VMware Horizon Connection Server 8 2209 (Horizon 8.7)

Security Target

Version 1.0

06 April 2023

Prepared for:

VMware, Inc.
3401 Hillview Avenue
Palo Alto, CA 94304

Prepared by:

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

Contents

1 Security Target Introduction 1
1.1 Security Target, TOE and CC Identification 1
1.2 Conformance Claims 1
1.3 Conventions 3
1.3.1 Terminology 4
1.3.2 Acronyms 4
2 Product and TOE Description 6
2.1 Introduction 6
2.2 Product Overview 6
2.3 TOE Overview 7
2.4 TOE Architecture 7
2.4.1 Physical Boundary 7
2.4.2 Logical Boundary 11
2.4.2.1 Cryptographic Support 11
2.4.2.2 User Data Protection 12
2.4.2.3 Identification and Authentication 12
2.4.2.4 Security Management 12
2.4.2.5 Privacy 12
2.4.2.6 Protection of the TSF 12
2.4.2.7 Trusted Path/Channels 12
2.5 TOE Documentation 12
3 Security Problem Definition 14
4 Security Objectives 15
5 IT Security Requirements 16
5.1 Extended Requirements 16
5.2 TOE Security Functional Requirements 17
5.2.1 Cryptographic Support (FCS) 18
5.2.1.1 FCS_CKM_EXT.1 Cryptographic Key Generation Services 18
5.2.1.2 FCS_CKM.1/AK Cryptographic Asymmetric Key Generation 18
5.2.1.3 FCS_CKM.1/SK Cryptographic Symmetric Key Generation 19
5.2.1.4 FCS_CKM.2 Cryptographic Key Establishment 19
5.2.1.5 FCS_COP.1/Hash Cryptographic Operation – Hashing 19
5.2.1.6 FCS_COP.1/KeyedHash Cryptographic Operation – Keyed-Hash Message Authentication 20
5.2.1.7 FCS_COP.1/Sig Cryptographic Operation – Signing 20
5.2.1.8 FCS_COP.1/SKC Cryptographic Operation – Encryption/Decryption 20
5.2.1.9 FCS_HTTPS_EXT.1/Client HTTPS Protocol 20
5.2.1.10 FCS_HTTPS_EXT.1/Server HTTPS Protocol 21
5.2.1.11 FCS_HTTPS_EXT.2 HTTPS Protocol with Mutual Authentication 21
5.2.1.12 FCS_RBG_EXT.1 Random Bit Generation Services 21
5.2.1.13 FCS_RBG_EXT.2 Random Bit Generation from Application 21
5.2.1.14 FCS_STO_EXT.1 Storage of Credentials 21
5.2.1.15 FCS_TLS_EXT.1 TLS Protocol (TLS Package) 22

5.2.1.16 FCS_TLSC_EXT.1 TLS Client Protocol (TLS Package) 22
5.2.1.17 FCS_TLSC_EXT.2 TLS Client Support for Mutual Authentication (TLS Package) 22
5.2.1.18 FCS_TLSC_EXT.5 TLS Client Support for Supported Groups Extension (TLS Package) 23
5.2.1.19 FCS_TLSS_EXT.1 TLS Server Protocol (TLS Package) 23
5.2.1.20 FCS_TLSS_EXT.2 TLS Server Support for Mutual Authentication (TLS Package) 24
5.2.1.21 FCS_TLSS_EXT.4 TLS Server Support for Renegotiation (TLS Package) 24
5.2.2 User Data Protection (FDP) 24
5.2.2.1 FDP_DAR_EXT.1 Encryption of Sensitive Application Data 24
5.2.2.2 FDP_DEC_EXT.1 Access to Platform Resources 24
5.2.2.3 FDP_NET_EXT.1 Network Communications 24
5.2.3 Identification and Authentication (FIA) 25
5.2.3.1 FIA_X509_EXT.1 X.509 Certificate Validation 25
5.2.3.2 FIA_X509_EXT.2 X.509 Certificate Authentication 25
5.2.4 Security Management (FMT) 26
5.2.4.1 FMT_CFG_EXT.1 Secure by Default Configuration 26
5.2.4.2 FMT_MEC_EXT.1 Supported Configuration Mechanism 26
5.2.4.3 FMT_SMF.1 Specification of Management Functions 26
5.2.5 Privacy (FPR) 26
5.2.5.1 FPR_ANO_EXT.1 User Consent for Transmission of Personally Identifiable Information 26
5.2.6 Protection of the TSF (FPT) 26
5.2.6.1 FPT_AEX_EXT.1 Anti-Exploitation Capabilities 26
5.2.6.2 FPT_API_EXT.1 Use of Supported Services and APIs 27
5.2.6.3 FPT_IDV_EXT.1 Software Identification and Versions 27
5.2.6.4 FPT_LIB_EXT.1 Use of Third Party Libraries 27
5.2.6.5 FPT_TUD_EXT.1 Integrity for Installation and Update 27
5.2.6.6 FPT_TUD_EXT.2 Integrity for Installation and Update 27
5.2.7 Trusted Path/Channels (FTP) 28
5.2.7.1 FTP_DIT_EXT.1 Protection of Data in Transit 28
5.3 TOE Security Assurance Requirements 28
6 TOE Summary Specification 30
6.1 Timely Security Updates 30
6.2 Cryptographic Support 31
6.3 User Data Protection 35
6.4 Identification and Authentication 36
6.5 Security Management 37
6.6 Privacy 38
6.7 Protection of the TSF 39
6.8 Trusted Path/Channels 39
7 Protection Profile Claims 42
8 Rationale 43
8.1 TOE Summary Specification Rationale 43
A TOE Usage of Third-Party Components 45
A.1 Platform APIs 45
A.2 Third-Party Libraries 46

Tables

Table 1: Terms and Definitions 4
Table 2: Acronyms 4
Table 3: TOE Security Functional Components 17
Table 4: Assurance Components 28
Table 5: Cryptographic Algorithm Claims 31
Table 6: TSF Network Usage 35
Table 7: Security Functions vs. Requirements Mapping 43

1 Security Target Introduction

The Security Target (ST) contains the following additional sections:

- Product and TOE Description (Section 2)
- Security Problem Definition (Section 3)
- Security Objectives (Section 4)
- IT Security Requirements (Section 5)
- TOE Summary Specification (Section 6)
- Protection Profile Claims (Section 7)
- Rationale (Section 8)
- **Error! Reference source not found.** (Appendix **Error! Reference source not found.**)

1.1 Security Target, TOE and CC Identification

ST Title – VMware Horizon Connection Server 8 2209 (Horizon 8.7) Security Target

ST Version – Version 1.0

ST Date – 06 April 2023

TOE Identification – VMware Horizon Connection Server 8 2209 (Horizon 8.7)

TOE Developer – VMware, Inc.

Evaluation Sponsor – VMware, Inc.

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:

- *Protection Profile for Application Software, Version 1.4, 07 October 2021* (App PP) with the following optional and selection-based SFRs:
  - FCS_CKM.1/AK
  - FCS_CKM.1/SK
  - FCS_CKM.2
  - FCS_COP.1/Hash
  - FCS_COP.1/KeyedHash
  - FCS_COP.1/Sig
  - FCS_COP.1/SKC
  - FCS_HTTPS_EXT.1/Client
  - FCS_HTTPS_EXT.1/Server
  - FCS_HTTPS_EXT.2
  - FCS_RBG_EXT.2
  - FIA_X509_EXT.1
  - FIA_X509_EXT.2
  - FPT_TUD_EXT.2

- *Functional Package for Transport Layer Security (TLS), Version 1.1, February 12, 2019* (TLS Package) with the following optional and selection-based SFRs:
  - FCS_TLSC_EXT.1
  - FCS_TLSC_EXT.2
  - FCS_TLSC_EXT.5
  - FCS_TLSS_EXT.1
  - FCS_TLSS_EXT.2
  - FCS_TLSS_EXT.4
- The following NIAP Technical Decisions apply to the TOE and have been accounted for in the ST development and the conduct of the evaluation, or were considered to be non-applicable:

  TD0442: Updated TLS Ciphersuites for TLS Package

  - This TD is applicable to the TOE.

  TD0469: Modification of test activity for FCS_TLSS_EXT.1.1 test 4.1

  - This TD is applicable to the TOE but applies specifically to test activities so the ST itself is unaffected.

  TD0499: Testing with pinned certificates

  - This TD is applicable to the TOE because it affects an SFR that the TOE claims. However, the TOE does not support certificate pinning so the TD's modification to the testing for this does not affect the claims made for the TSF.

  TD0513: CA Certificate loading

  - This TD is applicable to the TOE but applies specifically to test activities so the ST itself is unaffected.

  TD0588: Session Resumption Support in TLS package

  - This TD is applicable to the TOE.

  TD0624: Addition of DataStore for Storing and Setting Configuration Options

  - This TD is not applicable to the TOE because it does not have an Android platform version.

  TD0628: Addition of Container Image to Package Format

  - This TD is not applicable to the TOE because it is not packaged in a container format.

  TD0650: Conformance claim sections updated to allow for MOD_VPNC_V2.3 and 2.4

  - This TD is not applicable to the TOE because it does not claim VPN client functionality.

  TD0655: Mutual authentication in FTP_DIT_EXT.1 for SW App

  - This TD is applicable to the TOE.

  TD0664: Testing activity for FPT_TUD_EXT.2.2

  - This TD is applicable to the TOE.

TD0669: FIA_X509_EXT.1 Test 4 Interpretation

- This TD is applicable to the TOE.

TD0709: Number of elements for iterations of FCS_HTTPS_EXT.1

- This TD is applicable to the TOE.

TD0717: Format changes for PP_APP_V1.4

- This TD is applicable to the TOE.

TD0719: ECD for PP APP V1.3 and 1.4

- This TD is not applicable to the TOE; this TD updates the App PP to include a formal ECD which is needed for the PP itself to conform to CC Part 3. This does not change the ST or how the evaluation of the TOE is conducted.

TD0726: Corrections to (D)TLSS SFRs in TLS 1.1 FP

- This TD is applicable to the TOE.

- 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

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. An iterated SFR is indicated by a slash followed by a descriptor for the purpose of the iteration. For example, FCS_HTTPS_EXT.1/Client indicates that the FCS_HTTPS_EXT.1 requirement applies specifically to HTTPS client functionality whereas FCS_HTTPS_EXT.1/Server applies specifically to HTTPS server functionality.
  - Assignment: allows the specification of an identified parameter. Assignments are indicated using italics and are surrounded by brackets (e.g., [*assignment item*]). Note that an assignment within a selection would be identified in both italics and underline, with the brackets themselves underlined since they are explicitly part of the selection text, unlike the brackets around the selection itself (e.g., [selection item, [*assignment item inside selection*]]).
  - Selection: allows the specification of one or more elements from a list. Selections are indicated using underlines and are surrounded by brackets (e.g., [selection item]).
  - Refinement: allows technical changes to a requirement to make it more restrictive and allows non-technical changes to grammar and formatting. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …"). Note

that minor grammatical changes that do not involve the addition or removal of entire words (e.g., for consistency of quantity such as changing “meets” to “meet”) do not have formatting applied.

- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.
- The ST does not show selection/assignment operations that have been completed by the PP authors, though it does preserve brackets to show where such operations have been made.

1.3.1 Terminology

The following terms and abbreviations are used in this ST:

Table 1: Terms and Definitions

Term	Definition
Agent	A Horizon component that acts as an endpoint on a protected resource and serves content on that resource (individual applications or an interactive desktop session) to an authorized Horizon Client.
Blast	A communications protocol that is used to transmit interactive desktop and application sessions (user inputs and audio/visual outputs).
Client	A Horizon component that resides on an end user device that the user can run to access enterprise computing resources via the virtual desktop.
Cloud Pod	A self-contained Horizon deployment on a particular network. Multiple cloud pods can be federated, allowing a client on one pod to access resources on another.
Connection Server	A Horizon component that is responsible for determining the authorizations of a Horizon Client user and facilitating the establishment of Agent communications so that authorized resources can be served to that user.
Horizon	A collection of products that are used to allow an organizational user to access shared enterprise resources in a protected network from a single client application.
Unified Access Gateway	A network device that acts as a proxy between a Horizon Client on an unprotected network and other Horizon components on a protected internal network. The Unified Access Gateway is responsible for authenticating Horizon Client users and passing their validated identity to a Connection Server via SAML assertion. It is also responsible for establishing Horizon Agent connectivity on behalf of the client.
vCenter	A VMware product for creating and managing virtual machines. Used by the TOE to create, start, and stop virtual machines running Horizon Agents on demand. These virtual machines are used to provide appropriate content to a Horizon Client user based on their authorizations.
Virtual Desktop	The virtual desktop is the set of enterprise computing resources that are served to a user within an interactive Horizon Client session. For the purpose of the TSF, the important consideration is that all virtual desktop content is transmitted over TLS.

1.3.2 Acronyms

Table 2: Acronyms

Term	Definition
ASLR	Address Space Layout Randomization
CRUD	Create, Read, Update, and Delete
DRBG	Deterministic Random Bit Generator
ECC	Elliptic Curve Cryptography
JRE	Java Runtime Environment
PGP	Pretty Good Privacy
PII	Personally Identifiable Information
PP	Protection Profile
SAML	Security Assertion Markup Language
SAN	Subject Alternative Name
SSPI	Security Support Provider Interface
ST	Security Target
SWID	Software Identification (standard)
TOE	Target of Evaluation
TSF	TOE Security Functionality
UAG	Unified Access Gateway
VPN	Virtual Private Network
WAN	Wide-Area Network

2 Product and TOE Description

2.1 Introduction

VMware Horizon is a collection of applications that work together to deliver centralized enterprise resources to end users. This is done by providing users with a "virtual desktop" that consolidates their authorized enterprise computing environments and applications into a single view that is presented to them through a client application.

For this Security Target, the Target of Evaluation (TOE) is the VMware Horizon Connection Server 8 application, specifically version 2209 or 8.7. This is a server application that is responsible for application-layer user authorization that allows endpoint clients to connect to the agent servers that offer applications and services in the virtual desktop.

The TOE conforms to the App PP and TLS Package. As such, the security-relevant functionality of the product is limited to the claimed requirements in those standards. The security-relevant functionality is described in sections 2.3 and 2.4. The product overview in section 2.2 below is intended to provide the reader with an overall summary of the entire product so that its intended usage is clear. The subset of the product functionality that is within the evaluation scope is subsequently described in the sections that follow it.

2.2 Product Overview

VMware Horizon is a suite of applications that establish a virtualization environment within an organization. The Horizon applications collectively allow users to access virtualized desktops or enterprise resources from their end user device. These resources are made available with granular security controls that allow users to access only the capabilities for which they are authorized.

VMware Horizon as a suite consists of several components:

- Horizon Clients are applications that are installed on end user devices. A user accesses their virtual desktop through the Horizon Client.
- Horizon Agents are applications that run on virtual servers in the enterprise environment. These agents facilitate remote access to the desktop of a virtual server or to specific applications running on that server that may be served directly to the virtual desktop.
- The Horizon Connection Server is responsible for brokering connections between Horizon Clients and Horizon Agents to authenticate users and serve appropriate resources to a particular user based on enterprise permissions.

A VMware Horizon deployment typically includes one or more instances of the VMware Unified Access Gateway (UAG) as well. The purpose of the UAG is to enforce separation of internal and external networks and authenticate Horizon Client users. This allows the Horizon Client to act as a TLS VPN endpoint to access services within the protected network when the end user device is in an external setting such as an untrusted mobile Wi-Fi network.

In cases where a Horizon deployment needs to give users access to resources that span multiple physical data centers or are maintained by multiple organizations, Horizon also supports a cloud pod architecture. This allows for multiple Connection Servers to be federated so that access to Horizon Agent resources on disparate WANs can be served through a single Horizon Client session.

2.3 TOE Overview

The Target of Evaluation (TOE) is VMware Horizon Connection Server 8 application. The specific evaluated version of the application is version 2209 or 8.7; these are synonymous. All references to "Connection Server" throughout the ST refer to this specific version. The TOE is a Windows application.

With respect to the security functionality of the TOE, the TSF is limited to the relevant functionality that is defined in the claimed PP and package. The logical boundary of the TOE is summarized in section 2.4.2. However, the following general capabilities are considered to be within the scope of the TOE:

- **Protection of sensitive data at rest:** the TOE leverages secure platform storage and encryption mechanisms to protect credentials and other sensitive data at rest.
- **Protection of data in transit:** the TOE secures data in transit between itself and its operational environment using TLS and HTTPS.
- **Trusted updates:** the TOE provides visibility into its current running version and the vendor distributes updates to it that are digitally signed so that administrators can securely maintain up-to-date software.
- **Cryptographic services:** the TOE includes an implementation of Bouncy Castle with CAVP validated algorithm services that it uses to secure data at rest and in transit.
- **Secure interaction with operating system:** the TOE is designed to interact with its underlying host operating system platform in such a way that the TOE cannot be used as an attack vector to compromise an operating system.

Notably, there are no standardized security requirements in the claimed PP (or any other published PP at the time of this ST's publication) for application layer access authorization. Therefore, the security of the TOE's external interfaces is only assessed with respect to the ability of the TOE to protect these communications from unauthorized modification or disclosure.

2.4 TOE Architecture

The Horizon Connection Server TOE consists of the Horizon Connection Server application. The TOE has a Windows platform version only. The application consists of Java and C++ code and runs along with several services on the operating system. Third-party components are dynamically linked into the TOE or compiled into the binary. The third-party components used by the TOE are listed in Appendix A.2.

2.4.1 Physical Boundary

Figure 1 shows the TOE in a sample deployment with other VMware Horizon applications in its operational environment. Note the following:

- This figure also includes a UAG and therefore assumes that the TOE platform is located inside of the protected network in which all components reside except for the Horizon Client and its underlying system. The UAG acts as a reverse proxy to handle all inbound TLS and HTTPS connections from the Horizon Client.
- Firewalls are not shown between internal and external networks but it is assumed that the UAG is deployed in a DMZ between them.

- Multiple UAGs may be deployed in a load balancer configuration to ensure resource availability. As the claimed PP and package do not have availability requirements, only one UAG is deployed in the tested configuration.
- The second Horizon Connection Server that is depicted on the diagram is a second server deployed in an external environment. The external server has its own associated Horizon Agents and other external interfaces. These are omitted for simplicity.
- Horizon Agents are deployed on multiple virtual systems. Horizon Agents support multiple hypervisors but the TOE's evaluated configuration assumes the use of VMware ESXi for this. An instance of VMware vCenter is used for management of the VMs on which Horizon Agents run. For simplicity's sake, the Horizon Agent is represented by a single logical component on the figure. The associated physical device and hypervisor are not shown on the figure as the TSF only interfaces logically with the specific application component of those systems. The UAG and Connection Server are similarly deployed in virtualized environments and the figure does not show their dependent components.
- The Connection Server network may include multiple virtual systems on the same physical host that are networked virtually. Specifically, the same physical host may include separate VMs running Horizon Connection Server, Horizon Agent, and vCenter components all as part of the same managed VM infrastructure.
- The environment assumes that all components have access to the organization's Certification Authority for issuance and validation of X.509 certificates.
- The interface to the authentication server in the TOE's operational environment is from the TOE's OS platform. The connection is implemented through the Windows Security Support Provider Interface (SSPI). Its presence is required for the TOE to function but the TSF does not directly interface with it.
- The 'Database' component refers to the optional Event Database (SQL Server or Postgres) that is used by the TOE if configured.
- The operational environment includes two CRL distribution points: one for certificate revocation checking for certificates presented to Horizon Connection Server, and one for external network components. The external network's CRL distribution point is non-interfering with respect to the security of the TSF.

