 certificates using the certificate path validation algorithm defined in RFC 5280.

The TOE authenticates users using a username/password combination or X.509 TLS Client Certificates.

1.3.3.4 Security Management

The TOE allows the configuration of users, file servers, file transfer services, keys and certificates, and cryptographic protocols.

1.3.3.5 Privacy

The TOE does not transmit Personally Identifiable Information (PII) over the network.

1.3.3.6 Protection of the TSF

The TOE employs several mechanisms to ensure that it is secure on the host platform. The TOE only allocates a limited amount of memory with both write and execute permission to support just-in-time compiling. The TOE supports ASLR, stack-based overflow protections, and platform security mechanisms (Windows Defender and SELinux).

The TOE is distributed as a Microsoft .EXE file (Windows) or a RPM (CentOS). The installers are signed by Fortra so their integrity can be verified by the platform.

1.3.3.7 Trusted Path/Channels

The TOE protects all data in transit using TLSv1.2 or SSHv2.

1.3.4 TOE Documentation

- Fortra’s GoAnywhere Managed File Transfer v6.8 Security Target, Version 1.1 [ST]
- Fortra’s GoAnywhere Managed File Transfer v6.8 Common Criteria Configuration Guide, Version 1.1 [AGD]

2 Conformance Claims

2.1 CC Conformance

This TOE is conformant to:

- Common Criteria for Information Technology Security Evaluations Part 1, Version 3.1, Revision 5, April 2017
- Common Criteria for Information Technology Security Evaluations Part 2, Version 3.1, Revision 5, April 2017: Part 2 extended
- Common Criteria for Information Technology Security Evaluations Part 3, Version 3.1, Revision 5, April 2017: Part 3 extended

2.2 Protection Profile Conformance

This TOE is conformant to:

- Protection Profile for Application Software, Version 1.3, dated 01 March 2019 [SWAPP]
- Functional Package for Transport Layer Security (TLS), Version 1.1, dated 01 March 2019 [TLS-PKG]
- Extended Package for Secure Shell (SSH), Version 1.0, dated 19 February 2016 [SSH-EP]

2.3 Conformance Rationale

This Security Target provides exact conformance to Version 1.3 of the Protection Profile for Application Software, Version 1.1 of the Functional Package for Transport Layer Security (TLS), and Version 1.0 of the Extended Package for Secure Shell (SSH). The security problem definition and security objectives in this Security Target are taken from the Protection Profile unmodified. The security requirements in this Security Target are all taken from the Protection Profile, Functional Package, and Extended Package performing only operations defined there.

2.3.1 Technical Decisions

All NIAP Technical Decisions (TDs) issued to date that are applicable to [SWAPP], [TLS-PKG], and [SSH-EP] have been considered. The following tables identify all applicable TD:

Identifier	Applicable	Exclusion Rationale (if applicable)
0668 – X.509 SFR Applicability in App PP	Yes	
0601 – X.509 SFR Applicability in App PP	Yes	
0600 – Conformance claim sections updated to allow for MOD_VPNC_V2.3	Yes	
0598 – Expanded AES Modes in FCS_COP for App PP	Yes	
0582 – PP-Configuration for Application Software and Virtual Private Network (VPN) Clients now allowed	Yes	
0561 – Signature verification update	Yes	
0554 – iOS/iPadOS/Android AppSW Virus Scan	No	This TD only applies to iOS and Android platforms. The TOE runs on Linux and Windows platforms.
0548 – Integrity for installation tests in AppSW PP 1.3	Yes	

Identifier	Applicable	Exclusion Rationale (if applicable)
0544 – Alternative testing methods for FPT_AEX_EXT.1.1	Yes	
0543 – FMT_MEC_EXT.1 evaluation activity update	Yes	
0519 – Linux symbolic links and FMT_CFG_EXT.1	Yes	
0515 – Use Android APK manifest in test	No	This TD only applies to Android platforms. The TOE runs on Linux and Windows platforms.
0510 – Obtaining random bytes for iOS/macOS	No	This TD only applies to iOS and macOS platforms. The TOE runs on Linux and Windows platforms.
0498 – Application Software PP Security Objectives and Requirements Rationale	Yes	
0495 – FIA_X509_EXT.1.2 Test Clarification	Yes	
0465 – Configuration Storage for .NET Apps	Yes	
0445 – User Modifiable File Definition	Yes	
0437 – Supported Configuration Mechanism	Yes	
0435 – Alternative to SELinux for FPT_AEX_EXT.1.3	Yes	
0434 – Windows Desktop Applications Test	Yes	
0427 – Reliable Time Source	Yes	
0416 – Correction to FCS_RBG_EXT.1 Test Activity	Yes	

Table 5 SWAPP Technical Decisions

Identifier	Applicable	Exclusion Rationale (if applicable)
0588 – Session Resumption Support in TLS package	Yes	
0513 – CA Certificate loading	Yes	
0499 – Testing with pinned certificates	Yes	
0469 – Modification of test activity for FCS_TLSS_EXT.1.1 test 4.1	Yes	
0442 – Updated TLS Ciphersuites for TLS Package	Yes	

Table 6 TLS-PKG Technical Decisions

Identifier	Applicable	Exclusion Rationale (if applicable)
0598 – Expanded AES Modes in FCS_COP for App PP	Yes	
0446 – Missing selections for SSH	Yes	
0420 – Conflict in FCS_SSHC_EXT.1.1 and FCS_SSHS_EXT.1.1	Yes	
0332 – Support for RSA SHA2 host keys	Yes	
0331 – SSH Rekey Testing	Yes	

Identifier	Applicable	Exclusion Rationale (if applicable)
0240 – FCS_COP.1.1(1) Platform provided crypto for encryption/decryption	Yes	

Table 7 SSH-EP Technical Decisions

3 Security Problem Definition

The security problem definition has been taken from [SWAPP] and is reproduced here for the convenience of the reader. The security problem is described in terms of the threats that the TOE is expected to address, assumptions about the operational environment, and any organizational security policies that the TOE is expected to enforce.

3.1 Threats

The following threats are drawn directly from the [SWAPP].

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

Table 8 Threats

3.2 Assumptions

The following assumptions are drawn directly from the [SWAPP].

ID	Assumption
A.PLATFORM	The TOE relies upon a trustworthy computing platform with a reliable time clock for its execution. This includes the underlying platform and whatever runtime environment it provides to the TOE.
A.PROPER_USER	The user of the application software is not willfully negligent or hostile, and uses the software in compliance with the applied enterprise security policy.
A.PROPER_ADMIN	The administrator of the application software is not careless, willfully negligent or hostile, and administers the software in compliance with the applied enterprise security policy.

Table 9 Assumptions

3.3 Organizational Security Policies

There are no OSPs.

4 Security Objectives

The security objectives have been taken from [SWAPP] and are reproduced here for the convenience of the reader.

4.1 Security Objectives for the TOE

The following security objectives for the TOE were drawn directly from the [SWAPP].

