CyberArk Software Ltd.

Privileged Access Security – Digital Vault Server

Including Enterprise Password Vault (EPV) v10.4

Security Target

Document Version: 0.17

Prepared for:

CYBERARK

CyberArk Software Ltd.
9 Hapsagot St. Park Ofer 2
P.O.B. 3143
Petach-Tikva 4951040
Israel

Phone: +1 888 808 9005
www.cyberark.com

Prepared by:

Corsec

Corsec Security, Inc.
13921 Park Center Road
Suite 460
Herndon, VA 20171
United States of America

Phone: +1 703 267 6050
www.corsec.com

Table of Contents

List of Figures

List of Tables

1. Introduction

This section identifies the Security Target (ST), Target of Evaluation (TOE), and the organization of the ST. The TOE is the CyberArk Software Ltd. (CyberArk) Privileged Access Security – Digital Vault Server and will hereafter be referred to EPV or as the TOE throughout this document. The TOE is a software-based solution that runs on Windows and is the core component of CyberArk's Privileged Access Security (PAS) Solution. PAS enables organizations to secure, provision, control, and monitor all activities associated with privileged identities used in enterprise systems and applications. EPV securely manages, stores and controls access to privileged accounts.

1.1 Purpose

This ST is divided into 10 sections, as follows:

- Introduction (Section 1) – Provides a brief summary of the ST contents and describes the organization of other sections within this document. It also provides an overview of the TOE security functionality and describes the physical and logical scope for the TOE as well as the ST and TOE references.
- Conformance Claims (Section 2) – Provides the identification of any Common Criteria (CC), Protection Profile (PP), and Evaluation Assurance Level (EAL) package claims. It also identifies whether the ST contains extended security requirements.
- Security Problem (Section 3) – Describes the threats, organizational security policies, and assumptions that pertain to the TOE and its environment.
- Security Objectives (Section 4) – Identifies the security objectives that are satisfied by the TOE and its environment.
- Extended Components (Section 5) – Identifies new components (extended Security Functional Requirements (SFRs) and extended Security Assurance Requirements (SARs)) that are not included in CC Part 2 or CC Part 3.
- Security Assurance Requirements (Section 6) – Presents the SARs met by the TOE.
- Security Functional Requirements (Section 7) – Presents the SFRs met by the TOE.
- TOE Summary Specification (Section 8) – Describes the security functions provided by the TOE that satisfy the SFRs and objectives.
- Rationale (Section 9) – Presents the rationale for the SFR dependencies as to their consistency, completeness, and suitability.
- Acronyms and Terms (Section 10) – Defines the acronyms and terminology used within this ST.

1.2 Security Target and TOE References

Table 1 below shows the ST and TOE references.

Table 1 – ST and TOE References

ST Title	*CyberArk Software Ltd. Privileged Access Security – Digital Vault Server including Enterprise Password Vault (EPV) v10.4 Security Target*
ST Version	Version 0.17
ST Author	Corsec Security, Inc.

ST Publication Date	September 25, 2019
TOE Reference	CyberArk Privileged Access Security – Digital Vault Server including Enterprise Password Vault (EPV) v10.4.1.27

1.3 Product Overview

The product is the CyberArk PAS Solution, which enables organizations to secure, provision, control, and monitor all activities associated with the privileged identities used in enterprise systems.[1] PAS contains multiple applications that work together to provide the following functionality: configure and administer PAS using a web-based interface; store, manage and control access to privileged accounts; establish connections to remote targets using the privileged account credentials; enforce password policy; control access to privileged commands; and record and securely store administrator and session activities.

The PAS software suite includes EPV and other PAS applications. EPV manages the secure storage and access to privileged account files, and to the administrator and session activity files. The privileged account files are used by PAS applications to connect to target machines. PAS applications create and encrypt the privileged account files and send the files to EPV over a TLS connection. In the reverse direction, PAS applications retrieve the privileged account files from EPV. The privileged account files are never decrypted by the EPV. The CyberArk Version Check tool is used to query the current version of PAS software installed on the host and to check if an update is available for the components of the TOE.

Additional PAS applications provide the functionality to create and encrypt the privileged account files, make secure RDP[2] connections to remote targets, provide administrator access to configure PAS over a web-based GUI[3], enforce password policies, make secure SSH connections to remote targets, and control access to privileged commands. The PAS applications described below interact with EPV to provide the complete functionality of the PAS software suite. They are not covered by the evaluation.