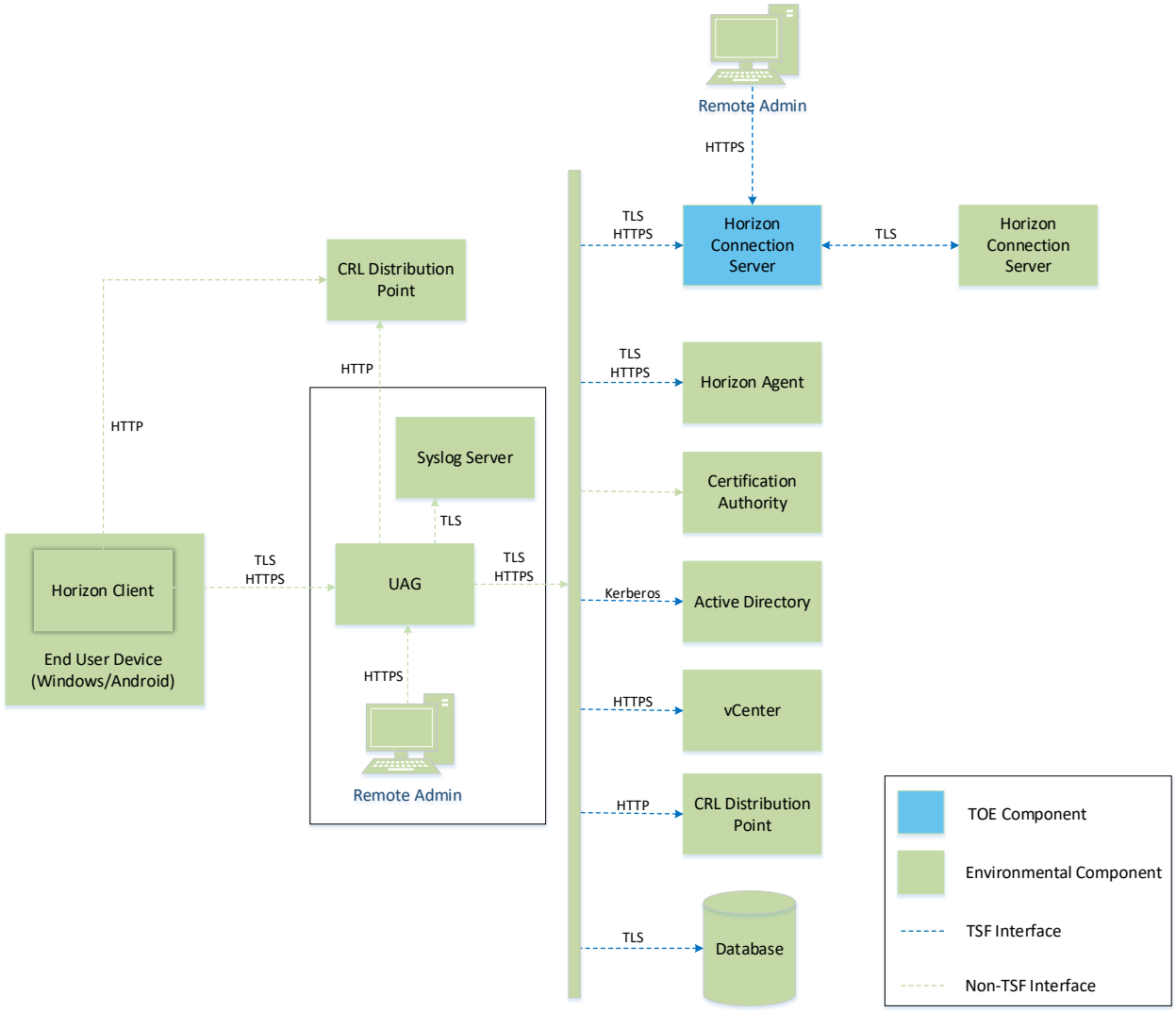

Figure 1 - TOE Boundary

The TOE receives inbound connections from Horizon Agent systems for awareness of when a particular Agent is able to facilitate a virtual desktop session. Inbound connection requests that originate from Horizon Clients are handled by the UAG, which authenticates the user and passes the request along to the TOE as a SAML assertion containing the user's identity. The TOE uses this assertion to look up user permissions and notifies an available Horizon Agent that the Horizon Client in question is requesting a session and what resources are authorized to them. The subsequent connection between the Horizon Client and Horizon Agent occurs without the TOE's involvement.

In the evaluated configuration, the TOE can interact with another instance of a Horizon Connection Server in the following case: multiple distinct environments can be federated in a cloud pod architecture, allowing a Horizon Client in one environment to access virtual desktop services provided by a separate environment.

Regarding interactions with other components, the TOE has a remote web-based management interface. Active Directory is used for user and administrator authentication. The Database component refers to the optional Event Database, which runs SQL Server or Postgres.

All interactions between multiple Connection Servers use mutually-authenticated TLS. All other uses of TLS use server-side authentication, regardless of whether the TOE acts as the client or the server.

The TOE has the following system requirements for its host platform in its evaluated configuration:

- Windows Server 2019, virtualized on VMware ESXi 7.0
- Platform must be configured into FIPS-compliant mode of operation
- Pentium IV 2.0GHz processor or higher – 4 CPUs recommended
    - The TOE's tested configuration uses an Intel Xeon 6230R (Cascade Lake)
- 4 GB RAM – at least 10GB recommended for deployments of 50 or more remote desktops
- 100 Mbps NIC – 1 Gbps recommended

The following network ports must be open for the TOE to function:

- TCP/443 (for inbound remote administration, inbound UAG connectivity, and outbound connectivity to vCenter)
- TCP/8472 (for connectivity to a Connection Server in an external cloud pod)
- TCP/4001 (for connectivity with Horizon Agents)
- TCP/88 (for Kerberos connectivity to remote authentication server)
- Database server access also requires open ports but this is configured by the remote server

The TOE's operational environment includes the following:

- Other VMware Horizon components (at least one each of Horizon Client and Horizon Agent).
- Network access between "outer" and "protected" networks mediated through at least one VMware UAG.
- Platform (hardware and software) on which the TOE is hosted.
    - The TOE is capable of running on a general-purpose Windows operating system on standard consumer-grade hardware. For the evaluated configuration, the TOE was tested on Windows Server 2019, virtualized on VMware ESXi 7.0 in a vSphere deployment.
- VMware VM Encryption and Data Protection API (DPAPI) are required for the TOE platform to ensure adequate data-at-rest protection.
- Database server – any of the following are supported:
    - Postgres up to version 13.2
    - SQL Server up to version 2019
- Authentication server (Active Directory)
- Access to a Certification Authority and corresponding revocation checking mechanism is needed to validate presented X.509 certificates.

The TOE has the following logical exclusions, in addition to any functionality that is not directly related to any of the SFR claims made in this ST:

- OpenSSL – The product includes an OpenSSL cryptographic library. This is used for functionality that is outside the evaluated configuration of the TOE so it is therefore excluded. Specifically, OpenSSL is used by multiple interfaces to facilitate remote client connectivity in the case where the environment does not include a UAG. Since the environment does include a UAG, these interfaces will be disabled when the TOE is in its evaluated configuration.
- Enrollment Server – The product can be configured as an Enrollment Server, which another Connection Server interfaces with for registration of Horizon Clients and establishment of end user credentials. This is excluded from the evaluated configuration because Horizon Client users are expected to authenticate using smart card PKI tokens, which do not require the use of an Enrollment Server.
- Replica Server – A secondary instance of a Connection Server can be deployed in an environment for failover or load balancing purposes. This is excluded from the evaluated configuration because the claimed PP does not enforce availability requirements or specify availability as a security objective.
- For environmental components that the TOE interfaces with, if the application layer behavior of that interface is not within the scope of the claimed PP, the interface is only security-relevant to the extent that it asserts protection of data in transit. For example, the event database server interface is security-relevant because it uses TLS to protect data in transit, not because of the data it transmits, which is not relevant to any of the requirements in the claimed PP.

The TOE has multiple editions with different features that are activated by licensing. The security functionality claimed within the TOE boundary is not affected by which license is used. The highest tier edition (Enterprise) was used for the tested configuration.

2.4.2 Logical Boundary

This section summarizes the security functions provided by the TOE:

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

2.4.2.1 Cryptographic Support

The TOE makes use of cryptography to protect data at rest and in transit.

For data in transit, the TOE implements TLS with and without HTTPS as a client and a server. The TOE supports mutual authentication for some interfaces.

The TOE implements cryptography used for these functions using its own implementation of Bouncy Castle (BC-FJA) with NIST-approved algorithms. The TOE's DRBG is seeded using entropy from the underlying OS platform.

For data at rest, the TOE relies on its operational environment to protect stored credential data.

2.4.2.2 User Data Protection

The TOE relies on volume encryption via VMware VM Encryption to protect sensitive data at rest, as well as the mechanisms used to protect credential data at rest.

The TOE relies on the network connectivity and logging functions of its host OS platform.

2.4.2.3 Identification and Authentication

The TOE supports X.509 certificate validation as part of establishing TLS and HTTPS connections. Depending on the specific check being performed, the TSF is either responsible for certificate validation or relies on its OS platform for this function. The TOE supports various certificate validity checking methods and can also check certificate revocation status using CRL or OCSP. If the validity status of a certificate cannot be determined, the certificate will be rejected. All other cases where a certificate is found to be invalid will result in rejection without an administrative override.

2.4.2.4 Security Management

The TOE itself and the configuration settings it uses are stored in locations recommended by the platform vendor. The TOE is administered over a dedicated logical interface that requires administrator authentication prior to access. This interface is used to perform various security-relevant management functions.

2.4.2.5 Privacy

The TOE does not have a mechanism to request or transmit personally identifiable information (PII) of any individuals.

2.4.2.6 Protection of the TSF

The TOE enforces various mechanisms to prevent itself from being used as an attack vector to its host OS platform. The TOE implements address space layout randomization (ASLR), does not allocate any memory with both write and execute permissions, does not write user-modifiable files to directories that contain executable files, is compiled using stack overflow protection, and is compatible with the security features of its host OS platform.

The TOE contains libraries and invokes system APIs that are well-known and explicitly identified.

The TOE has a mechanism to determine its current software version. Software updates to the TOE are acquired by a mechanism outside of the product itself (i.e. the TOE is not self-updating). All updates are digitally signed to guarantee their authenticity and integrity.

2.4.2.7 Trusted Path/Channels

The TOE encrypts sensitive data in transit between itself and its operational environment using TLS and HTTPS. These interfaces are used to secure sensitive data in transit between the TOE and its operational environment.

2.5 TOE Documentation

VMware provides the following product documentation in support of the installation and secure use of the TOE:

- Horizon Overview and Deployment Planning (https://docs.vmware.com/en/VMware-Horizon/2209/horizon-architecture-planning.pdf)
- Horizon Installation and Upgrade (https://docs.vmware.com/en/VMware-Horizon/2209/horizon-installation.pdf)
- Horizon Administration (https://docs.vmware.com/en/VMware-Horizon/2209/horizon-console-administration.pdf)
- Cloud Pod Architecture in Horizon (https://docs.vmware.com/en/VMware-Horizon/2209/horizon-cloud-pod-architecture.pdf)
- Horizon Security (https://docs.vmware.com/en/VMware-Horizon/2209/horizon-security.pdf)
- Linux Desktops and Applications in Horizon (https://docs.vmware.com/en/VMware-Horizon/2209/linux-desktops-setup.pdf)
- Windows Desktops and Applications in Horizon (https://docs.vmware.com/en/VMware-Horizon/2209/virtual-desktops.pdf)
- VMware Horizon Connection Server Common Criteria Evaluated Configuration Guide

3 Security Problem Definition

This ST includes by reference the Security Problem Definition, composed of threats and assumptions, from the App PP. The Common Criteria also provides for organizational security policies to be part of a security problem definition, but no such policies are defined in the App PP.

As a functional package, the TLS Package does not contain a Security Problem Definition. The TOE's use of TLS is intended to mitigate the T.NETWORK_ATTACK and T.NETWORK_EAVESDROP threats defined by the App PP.

In general, the threat model of the App PP is designed to protect against the following:

- Disclosure of sensitive data at rest or in transit that the user has a reasonable expectation of security for.
- Excessive or poorly-implemented interfaces with the underlying platform that allow an application to be used as an intrusion point to a system.