ID	TOE Objective
O.INTEGRITY	Conformant TOEs ensure the integrity of their installation and update packages, and also leverage execution environment-based mitigations. Software is seldom, if ever, shipped without errors. The ability to deploy patches and updates to fielded software with integrity is critical to enterprise network security. Processor manufacturers, compiler developers, execution environment vendors, and operating system vendors have developed execution environment-based mitigations that increase the cost to attackers by adding complexity to the task of compromising systems. Application software can often take advantage of these mechanisms by using APIs provided by the runtime environment or by enabling the mechanism through compiler or linker options. Addressed by: FDP_DEC_EXT.1, FMT_CFG_EXT.1, FPT_AEX_EXT.1, FPT_TUD_EXT.1
O.QUALITY	To ensure quality of implementation, conformant TOEs leverage services and APIs provided by the runtime environment rather than implementing their own versions of these services and APIs. This is especially important for cryptographic services and other complex operations such as file and media parsing. Leveraging this platform behavior relies upon using only documented and supported APIs. Addressed by: FMT_MEC_EXT.1, FPT_API_EXT.1, FPT_API_EXT.2, FPT_LIB_EXT.1, FPT_TUD_EXT.2, FCS_CKM.1(1)
O.MANAGEMENT	To facilitate management by users and the enterprise, conformant TOEs provide consistent and supported interfaces for their security-relevant configuration and maintenance. This includes the deployment of applications and application updates through the use of platform-supported deployment mechanisms and formats, as well as providing mechanisms for configuration. This also includes providing control to the user regarding disclosure of any PII. Addressed by: FMT_SMF.1, FPT_IDV_EXT.1, FPT_TUD_EXT.1, FPR_ANO_EXT.1, FCS_COP.1(3)
O.PROTECTED_STORAGE	To address the issue of loss of confidentiality of user data in the event of loss of physical control of the storage medium, conformant TOEs will use data-at-rest protection. This involves encrypting data and keys stored by the TOE in order to prevent unauthorized access to this data. This also includes unnecessary network communications whose consequence may be the loss of data. Addressed by: FDP_DAR_EXT.1, FCS_STO_EXT.1, FCS_RBG_EXT.1, FCS_CKM.1(3), FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(4)
O.PROTECTED_COMMS	To address both passive (eavesdropping) and active (packet modification) network attack threats, conformant TOEs will use a trusted channel for sensitive data. Sensitive data includes cryptographic keys, passwords, and any other data specific to the application that should not be exposed outside of the application. Addressed by: FTP_DIT_EXT.1, FCS_RBG_EXT.1, FCS_RBG_EXT.2, FCS_CKM_EXT.1, FCS_CKM.2, FCS_HTTPS_EXT.1, FDP_NET_EXT.1, FIA_X509_EXT.1

Table 10 Objectives for the TOE

4.2 Security Objectives for the Operational Environment

The following security objectives for the operational environment assist the TOE in correctly providing its security functionality. These track with the assumptions about the environment.

ID	Objective for the Operation 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.

Table 11 Objectives for the environment

5 Security Requirements

This section identifies the Security Functional Requirements for the TOE and/or Platform. The Security Functional Requirements included in this section are derived from Part 2 of the Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5 and all international interpretations.

Requirement	Description
FCS_RBG_EXT.1	Random Bit Generation Services
FCS_RBG_EXT.2	Random Bit Generation from Application
FCS_CKM_EXT.1	Cryptographic Key Generation Services
FCS_CKM.1(1)	Cryptographic Asymmetric Key Generation
FCS_CKM.2	Cryptographic Key Establishment
FCS_COP.1(1)	Cryptographic Operation - Encryption/Decryption
FCS_COP.1(1)/SSH	Cryptographic Operation - Encryption/Decryption (Refined)
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_HTTPS_EXT.1/Client	HTTPS Protocol
FCS_HTTPS_EXT.1/Server	HTTPS Protocol
FCS_HTTPS_EXT.2	HTTPS Protocol with Mutual Authentication
FCS_STO_EXT.1	Storage of Credentials
FCS_SSH_EXT.1	SSH Protocol
FCS_SSHC_EXT.1	SSH Protocol - Client
FCS_SSHS_EXT.1	SSH Protocol - Server
FCS_TLS_EXT.1	TLS Protocol
FCS_TLSC_EXT.1	TLS Client Protocol
FCS_TLSC_EXT.2	TLS Client Support for Mutual Authentication
FCS_TLSC_EXT.5	TLS Client Support for Supported Groups Extension
FCS_TLSS_EXT.1	TLS Server Protocol
FCS_TLSS_EXT.2	TLS Server Support for Mutual Authentication
FDP_DEC_EXT.1	Access to Platform Resources
FDP_NET_EXT.1	Network Communications
FDP_DAR_EXT.1	Encryption Of Sensitive Application Data
FIA_X509_EXT.1	X.509 Certificate Validation
FIA_X509_EXT.2	X.509 Certificate Authentication
FMT_MEC_EXT.1	Supported Configuration Mechanism
FMT_CFG_EXT.1	Secure by Default Configuration
FMT_SMF.1	Specification of Management Functions
FPR_ANO_EXT.1	User Consent for Transmission of Personally Identifiable Information
FPT_API_EXT.1	Use of Supported Services and APIs

Requirement	Description
FPT_AEX_EXT.1	Anti-Exploitation Capabilities
FPT_TUD_EXT.1	Integrity for Installation and Update
FPT_TUD_EXT.2	Integrity for Installation and Update
FPT_LIB_EXT.1	Use of Third Party Libraries
FPT_IDV_EXT.1	Software Identification and Versions
FTP_DIT_EXT.1	Protection of Data in Transit

Table 12 SFRs

5.1 Conventions

The CC defines operations on Security Functional Requirements: assignments, selections, assignments within selections and refinements. This document uses the following font conventions to identify the operations defined by the CC:

- Assignment: Indicated with *italicized* text;
- Refinement: Indicated with **bold** text;
- Selection: Indicated with underlined text;
- Iteration: Indicated by appending the iteration number in parenthesis, e.g., (1), (2), (3); and/or a slash followed by a short identifier, e.g., /SSH;
- Where operations were completed in the PP itself, the formatting used in the PP has been retained.

Explicitly stated SFRs are identified by having a label 'EXT' after the requirement name for TOE SFRs. Formatting conventions outside of operations matches the formatting specified within the PP.

5.2 Security Functional Requirements

5.2.1 Cryptographic Support (FCS)

FCS_RBG_EXT.1 Random Bit Generation Services

FCS_RBG_EXT.1.1

The application shall [

- *implement DRBG functionality*

] for its cryptographic operations.

FCS_RBG_EXT.2 Random Bit Generation from Application

FCS_RBG_EXT.2.1

The application shall perform all deterministic random bit generation (DRBG) services in accordance with NIST Special Publication 800-90A using [*CTR_DRBG (AES)*]

FCS_RBG_EXT.2.2

The deterministic RBG shall be seeded by an entropy source that accumulates entropy from a platform-based DRBG and [

- *no other noise source*

] with a minimum of [

- *256 bits*

] of entropy at least equal to the greatest security strength (according to NIST SP 800-57) of the keys and hashes that it will generate.

FCS_CKM_EXT.1 Cryptographic Key Generation Services

FCS_CKM_EXT.1.1

The application shall [

- *implement asymmetric key generation*

].

FCS_CKM.1(1) Cryptographic Asymmetric Key Generation

FCS_CKM.1.1(1)

The **application** shall [

- *implement functionality*

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

- ***[RSA schemes]*** *using cryptographic key sizes of* ***[2048-bit or greater]*** *that meet the following* ***FIPS PUB 186-4, "Digital Signature Standard (DSS), Appendix B.3"*** *,*
- ***[ECC schemes]*** *using* ***[“NIST curves” P-256, P-384 and [P-521]]***_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*** *,*

].

FCS_CKM.2 Cryptographic Key Establishment

FCS_CKM.2.1

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

[

- ***[RSA-based key establishment schemes]*** *that meet the following:* ***RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, “Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1” ,***
- ***[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”] ,***
- ***[Key establishment scheme using Diffie-Hellman group 14]*** *that meets the following:* ***RFC 3526, Section 3 ,***

].