Privileged Session Manager (PSM) allows users to retrieve privileged account information from EPV and enables users to log onto remote devices over a secure RDP connection. PSM records the activities that are performed in the privileged session and uploads the recording to EPV, where they are accessed and viewed by authorized users.

Password Vault Web Access (PVWA) enables administrators to access and configure the PAS Solution remotely using a web browser over an HTTPS session. PVWA allows administrators to define access control rules on credentials and platforms, to configure the Master Policies in EPV, and to access and manage privileged accounts on EPV.

Central Policy Manager (CPM) ensures that secure passwords are used and created for all accounts within EPV. An administrator uses the PVWA GUI to configure the policies that CPM enforces as there is no direct access to CPM.

Privileged Session Manager SSH (Secure Shell) Proxy (PSMP) allows users to obtain privileged account information from EPV then logs the user onto a target device over a secured SSH connection. PSMP records the activities that

[1] Note that the components of the PAS Solution were not evaluated as a distributed TOE but as standalone TOEs. This Security Target covers the evaluation of the Enterprise Password Vault.
[2] RDP – Remote Desktop Protocol
[3] GUI – Graphical User Interface

are performed in the privileged session and uploads the recording to EPV, where they are accessed and viewed by authorized users.

On-Demand Privileges Manager (OPM) allows users to obtain privileged account permissions and privileged command access from their local Linux session without obtaining the root credentials or super user access. OPM enables users to granularly access and use privilege accounts according to the command permissions, which are created and managed in EPV.

1.4 TOE Overview

The TOE Overview summarizes the usage and major security features of the TOE. The TOE Overview provides a context for the TOE evaluation by identifying the TOE type, describing the product, and defining the specific evaluated configuration.

The TOE is the CyberArk Privileged Access Security – Digital Vault Server that is referred to as the EPV application, which is compiled with OpenSSL[4] FIPS[5] Object Module v2.0.14[6] and a MySQL v5.6.15 database (DB), and CyberArk Version Check tool. The TOE runs on a Windows operating system (OS) and is the core component of the CyberArk PAS Solution. It manages the storage and access to the privileged account files that are created by other PAS components. The privileged account files, along with each file's unique file key, are encrypted by PAS components and sent to the TOE. For each privileged account file sent to the TOE, the TOE encrypts the unique file key, then stores the privileged account file with its TOE-encrypted file key in a logical safe. Each safe has a unique key, which is used to encrypt the file key of the privileged account file stored within the safe. The encrypted privileged account files, which are sent to and retrieved by the TOE, are never decrypted by the TOE. The TOE also stores and controls access to recordings made by the PAS components. These recordings include administrator activities from PVWA, password policy configuration activities in CPM, privileged sessions between a remote target and PSM or PSMP, and privileged command requests on OPM.

The CyberArk Version Check tool is a script used to query the current versions of the EPV application and checks if an update is available. The tool relies on a file stored in a safe in EPV that will be use to check for the latest version.

The secure storage of the files is based on a logical data structure that stores files (privileged accounts and recordings) in safes that are stored in a Vault. Access to the Vault, safes and the files within is controlled by the TOE's Vault and Safe Access Control Policies. The Vault Access Control Policy controls EPV user access to the Vault. The Safe Access Control Policy controls safe member permissions to view or create safes and their permissions on the files within the safes. The TOE secures and controls access to the privileged credentials based on these policies.

In the evaluated configuration, the TOE runs on a hardened Windows server on an isolated network. The TOE's isolated network includes two additional servers for the other PAS components, an LDAP[7] server and a Certificate Authority (CA) server. The PSM, PVWA and CPM components run on a Windows server. The PSMP and OPM PAS components run on a RHEL server. Communication between the TOE and PAS components is over TLS. The TOE

[4] OpenSSL – Open Secure Sockets Layer
[5] FIPS – Federal Information Processing Standard
[6] Note that the OpenSSL FIPS Object Module is the name of the component created by OpenSSL and used with CyberArk's CAVP-validated cryptographic libraries. It is not meant to imply that this product had completed the CMVP validation
[7] LDAP – Lightweight Directory Access Protocol

interacts with the LDAP server over LDAPS (LDAP/TLS) and plaintext communication with the CA server. See Section 1.5 below for more information about these OE components and interactions.