This threat model is applicable to the TOE because aggregated and analyzed vulnerability scan results could show an attacker what system weaknesses are present in the environment if they were able to obtain this data. It is also applicable because the TOE is a collection of executable binaries that an attacker could attempt to use to compromise the underlying OS platform if it was designed in such a manner that this exploitation was possible.

4 Security Objectives

Like the Security Problem Definition, this ST includes by reference the security objectives defined in the App PP. This includes security objectives for the TOE (used to mitigate threats) and for its operational environment (used to satisfy assumptions).

As a functional package, the TLS Package does not contain a Security Problem Definition. The TOE's use of TLS is intended to satisfy the O.PROTECTED_COMMS objective of the App PP by implementing a specific method by which network communications are protected.

5 IT Security Requirements

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

The SFRs have all been drawn from the following Protection Profiles (PP) and Functional Packages:

- *Protection Profile for Application Software*, Version 1.4, October 7, 2021
- *Functional Packages for Transport Layer Security (TLS),* Version 1.1, February 12, 2019

As a result, any selection, assignment, or refinement operations already performed by that PP on the claimed SFRs are not identified here (i.e., they are not formatted in accordance with the conventions specified in section 1.3 of this ST). Formatting conventions are only applied on SFR text that was chosen at the ST author's discretion.

5.1 Extended Requirements

All of the extended requirements in this ST have been drawn from the App PP and TLS Package. These documents define the following extended SAR and extended SFRs; since they have not been redefined in this ST, the App PP and TLS Package should be consulted for more information regarding these extensions to CC Parts 2 and 3.

Defined in App PP:

- ALC_TSU_EXT.1 Timely Security Updates
- FCS_CKM_EXT.1 Cryptographic Key Generation Services
- FCS_HTTPS_EXT.1/Client HTTPS Protocol
- FCS_HTTPS_EXT.1/Server HTTPS Protocol
- FCS_HTTPS_EXT.2 HTTPS Protocol with Mutual Authentication
- FCS_RBG_EXT.1 Random Bit Generation Services
- FCS_RBG_EXT.2 Random Bit Generation from Application
- FCS_STO_EXT.1 Storage of Credentials
- FDP_DAR_EXT.1 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 Use of Third Party Libraries
- 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

Defined in TLS Package:

- 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
- FCS_TLSS_EXT.4 TLS Server Support for Renegotiation

5.2 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by the TOE.

Table 3: TOE Security Functional Components

Requirement Class	Requirement Component
FCS: Cryptographic Support	FCS_CKM_EXT.1 Cryptographic Key Generation Services
	FCS_CKM.1/AK Cryptographic Asymmetric Key Generation
	FCS_CKM.1/SK Cryptographic Symmetric Key Generation
	FCS_CKM.2 Cryptographic Key Establishment
	FCS_COP.1/Hash Cryptographic Operation – Hashing
	FCS_COP.1/KeyedHash Cryptographic Operation – Keyed-Hash Message Authentication
	FCS_COP.1/Sig Cryptographic Operation – Signing
	FCS_COP.1/SKC Cryptographic Operation – Encryption/Decryption
	FCS_HTTPS_EXT.1/Client HTTPS Protocol
	FCS_HTTPS_EXT.1/Server HTTPS Protocol
	FCS_HTTPS_EXT.2 HTTPS Protocol with Mutual Authentication
	FCS_RBG_EXT.1 Random Bit Generation Services
	FCS_RBG_EXT.2 Random Bit Generation from Application
	FCS_STO_EXT.1 Storage of Credentials
	FCS_TLS_EXT.1 TLS Protocol (TLS Package)
	FCS_TLSC_EXT.1 TLS Client Protocol (TLS Package)
	FCS_TLSC_EXT.2 TLS Client Support for Mutual Authentication (TLS Package)
	FCS_TLSC_EXT.5 TLS Client Support for Supported Groups Extension (TLS Package)
	FCS_TLSS_EXT.1 TLS Server Protocol (TLS Package)
	FCS_TLSS_EXT.2 TLS Server Support for Mutual Authentication (TLS Package)
	FCS_TLSS_EXT.4 TLS Server Support for Renegotiation (TLS Package)
FDP: User Data Protection	FDP_DAR_EXT.1 Encryption of Sensitive Application Data
	FDP_DEC_EXT.1 Access to Platform Resources
	FDP_NET_EXT.1 Network Communications
FIA: Identification and Authentication	FIA_X509_EXT.1 X.509 Certificate Validation
	FIA_X509_EXT.2 X.509 Certificate Authentication

Requirement Class	Requirement Component
FMT: Security Management	FMT_CFG_EXT.1 Secure by Default Configuration
	FMT_MEC_EXT.1 Supported Configuration Mechanism
	FMT_SMF.1 Specification of Management Functions
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 Use of Third Party Libraries
	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

5.2.1 Cryptographic Support (FCS)

5.2.1.1 FCS_CKM_EXT.1 Cryptographic Key Generation Services[1]

FCS_CKM_EXT.1.1 The application shall [

- Implement asymmetric key generation

].

5.2.1.2 FCS_CKM.1/AK Cryptographic Asymmetric Key Generation[2]

FCS_CKM.1.1/AK The application shall [

- implement functionality

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

- [ECC schemes] using ["NIST curves" P-384 and [P-256, P-521]] that meet the following: [FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4];
- [FFC Schemes] using cryptographic key sizes of [2048-bit or greater] that meet the following: FIPS PUB 186-4, "Digital Signature Standards (DSS)", Appendix B.1];
- [FFC Schemes] using ["safe-prime" groups] that meet the following: [NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and [RFC 7919]]

[1] Modified by TD0717

[2] Modified by TD0717

].

5.2.1.3 FCS_CKM.1/SK Cryptographic Symmetric Key Generation

FCS_CKM.1.1/SK The application shall generate symmetric cryptographic keys using a Random Bit Generator as specified in FCS_RBG_EXT.1 and specified cryptographic key sizes [128 bit, 256 bit].

5.2.1.4 FCS_CKM.2 Cryptographic Key Establishment

FCS_CKM.2.1 The application shall [implement functionality] to perform cryptographic key establishment in accordance with a specified cryptographic key establishment method:

[

- [Elliptic curve-based key establishment schemes] that meets the following: [NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"];
- [FFC Schemes using "safe-prime" groups] that meet the following: 'NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and [RFC 7919]

].

5.2.1.5 FCS_COP.1/Hash Cryptographic Operation – Hashing[3]

FCS_COP.1.1/Hash 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].

[3] Modified by TD0717

5.2.1.6 FCS_COP.1/KeyedHash Cryptographic Operation – Keyed-Hash Message Authentication[4]

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

- HMAC-SHA-256,
- HMAC-SHA-384

] and [

- no other algorithms

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

5.2.1.7 FCS_COP.1/Sig Cryptographic Operation – Signing[5]

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

- RSA schemes using cryptographic key sizes of [2048-bit or greater] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5

].

5.2.1.8 FCS_COP.1/SKC Cryptographic Operation – Encryption/Decryption[6]

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

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

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

5.2.1.9 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.

[4] Modified by TD0717

[5] Modified by TD0717

[6] Modified by TD0717

FCS_HTTPS_EXT.1.3/Client The application shall [not establish the application-initiated connection] if the peer certificate is deemed invalid.

5.2.1.10 FCS_HTTPS_EXT.1/Server HTTPS Protocol[7]

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.1.3/Server The application shall [not **notify** the user and not establish the application-initiated connection] if the peer certificate is deemed invalid.

5.2.1.11 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.

5.2.1.12 FCS_RBG_EXT.1 Random Bit Generation Services

FCS_RBG_EXT.1.1 The application shall [

- implement DRBG functionality

] for its cryptographic operations.

5.2.1.13 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 [Hash_DRBG(any)].

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.

5.2.1.14 FCS_STO_EXT.1 Storage of Credentials

FCS_STO_EXT.1.1 The application shall [

[7] Modified by TD0709

- invoke the functionality provided by the platform to securely store [*X.509 certificates, external system credentials*]

] to non-volatile memory.

5.2.1.15 FCS_TLS_EXT.1 TLS Protocol (TLS Package)

FCS_TLS_EXT.1.1 The product shall implement [

- TLS as a client,
- TLS as a server

].

5.2.1.16 FCS_TLSC_EXT.1 TLS Client Protocol (TLS Package)[8]

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_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246,
- TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288,
- TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288,
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 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

].

5.2.1.17 FCS_TLSC_EXT.2 TLS Client Support for Mutual Authentication (TLS Package)

FCS_TLSC_EXT.2.1 The product shall support mutual authentication using X.509v3 certificates.

[8] Modified by TD0442

5.2.1.18 FCS_TLSC_EXT.5 TLS Client Support for Supported Groups Extension (TLS Package)

FCS_TLSC_EXT.5.1 The product shall present the Supported Groups Extension in the Client Hello with the supported groups [

- secp256r1,
- secp384r1,
- secp521r1,
- ffdhe2048(256),
- ffdhe3072(257),
- ffdhe4096(258),
- ffdhe6144(259),
- ffdhe8192(260)].

5.2.1.19 FCS_TLSS_EXT.1 TLS Server Protocol (TLS Package)[9]

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_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246,
- TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288,
- TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288,
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

] and also supports functionality for [

- mutual authentication,
- session renegotiation,
- session resumption based on session IDs according to RFC 4346 (TLS 1.1) or RFC 5246 (TLS 1.2)

].

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[10] The product shall perform key establishment for TLS using [

- Diffie-Hellman groups [ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192] and no other groups,
- ECDHE parameters using elliptic curves [secp256r1, secp384r1, secp521r1] and no other curves

[9] Modified by TD0442 and TD0588

[10] Modified by TD0726

].

5.2.1.20 FCS_TLSS_EXT.2 TLS Server Support for Mutual Authentication (TLS Package)

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.

5.2.1.21 FCS_TLSS_EXT.4 TLS Server Support for Renegotiation (TLS Package)

FCS_TLSS_EXT.4.1 The product shall support the "renegotiation_info" TLS extension in accordance with RFC 5746.

FCS_TLSS_EXT.4.2 The product shall include the renegotiation_info extension in ServerHello messages.

5.2.2 User Data Protection (FDP)

5.2.2.1 FDP_DAR_EXT.1 Encryption of Sensitive Application Data

FDP_DAR_EXT.1.1 The application shall [

- protect sensitive data in accordance with FCS_STO_EXT.1,
- leverage platform-provided functionality to encrypt sensitive data

] in non-volatile memory.

5.2.2.2 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

].

5.2.2.3 FDP_NET_EXT.1 Network Communications

FDP_NET_EXT.1.1 The application shall restrict network communication to [

- user-initiated communication for [*remote administration*],
- respond to [*UAG-provided SAML tokens asserting identity of Horizon Client users, Horizon Agent bootstrap and status/event channel communications, communication with environmental Connection Servers*],

- [*application-initiated connectivity to external database, VMware vCenter, CRL distribution point or OCSP responder, environmental Connection Servers*]

].