FCS_COP.1(1) Cryptographic Operation - Encryption/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,
- AES-GCM (as defined in NIST SP 800-38D) mode,

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

FCS_COP.1(1)/SSH Cryptographic Operation - Encryption/Decryption (Refined)

FCS_COP.1.1(1)

The SSH software shall [*perform*] encryption/decryption services for data in accordance with a specified cryptographic algorithm AES-CTR (as defined in NIST SP 800-38A) mode and cryptographic key sizes [*128-bit, 256-bit*].

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-1,*
- *SHA-256,*
- *SHA-384,*
- *SHA-512,*

] and message digest sizes [

- *160,*
- *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 [P-521] that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 5*

] .

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-1,*
- *SHA-384,*

- *SHA-512,*

] with key sizes [*256-bits, 160-bits, 384-bits, 512-bits*] and message digest sizes 256 and [*160, 384, 512*] bits that meet the following: FIPS Pub 198-1 *The Keyed-Hash Message Authentication Code* and FIPS Pub 180-4 *Secure Hash Standard*.

FCS_HTTPS_EXT.1/Client HTTPS Protocol

FCS_HTTPS_EXT.1.1/Client

The application shall implement the HTTPS protocol that complies with RFC 2818.

FCS_HTTPS_EXT.1.2/Client

The application shall implement HTTPS using TLS as defined in the TLS package.

FCS_HTTPS_EXT.1.3/Client

The application shall [*not establish the application-initiated connection*] if the peer certificate is deemed invalid.

FCS_HTTPS_EXT.1/Server HTTPS Protocol

FCS_HTTPS_EXT.1.1/Server

The application shall implement the HTTPS protocol that complies with RFC 2818.

FCS_HTTPS_EXT.1.2/Server

The application shall implement HTTPS using TLS as defined in the TLS package.

FCS_HTTPS_EXT.2 HTTPS Protocol with Mutual Authentication

FCS_HTTPS_EXT.2.1

The application shall [*not establish the connection*] if the peer certificate is deemed invalid.

FCS_STO_EXT.1 Storage of Credentials

FCS_STO_EXT.1.1

The application shall [

- *not store any credentials,*

] to non-volatile memory.

FCS_SSH_EXT.1 SSH Protocol

FCS_SSH_EXT.1.1

The SSH software shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, 4254 and [*5656, 6668*] as a [*client, server*]

FCS_SSHC_EXT.1 SSH Protocol - Client

FCS_SSHC_EXT.1.1

The SSH client shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, and [*password-based*] .

FCS_SSHC_EXT.1.2

The SSH client shall ensure that, as described in RFC 4253, packets greater than [*65,535*] bytes in an SSH transport connection are dropped.

FCS_SSHC_EXT.1.3

The SSH software shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: aes128-ctr, aes256-ctr, [*aes128-cbc, aes256-cbc*].

FCS_SSHC_EXT.1.4

The SSH client shall ensure that the SSH transport implementation uses [*ssh-rsa, rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256*] and [*ecdsa-sha2-nistp384*] as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SSHC_EXT.1.5

The SSH client shall ensure that the SSH transport implementation uses [*hmac-sha1, hmac-sha1-96, hmac-sha2-256*] and [*no other MAC algorithms*] as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHC_EXT.1.6

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

FCS_SSHC_EXT.1.7

The SSH server shall ensure that the SSH connection be rekeyed after [*no more than 1 Gigabyte of data has been transmitted, no more than 1 hour*] using that key.

FCS_SSHC_EXT.1.8

The SSH client shall ensure that the SSH client authenticates the identity of the SSH server using a local database associating each host name with its corresponding public key or [*no other methods*] as described in RFC 4251 section 4.1.

FCS_SSHS_EXT.1 SSH Protocol - Server

FCS_SSHS_EXT.1.1

The SSH server shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, and [*password-based*] .

FCS_SSHS_EXT.1.2

The SSH server shall ensure that, as described in RFC 4253, packets greater than [*65,535*] bytes in an SSH transport connection are dropped.

FCS_SSHS_EXT.1.3

The SSH server shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: aes128-ctr, aes256-ctr, [*aes128-cbc, aes256-cbc*] .

FCS_SSHS_EXT.1.4

The SSH server shall ensure that the SSH transport implementation uses [*ssh-rsa, rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256*] and [*ecdsa-sha2-nistp384*] as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SSHS_EXT.1.5

The SSH server shall ensure that the SSH transport implementation uses [*hmac-sha1, hmac-sha2-256, hmac-sha2-512*] and [*no other MAC algorithms*] as its MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHS_EXT.1.6

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

FCS_SSHS_EXT.1.7

The SSH server shall ensure that the SSH connection be rekeyed after [*no more than 1 Gigabyte of data has been transmitted, no more than 1 hour*] using that key.

FCS_TLS_EXT.1 TLS Protocol

FCS_TLS_EXT.1.1

The product shall implement [

- *TLS as a client,*
- *TLS as a server,*

].

FCS_TLSC_EXT.1 TLS Client Protocol

FCS_TLSC_EXT.1.1

The product shall implement TLS 1.2 (RFC 5246) and [*no earlier TLS versions*] as a client that supports the cipher suites [

- *TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246,*
- *TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 5246,*
- *TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,*
- *TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246,*
- *TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288,*
- *TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288,*
- *TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,*
- *TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,*
- *TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,*
- *TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,*
- *TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,*
- *TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,*

] and also supports functionality for [

- *mutual authentication,*

].

FCS_TLSC_EXT.1.2

The product shall verify that the presented identifier matches the reference identifier according to RFC 6125.

FCS_TLSC_EXT.1.3

The product shall not establish a trusted channel if the server certificate is invalid [

- *with no exceptions,*

].

FCS_TLSC_EXT.2 TLS Client Support for Mutual Authentication

FCS_TLSC_EXT.2.1

The product shall support mutual authentication using X.509v3 certificates.

FCS_TLSC_EXT.5 TLS Client Support for Supported Groups Extension

FCS_TLSC_EXT.5.1

The product shall present the Supported Groups Extension in the Client Hello with the supported groups [

- *secp256r1,*
- *secp384r1,*

].

FCS_TLSS_EXT.1 TLS Server Protocol

FCS_TLSS_EXT.1.1

The product shall implement TLS 1.2 (RFC 5246) and [no earlier TLS versions] as a server that supports the cipher suites [

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246,
- TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 5246,
- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,
- TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246,
- TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288,
- TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288,
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,

] and also supports functionality for [

- mutual authentication,
- no session resumption or session tickets,

].

FCS_TLSS_EXT.1.2

The product shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0 and [*TLS 1.1*].

FCS_TLSS_EXT.1.3

The product shall perform key establishment for TLS using [

- *RSA with size [2048 bits, 3072 bits, 4096 bits] ,*
- *ECDHE parameters using elliptic curves [secp256r1, secp384r1] and no other curves ,*
- *no other key establishment methods*

].

FCS_TLSS_EXT.2 TLS Server Support for Mutual Authentication

FCS_TLSS_EXT.2.1

The product shall support authentication of TLS clients using X.509v3 certificates.

FCS_TLSS_EXT.2.2

The product shall not establish a trusted channel if the client certificate is invalid.

FCS_TLSS_EXT.2.3

The product shall not establish a trusted channel if the Distinguished Name (DN) or Subject Alternative Name (SAN) contained in a certificate does not match one of the expected identifiers for the client.

5.2.2 User Data Protection (FDP)

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,*

] .