1.5 TOE Environment

It is assumed that there will be no untrusted users or software on the TOE Server components. In addition, the TOE Server components are intended to be deployed in a physically secured cabinet, room, or data center with the appropriate level of physical access control and physical protection (e.g., badge access, fire control, locks, alarms, etc.).

In the evaluated configuration, the TOE and OE components are on the same network. All the OE components, except the CA Server, communicate with the TOE over TLS v1.2. The TOE interacts with the following: an LDAP server; a CA server; the CPM, PSM, and PVWA PAS components on the Windows server; and the PAS PSMP and OPM components on the RHEL server.

The Windows server PSM, CPM, and PVWA components and the RHEL server PSMP and OPM components authenticate to the TOE using CyberArk authentication. CyberArk authentication is based on the Secure Remote Password (SRP) protocol. There is no direct user access to the TOE. A user can only access the TOE via the PVWA component. The PVWA GUI provides the user interface for access to TOE functionality and protected data. The TOE supports mutual authentication between itself and the trusted OE PAS components and enforces the identification and authentication of users accessing the TOE via the components. The TOE enforces authentication and access control between itself and the Windows server and RHEL server components based on the component IP[8] address, user's Vault credentials, and authorizations. User-entered credentials are securely transmitted to the TOE from the PSM, CPM, and PVWA interfaces on the Windows server and the PSMP and OPM interfaces on the RHEL server. The TOE verifies the credentials locally or submits them to the LDAP server for validation.

Table 2 defines the environmental component requirements.

Table 2 – Environmental Components

Component	Requirements
CA Server	• Microsoft Windows Server 2012 R2[9] OS • Microsoft Active Directory Certificate Services (AD CS)
LDAP Server	• Microsoft Windows Server 2012 R2 OS • Microsoft AD
Vault Server	• Microsoft Windows Server 2012 R2 OS • Intel i7-6700 or Intel Xeon E5 family processor • .NET Framework 4.5.2

[8] IP – Internet Protocol
[9] R2 – Release 2

Component	Requirements
Windows Server	• Microsoft Windows Server 2012 R2 OS • Intel i7-6700 or Intel Xeon E5 family processor • PSM v10.4 software • CPM v10.4 software • PVWA v10.4 software • IIS[10] 8.5 • .NET Framework 4.5.2 • Remote Desktop Services (RDS) Session Host • Remote Desktop Client
RHEL Server	• RHEL v7.4 OS • Intel i7-6700 or Intel Xeon E5 family processor • PSMP v10.4 software ▪ Including PSMP's OpenSSH Client • OPM v10.4 software • RHEL OpenSSH Server Cryptographic Module v5.0 containing OpenSSH Server 7.4p1-11.el7 (included in the RHEL installation) • RHEL OpenSSL Cryptographic Module v5.0 containing OpenSSL 1.0.2k-8.el7 (included in the RHEL installation) • Terminal

1.5.1 LDAP Server

The TOE communications with the LDAP server using LDAPS (LDAP over TLS). The Active Directory accounts on the LDAP server are synched with the accounts on the TOE.

1.5.2 CA Server

The certificates for the Vault server, Windows server and RHEL server are issued by the CA in the operating environment. The communications between the TOE and CA server are used to validate the revocation status of certificates used to secure communications and are sent in plaintext. The TOE validates the Windows server and RHEL server certificates to support the establishment of TLS sessions.

1.5.3 Vault Server

For the evaluated configuration, the TOE must be installed on the Vault Server with the following OS and applications:

- Microsoft Windows Server 2012 R2 SP1 OS
- CyberArk Enterprise Password Vault (EPV) v10.4.1.27 software
- CyberArk Version Check tool v1.5
- .NET Framework 4.5.2

[10] IIS – Internet Information Services

1.5.4 Windows Server

The following components are installed on the Windows Server.

1.5.4.1 CPM

CPM manages privileged account credentials. CPM ensures that passwords of all privileged accounts, such as Windows service accounts, that are stored in the Vault are periodically rotated. All passwords that are managed by CPM conform to the Master Policy defined by the organization. CPM enforces policies by automatically changing passwords and storing the new passwords in the Vault. CPM also synchronizes passwords in Windows service accounts.