5.2.3 Identification and Authentication (FIA)

5.2.3.1 FIA_X509_EXT.1 X.509 Certificate Validation

FIA_X509_EXT.1.1 The application shall [invoke platform provided functionality, 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 [OCSP as specified in RFC 6960, CRL as specified in RFC 5280 Section 6.3, CRL as specified in RFC 8603]
- The application shall validate the extendedKeyUsage (EKU) field according to the following rules:
  - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing Purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
  - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the EKU field.
  - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the EKU field.
  - S/MIME certificates presented for email encryption and signature shall have the Email Protection purpose (id-kp 4 with OID 1.3.6.1.5.5.7.3.4) in the EKU field.
  - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the EKU field.
  - Server certificates presented for EST shall have the CMC Registration Authority (RA) purpose (id-kp-cmcRA with OID 1.3.6.1.5.5.7.3.28) in the EKU field.

5.2.3.2 FIA_X509_EXT.2 X.509 Certificate Authentication

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

5.2.4.1 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 users.

5.2.4.2 FMT_MEC_EXT.1 Supported Configuration Mechanism

FMT_MEC_EXT.1.1 The application shall [

- invoke the mechanisms recommended by the platform vendor for storing and setting configuration options].

5.2.4.3 FMT_SMF.1Specification of Management Functions

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

- [*log bundle collection,*
- *administer policy, including idle session policy,*
- *allocate roles to administrative users,*
- *administer entitlements to resources,*
- *helpdesk functions (view status of desktop or application sessions, administration of helpdesk access to desktop resources, perform remote assistance to TOE users on connected desktops, disconnect and log off desktop or application sessions, restart virtual desktop infrastructure VM, send notification to published desktop or application)*]

].

5.2.5 Privacy (FPR)

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

FPR_ANO_EXT.1.1 The application shall [

- not transmit PII over a network

].

5.2.6 Protection of the TSF (FPT)

5.2.6.1 FPT_AEX_EXT.1 Anti-Exploitation Capabilities

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

FPT_AEX_EXT.1.2 The application shall [

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

].

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

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

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

5.2.6.2 FPT_API_EXT.1 Use of Supported Services and APIs

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

5.2.6.3 FPT_IDV_EXT.1 Software Identification and Versions

FPT_IDV_EXT.1.1 The application shall be versioned with [[*date-based versioning, major/minor release versioning*]].

5.2.6.4 FPT_LIB_EXT.1 Use of Third Party Libraries

FPT_LIB_EXT.1.1 The application shall be packaged with only [*third-party libraries listed in Appendix A.2*].

Application Note: *The TOE uses a substantial number of third-party libraries so this information has been provided in an Appendix for readability purposes.*

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

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

5.2.6.6 FPT_TUD_EXT.2 Integrity for Installation and Update

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

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

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

5.2.7 Trusted Path/Channels (FTP)

5.2.7.1 FTP_DIT_EXT.1 Protection of Data in Transit[11]

FTP_DIT_EXT.1.1 The application shall [

- encrypt all transmitted [sensitive data] with [HTTPS as a client in accordance with FCS_HTTPS_EXT.1/Client, HTTPS as a server using mutual authentication in accordance with FCS_HTTPS_EXT.2, TLS as a server as defined in the Functional Package for TLS and also supports functionality for [mutual authentication], TLS as a client as defined in the Functional Package for TLS]

] between itself and another trusted IT product.

5.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are included by reference to the App PP.

Table 4: Assurance Components

Requirement Class	Requirement Component
ADV: Development	ADV_FSP.1 Basic Functional Specification
AGD: Guidance Documentation	AGD_OPE.1 Operational User Guidance
	AGD_PRE.1 Preparative Procedures
ALC: Life-cycle Support	ALC_CMC.1 Labeling 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

As a functional package, the TLS Package does not define its own SARs. The expectation is that all SARs required by the App PP will apply to the entire TOE, including the portions addressed by the TLS Package. Consequently, the evaluation activities specified in the App PP apply to the entire TOE evaluation, including any changes made to them by subsequent NIAP Technical Decisions as summarized in section 1.2 above.

[11] Modified by TD0655

The TLS Package does contain evaluation activities for how to evaluate its SFR claims as part of the evaluation of ASE_TSS.1, AGD_OPE.1, AGD_PRE.1, and ATE_IND.1. All Security Functional Requirements specified by the TLS Package will be evaluated in the manner specified in that package.

6 TOE Summary Specification

This chapter describes the security functions of the TOE:

- Timely Security Updates
- Cryptographic Support
- User Data Protection
- Identification and Authentication
- Security Management
- Privacy
- Protection of the TSF
- Trusted Path/Channels

6.1 Timely Security Updates

VMware uses an internal classification system to categorize product security flaws by severity level. The classifications and their respective service-level agreements for mitigation are as follows:

- Critical:
  - Vulnerabilities that can be exploited by an unauthenticated attacker from the Internet or those that break the guest/host Operating System isolation. The exploitation results in the complete compromise of confidentiality, integrity, and availability of user data and/or processing resources without user interaction. Exploitation could be leveraged to propagate an Internet worm or execute arbitrary code between Virtual Machines and/or the Host Operating System.
  - A fix or corrective action is begun immediately and will be made available in the shortest commercially reasonable time.
- Important:
  - Vulnerabilities that are not rated critical but whose exploitation results in the complete compromise of confidentiality and/or integrity of user data and/or processing resources through user assistance or by authenticated attackers. This rating also applies to those vulnerabilities which could lead to the complete compromise of availability when exploitation is by a remote unauthenticated attacker from the Internet or through a breach of virtual machine isolation.
  - A fix will be delivered as part of the next planned maintenance release of the product and will be released as a patch if appropriate to do so.
- Moderate:
  - Vulnerabilities where the ability to exploit is mitigated to a significant degree by configuration or difficulty of exploitation, but in certain deployment scenarios could still lead to the compromise of confidentiality, integrity, or availability of user data and/or processing resources.
  - A fix will be delivered with the next planned major or minor release of the product.
- Low:
  - All other issues that have a security impact. Vulnerabilities where exploitation is believed to be extremely difficult, or where successful exploitation would have minimal impact.
  - A fix will be delivered with the next planned major or minor release of the product.

The standard release cycle for VMware products is quarterly, so all Moderate and Low findings are typically resolved within a maximum of 90 days, while more significant findings are generally resolved in

less time. Both quarterly releases and mid-cycle patches can be obtained from https://customerconnect.vmware.com.

VMware provides an email address (security@vmware.com) that is used for the reporting of potential security findings. VMware encourages the use of Pretty Good Privacy (PGP) to encrypt any communications sent to this email address and provides a copy of their PGP public key at https://kb.vmware.com/s/article/1055.

VMware staff identifies potential vulnerabilities through third-party researchers reporting potential flaws via email, reports from field personnel, reports from customers, and monitoring of public vulnerability sites. When a report is received, VMware attempts to reproduce the finding and determine its severity. If a finding is discovered for which there is no current fix, VMware will publish a Knowledge Base article about the finding as well as any potential workarounds that may be used until an updated version of the product can be delivered.

6.2 Cryptographic Support

The TOE uses cryptography to secure data in transit between itself and its operational environment.

TSF cryptographic services are implemented by the Bouncy Castle cryptographic library included within the TOE boundary. The TOE uses VMware's BC-FJA (Bouncy Castle FIPS Java API) version 1.0.2.3 with JDK 11. The cryptographic algorithms supplied by the TOE are CAVP validated. The following table identifies the cryptographic algorithms used by the TSF, the associated standards to which they conform, and the NIST certificates that demonstrate that the claimed conformance has been met.

Table 5: Cryptographic Algorithm Claims

Functions	Libraries	Standards	Certificates
FCS_CKM.1/AK Cryptographic Asymmetric Key Generation			
ECC key pair generation (NIST curves P-256, P-384, P-521)	BC-FJA	FIPS PUB 186-4	A2841 (BC-FJA)
FFC key pair generation (2048 bit)	BC-FJA	FIPS PUB 186-4	A2841 (BC-FJA)
Safe-prime FFC key pair generation (2048, 3072, 4096, 6144, 8192 bit)	BC-FJA	NIST SP 800-56A Revision 3; RFC 7919	A2841 (BC-FJA)
FCS_CKM.2 Cryptographic Key Establishment			
Elliptic curve-based key establishment	BC-FJA	NIST SP 800-56A	A2841 (BC-FJA)
FFC safe-primes key establishment	BC-FJA	NIST SP 800-56A Revision 3; RFC 7919	A2841 (BC-FJA)
FCS_COP.1/Hash Cryptographic Operation – Hashing			
SHA-256, SHA-384, and SHA-512 (digest sizes 256, 384, and 512 bits)	BC-FJA	FIPS PUB 180-4	A2841 (BC-FJA)
FCS_COP.1/KeyedHash Cryptographic Operation – Keyed Hash Message Authentication			
HMAC-SHA-256 and SHA-384	BC-FJA	FIPS PUB 198-1 FIPS PUB 180-4	A2841 (BC-FJA)

Functions	Libraries	Standards	Certificates
FCS_COP.1/Sig Cryptographic Operation – Signing			
RSA (2048-bit, 3072-bit)	BC-FJA	FIPS PUB 186-4, Section 5	A2841 (BC-FJA)
FCS_COP.1/SKC Cryptographic Operation – Encryption/Decryption			
AES-CBC (128 bits, 256 bits)	BC-FJA	CBC as defined in NIST SP 800-38A	A2841 (BC-FJA)
AES-GCM (128 bits, 256 bits)	BC-FJA	GCM as defined in NIST SP 800-38D	A2841 (BC-FJA)
AES-CTR (128 bits)	BC-FJA	CTR as defined in NIST SP 800-38A	A2841 (BC-FJA)
FCS_RBG_EXT.2 Random Bit Generation from Application			
Hash_DRBG (128 bits, 256 bits)	BC-FJA	NIST SP 800-90A NIST SP 800-57	A2841 (BC-FJA)

The TSF generates keys in support of trusted communications, depending on the specific channel being used. The TOE generates ECC keys using P-256, P-384, and P-521 and ephemeral DH keys using ffdhe groups 2048, 3072, 4096, 6144, and 8192. These keys are generated in support of the DHE and ECDHE key establishment schemes that are used for TLS and TLS/HTTPS communications.

The TOE also generates 128-bit AES_CTR keys that are used as message keys. Message keys are single-use keys that are wrapped using the recipient's public key. The TOE implements DSA key generation to generate 2048-bit signing keys for data integrity.