FDP_NET_EXT.1 Network Communications

FDP_NET_EXT.1.1

The application shall restrict network communication to [

- *user-initiated communication for [check for updates],*
- *respond to [remote management, user file transfers],*
- *[authentication server, database server, file server, mail server]*

] .

FDP_DAR_EXT.1 Encryption Of Sensitive Application Data

FDP_DAR_EXT.1.1

The application shall [

- *leverage platform-provided functionality to encrypt sensitive data,*

] in non-volatile memory.

5.2.3 Identification and Authentication (FIA)

FIA_X509_EXT.1 X.509 Certificate Validation

FIA_X509_EXT.1.1

The application shall [implement functionality] to validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation
- The certificate path must terminate with a trusted CA certificate
- The application shall validate a certificate path by ensuring the presence of the basicConstraints extension, that the CA flag is set to TRUE for all CA certificates, and that any path constraints are met
- The application shall validate that any CA certificate includes caSigning purpose in the key usage field
- The application shall validate the revocation status of the certificate using [CRL as specified in RFC 5280 Section 6.3]
- The application shall validate the extendedKeyUsage (EKU) field according to the following rules:
    - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
    - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the EKU field.
    - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the EKU field.
    - S/MIME certificates presented for email encryption and signature shall have the Email Protection purpose (id-kp 4 with OID 1.3.6.1.5.5.7.3.4) in the EKU field.
    - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the EKU field.
    - Server certificates presented for EST shall have the CMC Registration Authority (RA) purpose (id-kp-cmcRA with OID 1.3.6.1.5.5.7.3.28) in the EKU field.

FIA_X509_EXT.1.2

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

FIA_X509_EXT.2 X.509 Certificate Authentication

FIA_X509_EXT.2.1

The application shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [*HTTPS ,TLS*].

FIA_X509_EXT.2.2

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

5.2.4 Security Management (FMT)

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.].

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 binaries and data files from modification by normal unprivileged users.

FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1

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

- *[configure users,*
- *configure database server,*
- *configure authentication server,*
- *configure mail server.*
- *configure file servers,*
- *configure file transfer services,*
- *configure keys and certificates,*
- *configure cryptographic protocols]*

] .

5.2.5 Privacy (FPR)

FPR_ANO_EXT.1 User Consent for Transmission of Personally Identifiable Information

FPR_ANO_EXT.1

The application shall [

- *not transmit PII over a network ,*

] .

5.2.6 Protection of TSF (FPT)

FPT_API_EXT.1 Use of Supported Services and APIs

FPT_API_EXT.1.1

The application shall use only documented platform APIs.

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 ,*
- *allocate memory regions with write and execute permissions for only [JVM bytecode to machine code just-in-time compilation]*

] .

Application Note: On Linux, the TOE does not allocate any memory regions with both write and execute permissions. On Windows, the TOE allocates memory regions with 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_TUD_EXT.1 Integrity for Installation and Update

FPT_TUD_EXT.1.1

The application shall [*provide the ability*] to check for updates and patches to the application software.

FPT_TUD_EXT.1.2

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

FPT_TUD_EXT.1.3

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

FPT_TUD_EXT.1.4

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

FPT_TUD_EXT.1.5

The application is distributed [*as an additional software package to the platform OS*]

FPT_TUD_EXT.2 Integrity for Installation and Update

FPT_TUD_EXT.2.1

The application shall be distributed using the format of the platform-supported package manager.

FPT_TUD_EXT.2.2

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

FPT_TUD_EXT.2.3

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

FPT_LIB_EXT.1 Use of Third Party Libraries

FPT_LIB_EXT.1.1

The application shall be packaged with only [*the third-party libraries listed in section 6.1*].

FPT_IDV_EXT.1 Software Identification and Versions

FPT_IDV_EXT.1.1

The application shall be versioned with [*SWID tags that comply with minimum requirements from ISO/IEC 19770-2:2015*] .

5.2.7 Trusted Path/Channel (FTP)

FTP_DIT_EXT.1 Protection of Data in Transit

FTP_DIT_EXT.1.1

The application shall [

- *encrypt all transmitted [data] with [HTTPS as a client in accordance with FCS_HTTPS_EXT.1/Client, HTTPS as a server in accordance with FCS_HTTPS_EXT.1/Server, HTTPS as a server with mutual authentication in accordance with FCS_HTTPS_EXT.2, TLS as defined in the TLS Package, SSH as conforming to the Extended Package for Secure Shell]*

] between itself and another trusted IT product.

Application Note: The cryptographic protocols are mapped to trusted IT products as follows:

- HTTPS/TLSv1.2 Web Server with or without TLS client authentication – Remote Administration
- TLSv1.2 client with or without TLS client authentication – Database server
- TLSv1.2 client with or without TLS client authentication – LDAP/AD server
- TLSv1.2 client without TLS client authentication – Mail server
- HTTPS/TLSv1.2 client with or without TLS client authentication – AS2, AS4, or WebDAV file servers
- SSHv2 client – SFTP or SCP file servers
- TLSv1.2 client – FTP/s file servers
- HTTPS/TLSv1.2 client – Amazon S3 or Azure Blob Storage servers
- HTTPS/TLSv1.2 client – REST, SOAP, or generic HTTPS servers
- HTTPS/TLSv1.2 server – AS2 or AS4 clients
- SSHv2 server – SFTP or SCP clients
- TLSv1.2 server – FTP/s clients

5.3 TOE SFR Dependencies Rationale for SFRs

The Protection Profile for Application Software contains all the requirements claimed in this Security Target. As such, the dependencies are not applicable since the PP has been approved.

5.4 Security Assurance Requirements

The TOE assurance requirements for this ST are taken directly from the Protection Profile for Application Software which are derived from Common Criteria Version 3.1, Revision 5. The assurance requirements are summarized in the table below.

Assurance Class	Components	Components Description
Development	ADV_FSP.1	Basic functional specification
Guidance documents	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative procedures
Life-cycle support	ALC_CMC.1	Labeling of the TOE
	ALC_CMS.1	TOE CM coverage
	ALC_TSU_EXT.1	Timely Security Updates
Security Target evaluation	ASE_CCL.1	Conformance claims
	ASE_ECD.1	Extended components definition
	ASE_INT.1	ST introduction
	ASE_OBJ.1	Security objectives for the operational environment
	ASE_REQ.1	Stated security requirements
	ASE_TSS.1	TOE summary specification
Tests	ATE_IND.1	Independent testing – conformance
Vulnerability assessment	AVA_VAN.1	Vulnerability survey

Table 13 Security Assurance Requirements

5.5 Assurance Measures

The TOE satisfies the identified assurance requirements. This section identifies the Assurance Measures applied by Fortra to satisfy the assurance requirements. The table below lists the details.

SAR	How the SAR will be met
ADV_FSP.1	The functional specification describes the external interfaces of the TOE; such as the means for a user to invoke a service and the corresponding response of those services. The description includes the interface(s) that enforces a security functional requirement, the interface(s) that supports the enforcement of a security functional requirement, and the interface(s) that does not enforce any security functional requirements. The interfaces are described in terms of their purpose (general goal of the interface), method of use (how the interface is to be used), parameters (explicit inputs to and outputs from an interface that control the behavior of that interface), parameter descriptions (tells what the parameter is in some meaningful way), and error messages (identifies the condition that generated it, what the message is, and the meaning of any error codes).