1.5.4.2 PSM

PSM serves as a control point to initiate privileged sessions with remote devices without divulging credentials. PSM fetches privileged account information from the Vault and uses the credentials to establish a session with target devices using the target device's native protocol.

1.5.4.3 PVWA

PVWA is a web interface used by administrators for all system administration activities to configure and manage the TOE and by users to request, access, and manage privileged accounts from the PVWA interface. PVWA can search the LDAP directories for users who will be added as Vault users.

1.5.5 RHEL Server

The following components are installed on the RHEL Server.

1.5.5.1 PSMP

PSMP facilitates the creation of privileged sessions without divulging credentials by retrieving the credentials necessary to connect to the remote target Linux devices from the EPV and initiating SSH connections to these devices on behalf of the user. PSMP retrieves the privileged account credentials only if the requestor has been authenticated by the TOE and has permission to access them. Users can connect directly to a target system or device through the PSMP and run specific commands on the target system according to the user's permissions and the allowed commands as defined by the organization's security policy in the Vault. Unauthorized commands will be blocked and will not be sent to the target.

1.5.5.2 OPM

OPM communicates with the TOE to retrieve and update the accounts and policies for the privileged accounts on the requesting Linux system. OPM allows users to obtain privileged account permissions and privileged command access from their local Linux session without obtaining the root credentials or super user access. Command permissions and policies are created and managed by the TOE.

1.6 TOE Description

This section primarily addresses the physical and logical components of the TOE that are included in the evaluation.

1.6.1 Physical Scope

Figure 1 illustrates the physical scope and the physical boundary of the overall solution and ties together all the components of the software-only TOE as well as the constituents of the TOE environment.

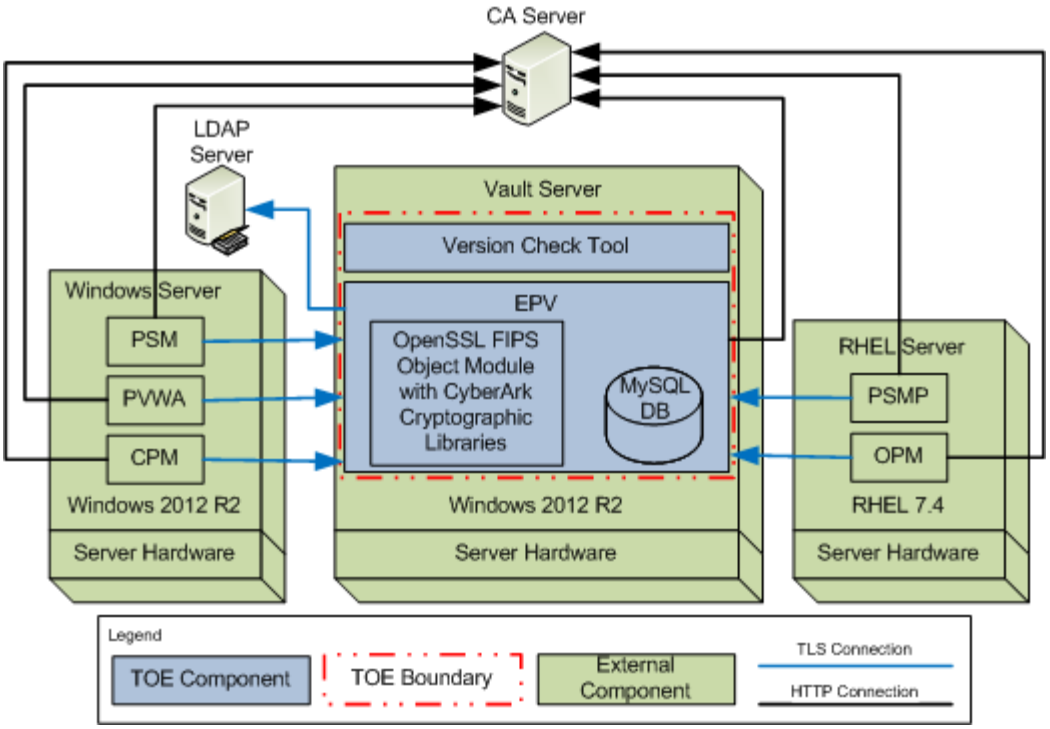

Figure 1 – Physical TOE Boundary

The TOE Boundary includes the CyberArk developed EPV software, Version Check tool, and the third-party software included in the TOE installation package. Any third-party source code or software that EPV has modified is considered TOE software.