To ensure sufficient key strength, the TOE implements DRBG functionality for key generation, using the Hash_DRBG. The proprietary Entropy Analysis Report (EAR) describes how the TSF extracts random data from software-based sources to ensure that an amount of entropy that is at least equal to the strength of the generated keys is present (i.e., at least 256 bits when the largest supported keys are generated) when seeding the DRBG for key generation purposes. The TOE relies on a third-party entropy source provided by the platform vendor. Specifically, random numbers are obtained from the BCryptGenRandom platform API. It is assumed that the platform provides at least 256 bits of entropy.

The TOE uses RSA digital signature functions in support of TLS and HTTPS communications.

The TOE uses hash functions in support of TLS communications (SHA-256, SHA-384), HTTPS communications (SHA-256, SHA-384), digital signatures (SHA-384, SHA-512), and certificate identity verification (SHA-256).

The TOE uses keyed-hash functions (HMAC-SHA-256, HMAC-SHA-384) in support of TLS and HTTPS communications.

The TOE uses TLS 1.2, both standalone and as part of HTTPS, for client and server communications. Older SSL and TLS versions are not accepted. The TOE's implementation of TLS conforms to RFC 5246 and its implementation of HTTPS conforms to RFC 2818. The specific TOE network interfaces are documented below in section 6.3. The TOE has both a TLS client and a TLS server; they both offer the following cipher suites in the evaluated configuration:

- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256

- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
- TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

All supported ciphersuites use finite field or elliptic curve ephemeral Diffie-Hellman as the method of key establishment. For finite field Diffie-Hellman, the TSF presents the groups ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, and ffdhe8192 in the Supported Groups extension as the parameter used for key establishment. For elliptic curve Diffie-Hellman, the TSF presents secp256r1, secp384r1, and secp521r1 in the Supported Groups extension as the parameter used for key establishment. If a presented certificate is invalid, the TSF will automatically reject it when in its evaluated configuration, regardless of whether the connection is TLS or HTTPS or if the presented certificate is from a client or server peer. Digital signatures for 2048-bit and 3072-bit RSA certificates are supported.

As part of certificate validation in the establishment of TLS connectivity, the TOE will validate the reference identifier of a presented server certificate. This is done through comparison of the DN in the connection URL against the CN and SAN in the certificate. IP addresses are not supported. Wildcards are only supported for the left-most label immediately preceding the public suffix. Certificate pinning is not supported.

When acting as a client, the TOE supports one-way TLS functionality for outbound communications to the remote database and vCenter.

When acting as a server, the TOE supports one-way or mutually-authenticated TLS functionality for remote administration and one-way TLS functionality for SAML assertions of Horizon Client users. The TOE's TLS server supports both session renegotiation and session resumption using session IDs.

The TOE also supports TLS connectivity between itself and external Connection Servers, i.e. when interacting across multiple cloud pods. The TOE can act as the client or the server in this case, depending on which Connection Server is initiating the connection. All communication between Connection Servers is mutually authenticated. When mutual authentication is configured, the expected identity of the client certificate is provided as a SHA-256 hash of the certificate, which includes the DN and SAN values. The TSF will compare the hash of a presented client certificate to the stored hash associated with that certificate's identity (subject name or SAN) and reject the client request if the values do not match.

The TSF is responsible for the TLS implementation of external database communications, remote administration, cloud pod communications, VMware vCenter communications, and passthrough communication of Horizon Client user SAML token from the environmental UAG.

The TOE relies on platform-provided storage mechanisms to protect stored credential data. All certificates are protected by the platform via the Windows Certificate Store. Any credential where the TOE authenticates to an external system (e.g. vCenter) is stored locally in an LDAP directory. These credentials are separate from credentials used by administrators to access the TOE, which are maintained by the environmental Active Directory. All credential data marked as sensitive in the LDAP directory is protected by the OS platform using DPAPI. The master key for encryption of the directory is stored in the Windows Registry and is also protected using DPAPI. The following credentials are protected in this manner:

- Per-domain per-administrator accounts for one-way and no-trust domain relationships

- Per-domain service accounts for no-trust domains
- Per-domain service accounts for VM domain join
- Per-resource access credentials
- vCenter CRUD service account
- Event database CRUD service account
- Network resource access credentials for Hybrid Logon user sessions

The Cryptographic Support security function is designed to satisfy the following security functional requirements:

- FCS_CKM_EXT.1 – The TOE implements its own cryptographic functionality for asymmetric key generation.
- FCS_CKM.1/AK – The TOE uses a CAVP validated implementation to generate asymmetric keys in support of TLS and HTTPS communications.
- FCS_CKM.1/SK – The TOE uses its DRBG to generate symmetric keys in support of TLS and HTTPS communications and in encrypting sensitive data.
- FCS_CKM.2 – The TOE performs CAVP validated key establishment in support of TLS and HTTPS communications.
- FCS_COP.1/Hash – The TOE uses a CAVP validated implementation to perform cryptographic hashing in support of TLS and HTTPS communications.
- FCS_COP.1/KeyedHash – The TOE uses a CAVP validated implementation to perform HMAC functions in support of TLS and HTTPS communications.
- FCS_COP.1/Sig – The TOE uses a CAVP validated implementation to generate and verify RSA digital signatures in support of TLS and HTTPS communications.
- FCS_COP.1/SKC – The TOE uses a CAVP validated implementation to perform AES encryption and decryption in support of TLS and HTTPS communications and to encrypt/decrypt data transmitted over an unprotected channel.
- FCS_HTTPS_EXT.1/Client – The TOE implements HTTPS as a client to secure data in transit.
- FCS_HTTPS_EXT.1/Server – The TOE implements HTTPS as a server to secure data in transit.
- FCS_HTTPS_EXT.2 – For cases where the TOE implements HTTPS as a server and verifies a client certificate as part of mutual authentication, an invalid client certificate will cause the connection attempt to be rejected.
- FCS_RBG_EXT.1 – The TOE implements its own random bit generation services.
- FCS_RBG_EXT.2 – The TOE uses a CAVP validated implementation to generate pseudo-random bits and this implementation is seeded with sufficiently strong entropy collected from the operational environment.
- FCS_STO_EXT.1 – The TOE uses platform-provided mechanisms to secure credential data at rest.
- FCS_TLS_EXT.1 – The TOE implements TLS to secure data in transit.
- FCS_TLSC_EXT.1 – The TOE implements TLS as a client.

- FCS_TLSC_EXT.2 – The TOE's TLS client implementation supports mutual authentication for some TLS functions.
- FCS_TLSC_EXT.5 – The TOE's TLS client implementation presents supported elliptic curves to the server in the Supported Groups extension.
- FCS_TLSS_EXT.1 – The TOE implements TLS as a server.
- FCS_TLSS_EXT.2 – The TOE's TLS server implementation supports mutual authentication for some TLS functions.
- FCS_TLSS_EXT.4 – The TOE's TLS server implementation supports renegotiation of connections.

6.3 User Data Protection

The App PP defines 'sensitive data' as follows: "Sensitive data may include all user or enterprise data or may be specific application data such as emails, messaging, documents, calendar items, and contacts. Sensitive data must minimally include PII, credentials, and keys. Sensitive data shall be identified in the application's TSS by the ST author."

The TSF relies on platform-provided storage mechanisms identified in FCS_STO_EXT.1 to protect credential data at rest in non-volatile storage. Sensitive data also includes log file data, which is protected at rest by VMware VM Encryption.

The only platform computing resources that the TOE requires are the use of network connectivity and system logging.

The TOE interfaces with external components in its operational environment to satisfy its core functionality. The following network interfaces are present in the TSF:

Table 6: TSF Network Usage

Function	Invoked By	Network Port	Secured By
Remote administration	User	TCP/443	One-way or mutual TLS/HTTPS (TOE acts as HTTPS server)
Client user SAML token for authorization	UAG	TCP/443	One-way TLS (TOE acts as HTTPS server)
Agent status and event channel	Agent	TCP/4001	No encryption; data is not sensitive
Connection Server cloud pod communications	TOE or environmental Connection Server	TCP/8472	Mutual TLS (TOE acts as either client or server)
External database connectivity	TOE	TCP/variable (depends on server configuration)	One-way TLS (TOE acts as TLS client)
VMware vCenter	TOE	TCP/443	One-way TLS/HTTPS (TOE acts as HTTPS client)

CRL distribution point or OCSP responder	TOE	TCP/80	No encryption; data is not sensitive

The User Data Protection security function is designed to satisfy the following security functional requirements:

- FDP_DAR_EXT.1 – Sensitive data at rest is protected in turn by the platform's use of VM Encryption and the TSF's use of platform credential storage repositories.
- FDP_DEC_EXT.1 – The TOE's use of platform services is well understood by users prior to authorizing the TOE activity.
- FDP_NET_EXT.1 – The TOE communicates over the network for well-defined purposes. Depending on the function, the use of network resources is user-initiated directly through the TSF, initiated from external applications, or initiated by the TOE itself.

6.4 Identification and Authentication

The TOE uses X.509 to validate the TLS server certificates of the environmental components that it communicates with, as well as TLS client certificates when the TOE is acting as a server for mutually-authenticated TLS connections.

The TOE implements or invokes the following functional behavior for all uses of X.509 certificates:

- Certificate validation and certificate path validation is performed in accordance with RFC 5280.
- The certificate path is checked to ensure that it terminates with a trusted CA certificate.
- The certificate path is validated 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.
- Any CA certificate is validated by ensuring that the key usage field includes the caSigning purpose.
- 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.
- 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.

The TOE relies on the OS platform to validate presented TLS server certificates. The TOE also relies on the OS platform to validate presented TLS client certificates from remote cloud pods. If mutual authentication is configured for the remote administrative interface, the TOE validates the presented TLS client certificate in this case.

The TOE uses both its own functionality as well as platform-provided functionality for X.509 certificate validation, depending on interface, as follows:

- Remote administration: TOE
- Connection Server cloud pod communications: Platform
- External database connectivity: Platform
- VMware vCenter: Platform

Note that for remote administration, only certificate chains with a path length of two are supported.

All revocation checking is done using OCSP in accordance with RFC 6960 or CRL in accordance with RFC 5280 section 6.3 and RFC 8603. In the event that the revocation status of a certificate cannot be verified

(i.e. the OCSP responder cannot be reached or the CRL is stale and a new one is unavailable), the TOE will reject the certificate.

When the TOE's use of the certificate validation function is to validate the authenticity of remote endpoints, the TSF chooses what certificates to use based on what is presented to it as part of establishing the TLS session. When the TOE's use of certificates is to present its own certificate to a remote endpoint, the certificate is chosen based on what the administrator has loaded into the Windows Certificate Store and given an alias that identifies its association with the TOE.

The Identification and Authentication security function is designed to satisfy the following security functional requirements:

- FIA_X509_EXT.1 – X.509 certificates are validated by the TSF or by the TOE platform when establishing trusted communications, depending on the channel.
- FIA_X509_EXT.2 – X.509 certificates are used for TLS. When revocation status of a certificate cannot be determined, the TSF rejects the certificate.