SAR	How the SAR will be met
AGD_OPE.1	The Administrative Guide provides the descriptions of the processes and procedures of how the administrative users of the TOE can securely administer the TOE using the interfaces that provide the features and functions detailed in the guidance.
AGD_PRE.1	The Installation Guide describes the installation, generation, and startup procedures so that the users of the TOE can put the components of the TOE in the evaluated configuration.
ALC_CMC.1	The Configuration Management (CM) documents describe how the consumer identifies the evaluated TOE. The CM documents identify the configuration items, how those configuration items are uniquely identified, and the adequacy of the procedures that are used to control and track changes that are made to the TOE. This includes details on what changes are tracked and how potential changes are incorporated.
ALC_CMS.1	
ALC_TSU_EXT.1	Fortra uses a systematic method for identifying and providing security relevant updates to the TOEs users via its support infrastructure.
ATE_IND.1	Fortra will provide the TOE for testing.
AVA_VAN.1	Fortra will provide the TOE for testing.

Table 14 TOE Security Assurance Measures

6 TOE Summary Specification

This chapter identifies and describes how the Security Functional Requirements identified above are met by the TOE.

SFR	Rationale
FCS_RBG_EXT.1	The TOE implements DRBG functionality.
FCS_RBG_EXT.2	The TOE provides random bit generation services using an SP 800-90A CTR_DRBG using AES-256. The TOE DRBG is seeded with at least 256 bits of entropy from the platform DRBG, CryptGenRandom on Windows and /dev/random on CentOS. See Table 4 for validation details.
FCS_CKM_EXT.1	The TSF implements asymmetric key generation as described in FCS_CKM.1(1).
FCS_CKM.1(1)	The TSF generates RSA 2048-bit, and 3072-bit keys following the algorithm specified in Sections 5.1 and B.3.3 of FIPS Pub 186-4. These keys are used for digital signature and key agreement services in TLS or digital signature services in SSH. The TSF generates ECDSA P-256, P-384, and P-521 keys following the algorithm specified in Sections 6.2.1 and B.4.2 of FIPS Pub 186-4. P-256 and P-384 keys are used for key agreement services in TLS and digital signature service in SSH. P-256, P-384, and P-521 keys are used for digital signature services in TLS and key agreement services in SSH. The TSF also generates Diffie-Hellman Group 14 keys as specified in RFC 3526 Section 3. These keys are used for key agreement services in SSH. See Table 4 for validation details.
FCS_CKM.2	The TSF performs RSAES-PKCS1-v1_5 key transport with 2048-bit, 3072-bit, and 4096-bit keys in TLS. The TSF performs Elliptic Curve Diffie-Hellman with curves P-256 and P-384 in TLS and SSH. SSH also supports P-521. The TSF performs Diffie-Helman with Group 14 in SSH. See Table 4 for validation details.
FCS_COP.1(1) FCS_COP.1(1)/SSH	The TSF provides symmetric encryption and decryption capabilities using AES in CBC, CTR, and GCM mode. 128-bit and 256-bits keys are supported in all modes. Encryption and decryption are implemented as described in NIST 197, NIST SP 800-38A, and NISP SP 800-38D. See Table 4 for validation details.
FCS_COP.1(2)	The TSF provides cryptographic hashing services using SHA-1, SHA-256, SHA-384, and SHA-512 as specified in FIPS Pub 180-4 "Secure Hash Standard." The TSF uses all of the hashes for RSA SigGen & SigVer, ECDSA SigGen & SigVer, HMAC, and SSH KDF; with the exception SHA-1 which is not used with ECDSA SigGen & SigVer. See Table 4 for validation details.
FCS_COP.1(3)	The TSF provides cryptographic signature services using RSA with key size of 2048 and greater or ECDSA using curves P-256, P-384, and P-521 as specified in FIPS PUB 186-4, "Digital Signature Standard". See Table 4 for validation details.
FCS_COP.1(4)	The TSF implements the following HMAC algorithms as specified in FIPS Pub 198-1, "The Keyed-Hash Message Authentication Code": **HMAC-SHA-1** **HMAC-SHA-256** **HMAC-SHA-384** **HMAC-SHA-512**

<table>
<tr><th>SFR</th><th>Rationale</th></tr>
<tr><td></td><td><table><tr><td>Key Length</td><td>160 bits</td><td>256 bits</td><td>384 bits</td><td>512 bits</td></tr><tr><td>Output MAC</td><td>96, 160 bits</td><td>256 bits</td><td>384 bits</td><td>512 bits</td></tr></table>See Table 4 for validation details.</td></tr>
<tr><td>FCS_HTTPS_EXT.1/Client
FCS_HTTPS_EXT.1/Server
FCS_HTTPS_EXT.2</td><td>The TSF implements HTTPS as specified in RFC 2818 using TLSv1.2 as the secure transport protocol. TLSv1.2 is specified in FCS_TLSC_EXT.1 and FCS_TLSS_EXT.1.
When acting as an HTTPS server, the TSF supports TLS client authentication for remote administration, HTTPS file access, and AS2 connections. If the client does not present a certificate or the certificate is not authorized, the TSF falls back to application layer authentication (e.g., username and password).
The TSF does not establish the connection (client or server) if the peer certificate is invalid.</td></tr>
<tr><td>FCS_STO_EXT.1</td><td>The TSF does not store any credentials in non-volatile memory. The TSF requires the environment to pass it the database password at start-up.
All other sensitive data (e.g., user passwords, private keys) is stored in the remote database.</td></tr>
<tr><td>FCS_SSH_EXT.1</td><td>The TSF acts as an SSH Client and an SSH server as specified in FCS_SSHC_EXT.1 and FCS_SSHS_EXT.1.</td></tr>
<tr><td>FCS_SSHC_EXT.1</td><td>The TSF is an SSHv2 client, compliant with RFCs 4251, 4252, 4253, 4254, 5656, and 6668.
The TSF supports public-key and password-based authentication.
The TSF examines packet_length field in the SSH packet header. The TSF discards any packets larger than 65,535 bytes.
The TSF supports the following encryption algorithms: aes128-ctr, aes256-ctr, aes128-cbc, and aes256-cbc.
The TSF supports the following public-key algorithms: ssh-rsa, rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256, and ecdsa-sha2-nistp384.
The TSF supports the following data integrity algorithms: hmac-sha1, hmac-sha1-96, hmac-sha2-256, and hmac-sha2-512.
The TSF supports the following key exchange methods: diffie-hellman-group14-sha1, ecdh-sha2-nistp256, ecdh-sha2-nistp384, and ecdh-sha2-nistp521.
The TSF tracks the number of bytes encrypted with each key and the time since the last rekey. The TSF initiates a rekey if 1 GB of data is encrypted with an individual key or when 1 hour has elapsed since the last rekey.
The TSF maintains a local database associating each server hostname or IP address with a trusted public key.</td></tr>
<tr><td>FCS_SSHS_EXT.1</td><td>The TSF is an SSHv2 sever, compliant with RFCs 4251, 4252, 4253, 4254, 5656, and 6668.
The TSF supports public-key and password-based authentication.
The TSF examines packet_length field in the SSH packet header. The TSF discards any packets larger than 65,535 bytes.
The TSF supports the following encryption algorithms: aes128-ctr, aes256-ctr, aes128-cbc, and aes256-cbc.
The TSF supports the following public-key algorithms: ssh-rsa, rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256, and ecdsa-sha2-nistp384.</td></tr>
</table>

SFR	Rationale
	The TSF supports the following data integrity algorithms: hmac-sha1, hmac-sha2-256, and hmac-sha2-512. The TSF supports the following key exchange methods: diffie-hellman-group14-sha1, ecdh-sha2-nistp256, ecdh-sha2-nistp384, and ecdh-sha2-nistp521. The TSF tracks the number of bytes encrypted with each key and the time since the last rekey. The TSF initiates a rekey if 1 GB of data is encrypted with an individual key or when 1 hour has elapsed since the last rekey.
FCS_TLS_EXT.1	The TSF acts as a TLS Client and a TLS server as specified in FCS_TLSC_EXT.1 and FCS_TLSS_EXT.1.
FCS_TLSC_EXT.1	The TSF is a TLSv1.2 client supporting the following ciphersuites: • TLS_RSA_WITH_AES_128_CBC_SHA • TLS_RSA_WITH_AES_256_CBC_SHA • TLS_RSA_WITH_AES_128_CBC_SHA256 • TLS_RSA_WITH_AES_256_CBC_SHA256 • TLS_RSA_WITH_AES_128_GCM_SHA256 • TLS_RSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 • TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 • TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 The TSF automatically configures references identifiers based on the FQDN or IP address configured used to specify the TLS server. When an FQDN has been configured, the TOE establishes reference identifiers of DNS-ID and CN-ID. When the TOE compares the reference identifies to the identifiers in the presented certificate, it will consider the identifiers matching if they are an exact match or if the presented identifier exactly matches with the exception of a wildcard in the left most position matching the left most position of the reference identifier. When an IP address has been configured, the TOE converts the IP address into a binary representation in network byte order. The TOE converts the presented identifier from the SAN and converts it to a binary representation in network byte order. The identifiers are only considered a match if the binary representation is an exact match. For FQDN identifiers, the TOE will use the SAN(s) in the presented certificate if present. The TSF will only match an FQDN identifier from the CN if the certificate does not contain any SANs. When an IP address is configured, the TOE only checks the SAN for a match. The TSF does not support certificate pinning. The TSF will not establish a trusted channel if the TLS server presents an invalid certificate.
FCS_TLSC_EXT.2	The TSF supports TLS mutual authentication for FTP/s, AS2, and HTTPS connections. The TSF is capable of presenting a certificate in response to a CertificateRequest message from the server. The TSF only sends a certificate if the administrator has configured a client certificate for the specific server in question.
FCS_TLSC_EXT.5	The TSF presents the Supported Groups (formerly named Supported Elliptic Curves) extension in the ClientHello message with support for groups secp256r1 and secp384r1.
FCS_TLSS_EXT.1	The TSF is a TLSv1.2 server supporting the following ciphersuites:

SFR	Rationale
	• TLS_RSA_WITH_AES_128_CBC_SHA • TLS_RSA_WITH_AES_256_CBC_SHA • TLS_RSA_WITH_AES_128_CBC_SHA256 • TLS_RSA_WITH_AES_256_CBC_SHA256 • TLS_RSA_WITH_AES_128_GCM_SHA256 • TLS_RSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 • TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 • TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 The TSF sends a Fatal protocol_version alert message if it receives a ClientHello requesting SSLv2.0, SSLv3.0, TLSv1.0, or TLSv1.1. The TSF supports RSA key agreement with RSA key sizes of 2048 bits, 3072 bits, or 4096 bits (i.e., the size of the RSA key in the TLS server certificate). The TSF supports ECDHE key agreement using curves P-256 and P-384.
FCS_TLSS_EXT.2	The TSF can be configured to request a client certificate when establishing TLS connections for remote administration, HTTPS file access, AS2, and FTP/s. When the TSF is configured to authenticate TLS clients using certificates, the TSF sends a Certificate Request message in the TLS handshake. The TSF requires the DN in the presented certificate to match a DN that has been configured as authorized for the services the client is connecting to. The TSF then verifies the certificate matches the certificate pinned to the user's account using a SHA-1 hash and validates the certificate chain as described in FIA_X509_EXT.1. If the client does not present a certificate or the certificate authentication fails, the TSF falls back to username/password authentication. If certificate authentication is required for a user, the username/password prompt will be presented; however, username/password authentication will always deny access to that user.
FDP_DEC_EXT.1	The only hardware resource the TSF access is network connectivity. The only sensitive information repository the TSF access is the system logs. The TSF writes application events to the system logs.
FDP_NET_EXT.1	The TOE performs the following network communication: • user-initiated communication: o check for updates • respond to: o remote management o user file transfers • automatically initiate communications with: o authentication server o database server o file server o mail server
FDP_DAR_EXT.1	The TSF relies on platform provided functionality to encrypt data at rest. The TSF requires the user to enable full drive encryption.

SFR	Rationale
FIA_X509_EXT.1	The TSF uses X.509v3 certificates as defined by RFC 5280 to support authentication for TLS connections. The TSF performs certificate validation during the TLS handshake when the non-TOE entity presents a certificate to the TSF. The X.509 certificates are validated using the certificate path validation algorithm defined in RFC 5280, which can be summarized as follows: • the public key algorithm and parameters are checked • the current date/time is checked against the validity period • revocation status is checked • issuer name of X matches the subject name of X+1 • name constraints are checked • policy OIDs are checked • policy constraints are checked; issuers are ensured to have CA signing bits • path length is checked The TSF performs revocation checking using CRLs.
FIA_X509_EXT.2	During the TLS handshake, the TSF uses the certificate presented by the TLS client or server to authenticate the remote endpoint of the connection. If the TSF cannot establish a connection to fetch a CRL, the TSF considers the certificate invalid and rejects the certificate.
FMT_MEC_EXT.1	The TOE stores settings and configuration options in C:\ProgramData for Windows and /etc for CentOS.
FMT_CFG_EXT.1	The TOE is not installed with any default credential. The TOE generates a self-signed HTTPS server certificate as part of the installation process and requires the user create a username and password prior to performing any other TOE operations. The TOE is installed with default file permissions which prevent unprivileged users from modifying application binaries and application configurations.
FMT_SMF.1	The TSF supports the following management functions: • configure users, • configure database server, • configure authentication server, • configure mail server. • configure file servers, • configure file transfer services, • configure keys and certificates, • configure cryptographic protocols – ciphers
FPR_ANO_EXT.1	The TOE does not transmit PII over the network. PII is defined as name, social security number, date and place of birth, mother's maiden name, biometric records, etc. While the TOE does transmit usernames these are pseudonyms that do not necessarily map to a specific individual. The TOE transmits user files which may incidentally contain PII; however, these are only generic files from the perspective of the TOE.
FPT_API_EXT.1	The TOE uses the Windows platform APIs listed in Section 6.1. The TOE uses the Linux (CentOS) platform APIs listed in Section 6.2.
FPT_AEX_EXT.1	**Windows** The TOE does not request that any memory is mapped to an explicit address. The TOE is compiled without any specific flags on Windows to enable ASLR (/DYNAMICBASE is enabled by default).