6.5 Security Management

The TOE is run locally as an application on the host platform. The user installing the TOE must be a domain user with local administrator privileges for the installation to be successful. When installing the TOE, the user will specify either their own account as the administrator for the remote web interface or they may specify any domain user or group. Administrative users are authenticated by the OS platform's interaction with an environmental Active Directory server that is used to validate their domain account; the TSF does not provide default credentials.

The TOE is installed by default to %ProgramFiles%\VMware\VMware View. This directory is owned by the Administrator account on the host OS platform, who has write access to them. All other users and groups have read-only access. Most security-relevant configuration data is stored in an environmental LDAP directory in %ProgramData%. This includes policy settings for user sessions on Horizon Agents (session timeout value, login banner, forced logoff warning message), administrator role and privilege assignments, user entitlements to applications and desktops, and user entitlements to desktops. The Windows Registry also stores configuration information related to logging (path to log files, maximum days kept).

The TOE is managed through its web GUI. Some management functions relate to the configuration of the TOE itself, but most relate to the configuration of environmental components. Specifically, the primary purpose of the TOE is to facilitate connectivity between Horizon Clients and Horizon Agents. Configuration then largely revolves around the access that individual Horizon Client users are granted to resources that are managed by Horizon Agents. Listed below are the management functions that are configurable via the GUI along with how they relate to the entire Horizon deployment:

- Log bundle collection – Setting log levels, initiating/cancelling requests to collect logs, downloading log bundles to the local file system, and deleting logs.
    - Configures the behavior of both the TOE by acquiring and deleting logs on the host OS platform that it has the ability to view and of environmental Horizon Agents by configuring what they log.

- Administer policy, including idle session policy – configuring the types of resources that clients are globally allowed/not allowed to interact with and how long an active client session can remain idle before termination.
    - Configures the behavior of the TOE by configuring which Horizon Agent resources it will grant access to on a global basis.
- Allocate roles to administrative users – configuring the administrative levels of privilege used to interact with the TOE's management functionality.
    - Configures the behavior of the TOE by configuring the TOE's own administrative access control.
- Administer entitlements to resources – Configuring the Horizon Client users that are authorized to launch a particular resource on a Horizon Agent system.
    - Configures the behavior of the TOE by configuring which Horizon Agent resources it will grant access to on a per-Horizon Client user basis.
- Helpdesk functions
    - View status of desktop or application sessions
        - Configures the behavior of the Horizon Agent by determining its status.
    - Administration of helpdesk access to desktop resources
        - Configures the behavior of the TOE by determining the helpdesk access that individual TOE administrators are granted.
    - Perform remote assistance to TOE users on connected desktops
        - Configures the behavior of the Horizon Agent by launching an interactive user session on the Agent system.
    - Disconnect and log off desktop or application sessions
        - Configures the behavior of the Horizon Agent by forcing it to terminate active connections with remote Horizon Clients.
    - Restart virtual desktop infrastructure VM
        - Configures the behavior of the operational environment by restarting the virtual machine on which a Horizon Agent resides.
    - Send notification to published desktop or application
        - Configures the behavior of the Horizon Agent by instructing the Horizon Agent to transmit information to a remote Horizon Client.

The Security Management security function is designed to satisfy the following security functional requirements:

- FMT_CFG_EXT.1 – The TOE is protected from direct modification by untrusted users via its host OS platform.
- FMT_MEC_EXT.1 – Configuration settings for the TOE are stored in an appropriate location in its host OS platform.
- FMT_SMF.1 – The TOE has a web GUI for remote management that allows for the execution of a number of management functions.

6.6 Privacy

The TOE's primary function is to connect authenticated Horizon Client users to the Horizon Agent resources that their organizational role entitles them to access. As such, the TOE only receives and handles

username and IP address data from a particular user. The TOE does not have a resource to receive or transmit PII for either a user or administrator.

- FPR_ANO_EXT.1 – The TOE does not transmit PII.

6.7 Protection of the TSF

The TOE implements several mechanisms to protect against exploitation. The TOE implements address space layout randomization (ASLR) through the combination of the use of the /dynamicbase compiler flag for compiled code and the intrinsic memory management of the Java Runtime Environment (JRE) for Java code. The TOE relies fully on its underlying host platform to perform memory mapping. The TOE also does not use both PROT_WRITE and PROT_EXEC on the same memory regions. There is no situation where the TSF maps memory to an explicit address. The TOE is written in C++ and Java. It is compiled with stack overflow protection through the use of the /GS compiler flag.

The TOE is compatible with the security features of Windows Defender Exploit Guard. The TOE uses only documented platform APIs. Appendix A.1 lists the APIs used by each platform version of the TOE. The TOE also makes use of third-party libraries. Appendix A.2 lists the libraries used by each platform version of the TOE. The TOE is versioned using both YYMM date-based versioning to correspond to the approximate release of a particular version and major/minor release versioning, e.g. 2209 refers to the TOE version released on or around September of 2022 and is also synonymous with version 8.7; SWID is not used. The TOE is a standalone application that is not natively bundled as part of a host OS.

The TOE does not have automatic checking for updates or automatic updates; updates to the TOE are obtained through the operational environment (e.g. through the VMware support site). The application version is identified both within the TOE itself in the admin UI as well as through the platform (i.e. the Windows Control Panel). The TOE will not download, modify, replace, or update its own binary code. The TOE is packaged as an .exe file and signed by VMware using 2048-bit RSA. Removing (uninstalling) the product will remove all executable code from the host system.

The Protection of the TSF security function is designed to satisfy the following security functional requirements:

- FPT_AEX_EXT.1 – The TOE interacts with its host OS platform in a manner that does not expose the system to memory-related exploitation.
- FPT_API_EXT.1 – The TOE uses documented platform APIs.
- FPT_IDV_EXT. 1 – The TOE is versioned using YYMM date-based and major/minor versioning.
- FPT_LIB_EXT.1 – The set of third-party libraries used by the TOE is well-defined.
- FPT_TUD_EXT.1 – There is a well-defined method for checking what version of the TOE is currently installed and whether updates to it are available. Updates are signed by the vendor and validated by the host OS platform prior to installation.
- FPT_TUD_EXT.2 – The TOE can be updated through installation packages.

6.8 Trusted Path/Channels

In the evaluated configuration, the TOE uses its own cryptographic implementation to encrypt sensitive data in transit. Listed below are the various external interfaces to the TOE that rely on trusted communications.

- Between remote administrator and TOE
  - Communications use TLS/HTTPS (TOE is server)
  - Implemented using TSF-provided cryptography (Bouncy Castle)
  - The TOE can optionally require a client certificate for mutual TLS authentication
  - TCP port 443
  - Used to gain remote administrative access to the TOE
- Between UAG and TOE
  - Communications use TLS/HTTPS (TOE is server)
  - Implemented using TSF-provided cryptograpy (Bouncy Castle)
  - Not mutually authenticated
  - TCP port 443
  - Used to pass a SAML assertion of an authenticated end user to the TOE so that the TOE can authorize a user connection to one or more Horizon Agents
- Between TOE and other Connection Server (in external cloud pod)
  - Communications use TLS (TOE is client or server)
  - Implemented using TSF-provided cryptography (Bouncy Castle)
  - Mutually authenticated
  - TCP port 8472
  - Used to broker access between a Horizon Client and a Horizon Agent that reside on separate networks
- Between TOE and external database
  - Communications use TLS (TOE is client)
  - Implemented using TSF-provided cryptography (Bouncy Castle)
  - Not mutually authenticated
  - No assigned TCP port
  - Used to record log events of administrator and user activity
- Between TOE and VMware vCenter
  - Communications use TLS/HTTPS (TOE is client)
  - Implemented using TSF-provided cryptography (Bouncy Castle)
  - Not mutually authenticated
  - TCP port 443

- Used to perform VM maintenance activities related to Horizon Agent systems

The Trusted Path/Channels security function is designed to satisfy the following security functional requirements:

- FTP_DIT_EXT.1 – The TOE relies on its own mechanisms to secure sensitive data in transit between itself and its operational environment.

7 Protection Profile Claims

This ST is conformant to the *Protection Profile for Application Software, Version 1.4, October 7, 2021* (App PP) and *Functional Package for Transport Layer Security (TLS), Version 1.1, February 12, 2019* (TLS Package) along with all applicable errata and interpretations from the certificate issuing scheme.

The TOE consists of a software application that runs on a Windows operating system as its platform.

As explained in section 3, Security Problem Definition, the Security Problem Definition of the App PP has been included by reference into this ST.

As explained in section 4, Security Objectives, the Security Objectives of the App PP has been included by reference into this ST.

All claimed SFRs are defined in the App PP and TLS Package. All mandatory SFRs are claimed. Some optional or objective SFRs are claimed. Selection-based SFR claims are consistent with the selections made in the mandatory SFRs that prompt their inclusion.

8 Rationale

This Security Target includes by reference the App PP Security Problem Definition, Security Objectives, and Security Assurance Requirements. The Security Target does not add, remove, or modify any of these items. Security Functional Requirements have been reproduced with the Protection Profile operations completed. All selections, assignments, and refinements made on the claimed Security Functional Requirements have been performed in a manner that is consistent with what is permitted by the App PP and TLS Package. The proper set of selection-based requirements have been claimed based on the selections made in the mandatory requirements. Consequently, the claims made by this Security Target are sufficient to address the TOE's security problem. Rationale for the sufficiency of the TOE Summary Specification is provided below.

8.1 TOE Summary Specification Rationale

This section in conjunction with Section 6, the TOE Summary Specification, provides evidence that the security functions meet the TOE security requirements. Each description includes rationale indicating which requirements the corresponding security functions satisfy. The combined security functions work together to satisfy all of the security requirements. The security functions described in Section 6 are necessary for the TSF to enforce the required security functionality. Table 7 demonstrates the relationship between security requirements and functions.