SFR	Rationale
	The TOE does not allocate memory with write and execute permissions, with the exception of the JVM code cache. The JRE uses this memory to perform just-in-time compilation of Java bytecode into machine code. The TOE is compatible with Windows Defender Exploit Guard. The TOE does not write any data or user files to the directory containing TOE executable files. The TSF is composed of Java and native code. The native code implements stack-based buffer overflow protections, being compiled with the /GS flag on Windows. All Java objects are strictly typed with explicit sizes, so it is not possible to overflow a buffer in Java code. **Linux** The TOE does not request that any memory is mapped to an explicit address. The TOE is composed of Java code. ASLR is provided by the platform-provided JRE. The TOE does not allocate memory with write and execute permissions. The TOE is compatible with SELinux. The TOE does not write any data or user files to the directory containing TOE executable files. The TSF is composed of Java code. All Java objects are strictly typed with explicit sizes, so it is not possible to overflow a buffer in Java code.
FPT_TUD_EXT.1 FPT_TUD_EXT.2	The TSF allows the user to initiate a check for updates by navigating to 'Help' -> 'Check for Updates'. The TSF allows users to query the current version of TSF software by opening the 'About' tab in the application. The TOE is distributed in the format of the platform-supported package manager, .EXE for Windows and .RPM for CentOS. Updates to the TOE are digitally signed and verified by the platform (Windows Installer or RPM Package manager) prior to installation. Updates for Windows are signed using a code signing cert issued by DigiCert SHA2 Assured ID Code Signing CA. The DigiCert SHA2 Assured ID Code Signing CA is signed by the DigiCert Assured ID Root CA which is trusted by the Windows platform. Updates for Linux are signed using an OpenPGP RSA 2048 key. Adding the public key to the rpm keyring allows the platform to verify the signature on updates. The signature verification on each platform allows the platform to verify the update came from an authorized source. The TSF does not update itself, but rather relies on the platform package manager to install updates. The TOE is not distributed with the platform OS. When the uninstallation procedures are performed, all traces of the TOE are removed, with the exception of configuration settings, output files, and audit events.
FPT_LIB_EXT.1	The TOE is bundled with the third-party libraries listed in section 6.1.
FPT_IDV_EXT.1	The TSF is installed with a SWID tag containing a SoftwareIdentity element and an Entity element.
FTP_DIT_EXT.1	The TSF encrypts all transmitted data using HTTPS/TLSv1.2, TLSv1.2, or SSHv2 as follows: • HTTPS/TLSv1.2 Web Server with or without TLS client authentication – Remote Administration • TLSv1.2 client with or without TLS client authentication – Database server • TLSv1.2 client with or without TLS client authentication – LDAP/AD server • TLSv1.2 client without TLS client authentication – Mail server

SFR	Rationale
	- HTTPS/TLSv1.2 client with or without TLS client authentication – AS2, AS4, or WebDAV file servers - SSHv2 client – SFTP or SCP file servers - TLSv1.2 client – FTP/s file servers - HTTPS/TLSv1.2 client – Amazon S3 or Azure Blob Storage servers - HTTPS/TLSv1.2 client – REST, SOAP, or generic HTTPS servers - HTTPS/TLSv1.2 server – AS2 or AS4 clients - SSHv2 server – SFTP or SCP clients - TLSv1.2 server – FTP/s clients
ALC_TSU_EXT.1	Fortra uses various security tools to regularly scan the TOE throughout the development lifecycle. Vulnerability reports are submitted using a form on the Fortra website which is protected using HTTPS. This protects the confidentiality of the vulnerability report. GoAnywhere Support and Development Teams collaborate to evaluate any reports of application vulnerabilities received. The teams work to understand the issue, understand the impact, and evaluate potential courses of action. After confirming and understanding the issue, the GoAnywhere team prepares the appropriate remediation for the problem. Fortra strives to meet the following timelines for addressing vulnerabilities: - Zero-day and Critical Vulnerabilities: within a week - High Vulnerabilities: within 30 days - Medium and Low Vulnerabilities: 3-4 months - If the vulnerability is in a third-party library, then Fortra must wait for the library developers to address the issue, but Fortra will provide mitigation recommendations to minimize potential risk. Notifications are sent out in the form of a Security Advisory email. This notification contains information describing the issue and remediation options. Depending on the nature and severity of the issue, certain details will potentially be omitted from this publication in order to give our customers enough time to apply the fix.

Table 15 TOE Summary Specification SFR Description

6.1 Windows APIs

The TOE uses the following Windows platform APIs:

- Locale.nls
- Kernel32.dll.mui
- SortDefault.nls
- comctl32.dll
- wsock32.dll
- ws2_32.dll
- winmr.dll
- winmmbase.dll
- winmm.dll
- windows.storage.dll
- win32u.dll
- version.dll
- user32.dll
- ucrtbase.dll
- shlwapi.dll
- shell32.dll
- SHCore.dll
- sechost.dll
- rsaenh.dll
- rpcrt4.dll
- rasadhlp.dll
- psapi.dll

- profapi.dll
- powrprof.dll
- oleaut32.dll
- ole32.dll
- ntdll.dll
- ntasn1.dll
- nsi.dll
- nlaapi.dll
- ncrypt.dll
- NapiNSP.dll
- mswsock.dll
- msvcrt.dll
- msasn1.dll
- KernelBase.dll
- kernel32.dll
- kernel.appcore.dll
- IPHLPAPI.DLL
- gdi32full.dll
- gdi32.dll
- FWPUCLNT.DLL
- dpapi.dll
- dnsapi.dll
- dhcpcsvc6.dll
- dhcpcsvc.dll
- cryptsp.dll
- cryptbase.dll
- crypt32.dll
- combase.dll
- cfgmgr32.dll
- bcryptprimitives.dll
- bcrypt.dll
- apphelp.dll
- advapi32.dll

6.2 Linux APIs

The TOE uses the following Linux (CentOS) platform APIs:

- /dev/null
- /dev/random
- /dev/urandom
- /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.282.b08-1.el7_9.x86_64/jre/bin/java
- /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.282.b08-1.el7_9.x86_64/jre/lib/amd64/jli/libjli.so
- /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.282.b08-1.el7_9.x86_64/jre/lib/amd64/libjava.so
- /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.282.b08-1.el7_9.x86_64/jre/lib/amd64/libmanagement.so
- /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.282.b08-1.el7_9.x86_64/jre/lib/amd64/libnet.so
- /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.282.b08-1.el7_9.x86_64/jre/lib/amd64/libnio.so
- /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.282.b08-1.el7_9.x86_64/jre/lib/amd64/libsunec.so
- /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.282.b08-1.el7_9.x86_64/jre/lib/amd64/libverify.so
- /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.282.b08-1.el7_9.x86_64/jre/lib/amd64/libzip.so
- /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.282.b08-1.el7_9.x86_64/jre/lib/amd64/server/libjvm.so
- /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.282.b08-1.el7_9.x86_64/jre/lib/charsets.jar
- /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.282.b08-1.el7_9.x86_64/jre/lib/ext/cldrdata.jar
- /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.282.b08-1.el7_9.x86_64/jre/lib/ext/localedata.jar
- /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.282.b08-1.el7_9.x86_64/jre/lib/ext/nashorn.jar
- /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.282.b08-1.el7_9.x86_64/jre/lib/ext/sunec.jar
- /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.282.b08-1.el7_9.x86_64/jre/lib/ext/sunjce_provider.jar
- /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.282.b08-1.el7_9.x86_64/jre/lib/jce.jar
- /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.282.b08-1.el7_9.x86_64/jre/lib/jfr.jar
- /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.282.b08-1.el7_9.x86_64/jre/lib/jsse.jar
- /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.282.b08-1.el7_9.x86_64/jre/lib/resources.jar
- /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.282.b08-1.el7_9.x86_64/jre/lib/rt.jar
- /usr/lib/locale/locale-archive
- /usr/lib64/ld-2.17.so

- /usr/lib64/libc-2.17.so
- /usr/lib64/libdl-2.17.so
- /usr/lib64/libgcc_s-4.8.5-20150702.so.1
- /usr/lib64/libm-2.17.so
- /usr/lib64/libnss_dns-2.17.so
- /usr/lib64/libnss_files-2.17.so
- /usr/lib64/libpthread-2.17.so
- /usr/lib64/libresolv-2.17.so
- /usr/lib64/librt-2.17.so
- /usr/lib64/libstdc++.so.6.0.19
- /usr/lib64/libz.so.1.2.7

6.3 Third-Party Libraries

The TOE is packaged with the following third-party libraries:

- Azul Zulu Java SE 8 Update 272 (Windows only)
- Apache Tomcat 9.0.41
- agent-commons.jar
- all-themes-1.0.8.jar
- apache-mime4j-core-0.7.2.jar
- aws-java-sdk-cloudfront-1.11.631.jar
- aws-java-sdk-core-1.11.631.jar
- aws-java-sdk-kms-1.11.631.jar
- aws-java-sdk-s3-1.11.631.jar
- aws-java-sdk-sts-1.11.631.jar
- azure-storage-5.5.0.jar
- batik-all-1.14.jar
- bc-fips-1.0.2.jar
- bcmail-fips-1.0.3.jar
- bcpg-fips-1.0.5.jar
- bcpkix-fips-1.0.4.jar
- bctls-fips-1.0.10.3.jar
- bluesky-1.0.6.jar
- bsh-2.0b6.jar
- chartcreator-1.2.0.jar
- commons-beanutils-1.9.4.jar
- commons-cli-1.3.1.jar
- commons-codec-1.14.jar
- commons-collections-3.2.2.jar
- commons-collections4-4.1.jar
- commons-compress-1.19.jar
- commons-configuration-1.7.jar
- commons-dbcp-1.3.jar
- commons-digester-1.8.1.jar
- commons-discovery-0.4.jar
- commons-el.jar
- commons-fileupload-1.4.jar
- commons-httpclient-3.1.jar
- commons-io-2.6.jar
- commons-lang-2.1.jar
- commons-lang3-3.9.jar
- commons-logging-1.2.jar
- commons-math3-3.6.1.jar
- commons-net-3.3.jar
- commons-pool-1.6.jar
- commons-validator-1.5.0.jar
- commons-vfs2-2.1.jar
- cryptojce.jar
- cryptojcommon.jar
- cssparser-0.9.14.jar
- curvesapi-1.06.jar
- db2jcc.jar
- derbyclient.jar
- derby.jar
- edi-definitions-1.0.jar
- ehcache-core-2.5.1.jar
- esapi-2.1.0.1.jar
- facestrace-0.9.0.jar
- FastInfoset.jar
- font-awesome-5.6.1.jar
- ga_classes.jar
- gaclient.jar
- gacmd.jar
- ga_resources.jar
- gateway-client.jar
- gateway-commons.jar

- gfaces.jar
- ghttpsclient.jar
- gmbal-api-only.jar
- gofast-all.jar
- gson-2.2.4.jar
- gt-commons.jar
- guava-26.0.jar
- ha-api.jar
- httpclient-4.5.13.jar
- httpcore-4.4.14.jar
- icu4j-63.1.jar
- ifxjdbc.jar
- imgscalr-lib-4.2.jar
- ion-java-1.0.2.jar
- ipworkszip.jar
- iText-2.1.7.jar
- jackson-annotations-2.10.1.jar
- jackson-core-2.10.1.jar
- jackson-databind-2.10.5.1.jar
- jakarta-oro.jar
- jasperreports-6.16.0.jar
- jasperreports-chart-themes-6.16.0.jar
- jasperreports-fonts-6.16.0.jar
- jasypt-1.9.2.jar
- java-jwt-3.3.0.jar
- javax.annotation-api.jar
- javax.xml.soap-api.jar
- jaxb-api.jar
- jaxb-core.jar
- jaxb-impl.jar
- jaxb-jxc.jar
- jaxb-xjc.jar
- jaxws-api.jar
- jaxws-rt.jar
- jaxws-tools.jar
- jcifs-1.3.18.jar
- jcmFIPS.jar
- jcommon-1.0.10.jar
- jfreechart-1.0.19.jar
- jgroups-4.1.2.Final.jar
- jmespath-java-1.11.631.jar
- jms.jar
- JNQ-1.3.6.jar
- joda-time-2.2.jar
- jsch-0.1.54.jar
- jsr181-api.jar
- jt400.jar
- jTDS3.jar
- jxl.jar
- jzlib-1.1.2.jar
- licenseapi-2.0.jar
- linoma-batik.jar
- linoma-commons.jar
- linoma-crypto.jar
- linoma-expressions.jar
- linoma-httpclient.jar
- linoma-kt.jar
- linoma-projects.jar
- linoma-report-fonts.jar
- linoma-security.jar
- linoma-tools.jar
- linoma-wss4j-ws-security-common-2.0.10.jar
- linoma-wss4j-ws-security-dom-2.0.10.jar
- log4j-1.2-api-2.13.3.jar
- log4j-api-2.13.3.jar
- log4j-core-2.13.3.jar
- log4j-slf4j-impl-2.13.3.jar
- lucene-analyzers-common-4.7.2.jar
- lucene-codecs-4.7.2.jar
- lucene-core-4.7.2.jar
- lucene-grouping-4.7.2.jar
- lucene-queries-4.7.2.jar
- lucene-queryparser-4.7.2.jar
- management-api.jar
- mariadb-java-client-1.7.1.jar
- maverick-legacy-common-1.7.34.jar
- maverick-legacy-server-1.7.34.jar
- mimepull.jar
- mina-core-2.1.4.jar
- msbase.jar
- mssqlserver.jar
- msutil.jar
- myfaces-api-2.2.12.jar
- myfaces-bundle-2.2.12.jar
- myfaces-impl-2.2.12.jar
- native-lib-loader-2.0.2.jar
- netty-all-4.1.48.Final.jar
- not-going-to-be-commons-ssl-0.3.18.jar
- ojdbc5.jar
- opensaml-2.6.6.jar

- openws-1.5.4.jar
- oro-2.0.8.jar
- owasp-java-html-sanitizer-r239.jar
- pesit-client-1.2.1.jar
- pesit-commons-1.1.3.jar
- pesit-server-1.1.4.jar
- poi-4.1.1.jar
- poi-ooxml-4.1.1.jar
- poi-ooxml-schemas-4.1.1.jar
- policy.jar
- postgresql-42.2.14.jar
- prettyfaces-jsf2-3.3.0.jar
- primefaces-7.0.14.jar
- primefaces-extensions-7.0.1.jar
- qname.jar
- resolver.jar
- saaj-impl.jar
- sardine.jar
- slf4j-api-1.7.25.jar
- snmp4j-2.3.4.jar
- spring-beans-5.2.9.RELEASE.jar
- spring-context-5.2.9.RELEASE.jar
- spring-core-5.2.9.RELEASE.jar
- sqljdbc4.jar
- sslj.jar
- stax2-api-3.1.4.jar
- stax2-api.jar
- stax-api-1.0-2.jar
- stax-ex.jar
- streambuffer.jar
- taglibs-standard-impl-1.2.3.jar
- taglibs-standard-jstlel-1.2.3.jar
- taglibs-standard-spec-1.2.3.jar
- tinyradius-1.1.0.jar
- tomahawk20-1.1.14.jar
- unboundid-ldapsdk-4.0.11.jar
- velocity-1.7.jar
- woodstox-core-asl-4.4.1.jar
- woodstox-core-asl.jar
- wsbuilder.jar
- wsdl4j.jar
- xml-apis-1.3.04.jar
- xml-apis-ext-1.3.04.jar
- xmlbeans-3.1.0.jar
- xmlgraphics-commons-2.6.jar
- xmlsec-2.1.4.jar
- xmltooling-1.4.6.jar