Table 7: Security Functions vs. Requirements Mapping

	Cryptographic Support	User Data Protection	Identification and Authentication	Security Management	Privacy	Protection of the TSF	Trusted Path/Channels
FCS_CKM_EXT.1	X						
FCS_CKM.1/AK	X						
FCS_CKM.1/SK	X						
FCS_CKM.2	X						
FCS_COP.1/Hash	X						
FCS_COP.1/KeyedHash	X						
FCS_COP.1/Sig	X						
FCS_COP.1/SKC	X						
FCS_HTTPS_EXT.1/Client	X						
FCS_HTTPS_EXT.1/Server	X						
FCS_HTTPS_EXT.2	X						
FCS_RBG_EXT.1	X						
FCS_RBG_EXT.2	X						
FCS_STO_EXT.1	X						
FCS_TLS_EXT.1	X						

	Cryptographic Support	User Data Protection	Identification and Authentication	Security Management	Privacy	Protection of the TSF	Trusted Path/Channels
FCS_TLSC_EXT.1	X						
FCS_TLSC_EXT.2	X						
FCS_TLSC_EXT.5	X						
FCS_TLSS_EXT.1	X						
FCS_TLSS_EXT.2	X						
FCS_TLSS_EXT.4	X						
FDP_DAR_EXT.1		X					
FDP_DEC_EXT.1		X					
FDP_NET_EXT.1		X					
FIA_X509_EXT.1			X				
FIA_X509_EXT.2			X				
FMT_CFG_EXT.1				X			
FMT_MEC_EXT.1				X			
FMT_SMF.1				X			
FPR_ANO_EXT.1					X		
FPT_AEX_EXT.1						X	
FPT_API_EXT.1						X	
FPT_IDV_EXT.1						X	
FPT_LIB_EXT.1						X	
FPT_TUD_EXT.1						X	
FPT_TUD_EXT.2						X	
FTP_DIT_EXT.1							X

A TOE Usage of Third-Party Components

This Appendix lists the platform APIs and third-party libraries that are used by the TOE.

A.1 Platform APIs

Listed below are the platform APIs used by the Horizon Connection Server product.

MSVCP140.DLL
CRYPT32.DLL
ACTIVEDS.DLL
ADVAPI32.DLL
API-MS-WIN-CRT-CONVERT-L1-1-0.DLL
API-MS-WIN-CRT-ENVIRONMENT-L1-1-0.DLL
API-MS-WIN-CRT-FILESYSTEM-L1-1-0.DLL
API-MS-WIN-CRT-HEAP-L1-1-0.DLL
API-MS-WIN-CRT-LOCALE-L1-1-0.DLL
API-MS-WIN-CRT-MATH-L1-1-0.DLL
API-MS-WIN-CRT-PROCESS-L1-1-0.DLL
API-MS-WIN-CRT-RUNTIME-L1-1-0.DLL
API-MS-WIN-CRT-STDIO-L1-1-0.DLL
API-MS-WIN-CRT-STRING-L1-1-0.DLL
API-MS-WIN-CRT-TIME-L1-1-0.DLL
API-MS-WIN-CRT-UTILITY-L1-1-0.DLL
BCRYPT.DLL
CRYPT32.DLL
CRYPTUI.DLL
FWPUCLNT.DLL
IMAGEHLP.DLL
KERNEL32.DLL
LIBIDN.DLL
MPR.DLL
MSWSOCK.DLL
NCRYPT.DLL
NETAPI32.DLL
NTDSAPI.DLL
OLE32.DLL
OLEAUT32.DLL
RPCRT4.DLL
SECUR32.DLL
SHELL32.DLL
SHLWAPI.DLL
USER32.DLL
VCRUNTIME140.DLL
VERSION.DLL
WININET.DLL
WINSCARD.DLL
WLDAP32.DLL
WS2 32.DLL

A.2 Third-Party Libraries

Listed below are the third-party libraries used by the Horizon Connection Server product.

Library	Version
7zip	22.01
accessors-smart	2.4.7
Activation	1.2.0
activation-api	1.2.0
adsddl-vmw	1.10.0.1
annotations-api	N/A
Aopalliance	N/A
apacheds-i18n	2.0.0.AM25
apacheds-kerberos-codec	2.0.0.AM25
api-asn1-api	2.0.0.AM1
api-asn1-ber	2.0.0.AM1
api-i18n	2.0.0.AM1
api-ldap-model	2.0.0.AM1
api-util	2.0.0.AM1
Asm	9.3, 9.1, 6.0
Aspectj	1.9.7
bctls-fips	1.0.11.1
bcutil-jdk15on	1.69
Boost	1.61
Bootstrap	8.5.87
bc-fips	1.0.2.3
bcpkix-fips	1.0.6
byte-buddy	1.12.10
byte-buddy-dep	1.12.10
c-ares	N/A
Catalina	8.5.87
Cglib	N/A
checker-qual	3.5.0
Classmate	1.5.1
commons-beanutils	1.9.4
commons-cli	1.3.1
commons-codec	1.15
commons-configuration	2.8.0
commons-configuration2	2.8.0
commons-daemon	1.3.1
commons-dbcp	2.13.10
commons-digester	N/A

commons-el	N/A
commons-fileupload	N/A
commons-io	2.8.0
commons-lang3	3.12.0
commons-logging	1.2
commons-modeler	N/A
commons-net	3.5
commons-pool	2.7.0
commons-text	1.10.0
commons-validator	1.7
Cryptacular	1.2.4
Cxf	3.4.10
cxf-rt-bindings-soap	3.4.10
cxf-rt-databinding-jaxb	3.4.10
cxf-rt-frontend-jaxws	3.4.10
cxf-rt-frontend-simple	3.4.10
cxf-rt-transports-http	3.4.10
cxf-rt-wsdl	3.4.10
Dnsjava	3.3.0
Dom	1.0
dom4j	2.1.3
el-api	N/A
error_prone_annotations	2.3.4
Failureaccess	1.0.1
Freemarker	2.3.31
Freetype	2.9
google-java-format	1.4
Gson	2.9.0
Guava	30.1-jre
hibernate-validator	6.2.3.Final
HikariCP	4.0.3
horizon-ad-service	1.0.0
http-parser	N/A
httpcomponents-client	4.5.13
httpcomponents-core	4.4.9
httpcomponents-mime	4.3.3
Icu	60
istack-commons	3.0.12
j2objc-annotations	1.3
Jackson	2.13.2
Jackson	2.6.0

jackson-annotations	2.13.2
jackson-databind	2.13.2
jackson-datatype-jdk8	2.13.3
jackson-datatype-jsr310	2.13.3
jackson-module-parameter-names	2.13.3
jakarta.annotation-api	1.3.5
jakarta.validation-api	2.0.2
jasper-el	N/A
java-support	7.2.0
javac-shaded-9-dev-r4023	3
javax.activation	1.2.0
javax.annotation-api	1.3.2
javax.jws-api	1.1
javax.xml.soap-api	1.4.0
Jaxb	2.4.0-b180725.0644
jaxb-api	2.4.0-b180830.0359
jaxws-api	2.3.1
jboss-logging	3.4.3.Final
Jcifs	1.2.6
jcip-annotations	1.0-1
Jcommander	1.48
jctools-core	3.1.0
Jjwt	0.9.0
Jline	0.9.94
Jms	9.3.1
Jmxtools	N/A
Jna	5.5.0
jna-platform	5.5.0
joda-time	2.10.3
json-java	20150729
json-smart	2.4.7
Jsp	2.3.FR
jsp-api	N/A
jsr305	3.0.1
jsr305	3.0.2
jul-to-slf4j	1.7.36
jvm-hotspot-openjdk	11.0.15.0.101u3
kerberos-client	2.0.0.AM25
Lcms	N/A
Libffi	3.2.1
Libidn	1.35

Libpng	1.6.37
log4j-1.2-api	2.17.1
log4j	2.17.1
log4j-api	2.17.1
log4j-slf4j-impl	2.17.1
log4j-to-slf4j	2.17.2
Logback	1.2.11
Mapstruct	1.2.0.Final
Mina	2.0.22
mssql-jdbc	7.4.1.jre8
mx4j	N/A
Namespace	1.4.01
Netty	4.1.90
netty-codec	4.1.90
netty-codec-http	4.1.90
netty-handler	4.1.90
netty-resolver	4.1.90
netty-transport	4.1.90
nghttp2	N/A
nimbus-jose-jwt	9.21
Nodejs	8.17.0
Objenesis	N/A
ojdbc6	N/A
ojdbc8	19.3.0.0
opensaml-java	3.2.0
opensaml-messaging-api	3.2.0
opensaml-messaging-impl	3.2.0
opensaml-profile-api	3.2.0
opensaml-profile-impl	3.2.0
opensaml-saml-api	3.2.0
opensaml-saml-impl	3.2.0
opensaml-security-api	3.2.0
opensaml-security-impl	3.2.0
opensaml-soap-api	3.2.0
opensaml-soap-impl	3.2.0
opensaml-xmlsec-api	3.2.0
opensaml-xmlsec-impl	3.2.0
openssl	1.0.2zg
org.apache.servicemix.bundles.antlr	2.7.7_5
ph-api-client	2.8.338288
ph-api-common	2.8.338288

ph-api-interfaces	2.8.338288
ph-api-serialization	2.8.338288
ph-extension-api	2.8.338288
ph-telemetry-stand-common	2.8.338288
ph-upload-logic	2.8.338288
postgresql	42.3.8
postgresql-jdbc	42.3.8
proxy-autodiscovery	2.8.338288
rapidxml	N/A
saaj	1.5.2
servlet-api	N/A
sesame-model	2.9.0
sesame-rio-api	2.9.0
sesame-rio-turtle	2.9.0
sesame-util	2.9.0
simple	3.1.3-vmware
slf4j	1.7.30
slf4j-log4j12	N/A
snakeyaml	1.32
spring_framework	5.3.20
spring-aop	5.3.20
spring-beans	5.3.20
spring-boot-autoconfigure	2.7.0
spring-context	5.3.20
spring-expression	5.3.20
spring-jdbc	5.3.20
spring-plugin-core	1.2.0.RELEASE
spring-plugin-metadata	1.2.0.RELEASE
spring-security-config	5.6.5
spring-security-core	5.6.5
spring-security-crypto	5.7.1
spring-security-web	5.6.5
spring-tx	5.3.20
spring-web	5.3.20
spring-webmvc	5.3.20
springfox-core	2.9.2
springfox-enum-plugin	1.2.0
springfox-schema	2.9.2
springfox-spi	2.9.2
springfox-spring-web	2.9.2
springfox-swagger-common	2.9.2

springfox-swagger-ui	2.9.2
springfox-swagger2	2.9.2
stax2	3.1.4
Swagger	1.5.20
swagger-annotations	1.5.20
taglibs-standard	1.2.5
Tomcat	8.5.87
tomcat-api	N/A
tomcat-coyote	N/A
tomcat-dbcp	N/A
tomcat-embed-el	9.0.63
tomcat-jni	N/A
tomcat-juli	N/A
tomcat-util-scan	N/A
txw2	2.4.0-b180830.0438
unboundid-ldapsdk	5.1.3
v8	6.2.414.78
visual_studio_runtime	14.16.27033.0
vmw-hzn-log4j-shim	1.0
vmw-hzn-log4j2-binding	1.0
vmw-hzn-logger-common	1.0
vmw-hzn-logger-impl	1.0
woodstox-core-asl	4.4.1
wsdl4j	1.6.3
xerces-j	2.12.2
xml-commons-resolver	1.2
xmlschema	2.2.5
xmlsec-java	2.2.3