1.6.1.1 TOE Software

The TOE is software-only and is comprised of the EPV application, which is compiled with OpenSSL FIPS Object Module v2.0.14 and a MySQL v5.6.15 DB, and the Version Check tool. The TOE software, CyberArk EPV v10.4.1.27 and CyberArk Version Check tool v1.5, must be installed on the Vault server. The installed software is comprised of the applications below.

- EPV is the application software that manages and controls access to the Vault, safes, and sensitive data. The EPV software is compiled with the following libraries:
  - The OpenSSL FIPS Object Module v2.0.14 with the two CyberArk cryptographic libraries, CyberArk PAS Cryptographic Library for Windows v1.0 and CyberArk Privileged Account Security TLS Library for Windows v1.0, provide cryptographic functionality for the TOE. EPV calls the OpenSSL FIPS Object Module v2.0.14 for the cryptographic services that will use the CyberArk libraries when required to secure sensitive data at rest and in transit, and to establish secure communications with other components in the OE.
  - The MySQL v 5.6.15 DB is used for the storage of sensitive data.

- CyberArk Version Check tool v1.5

1.6.1.2 Guidance Documentation

Table 3 lists the TOE Guidance Documentation to install, configure, and maintain the TOE.

Table 3 – Guidance Documentation

Document Name	Description
CyberArk: Privileged Access Security Installation Guide; Version 10.4; PASIN-10-4-0-1	Includes steps for the basic initialization and setup of the TOE.
CyberArk: Privileged Access Security System Requirements; Version 10.4; PASSR-10-4-0-1	
CyberArk; Privileged Access Security End-user Guide; Version 10.4; PASEUG-10-4-0-1	Contains detailed steps for how to properly configure and maintain the TOE.
CyberArk; Privileged Access Security Reference Guide; Version 10.4; PASRG-10-4-0-1	
CyberArk; Privileged Access Security Implementation Guide; Version 10.4; PASIMPG-10-4-0-1	
CyberArk Software Ltd.; Privileged Access Security Enterprise Password Vault Architecture v10.4; Guidance Documentation Supplement; Version: 0.8	Contains information regarding specific configuration for the TOE evaluated configuration.

1.6.2 Logical Scope

The logical boundary of the TOE will be broken down into the following security classes, which are further described in Sections 7 and 8 of this ST. The logical scope also provides the description of the security features of the TOE. The security functional requirements implemented by the TOE are usefully grouped under the following Security Function Classes:

- Cryptographic Support
- User Data Protection
- Identification and Authentication
- Security Management
- Privacy
- Protection of the TOE Security Function (TSF)
- Trusted Path/Channel

1.6.2.1 Cryptographic Support

The TOE implements the OpenSSL FIPS Object Module v2.0.14 with the CyberArk libraries to provide the following cryptographic services: encryption and decryption, hashing, digital signature generation and verification, key generation, and random number generation.

The TOE performs Advanced Encryption Standard (AES) AES 256-bit Cipher Block Chaining (CBC) encryption and decryption to protect sensitive data at rest. TLS is used to protect data in transit between the TOE and the Windows server and RHEL server PAS components. The TOE supports the following algorithms for use in TLS: AES in CBC and Galois Counter Mode (GCM) to encrypt and decrypt data in transit; Elliptic Curve Diffie-Hellman Ephemeral (ECDHE) for key establishment; RSA for signature generation and verification (authentication); and (Hashed Message Authentication Code (HMAC) with SHA-256, and SHA-384 for authentication and message integrity. Mutual authentication between the TOE, Windows server, and RHEL server is performed using x.509

certificates. SP 800-90A AES256-CTR DRBG is used for symmetric key generation and random number generation; key sizes vary depending up the negotiated cipher suite.

1.6.2.2 User Data Protection

The TOE encrypts all sensitive data stored in non-volatile memory. The TOE will limit its access to network connectivity when accessing the platform's hardware resources. The network connection is used for communications between the TOE and its OE components. The connection is user-initiated from a PAS component. The TOE initiates the communication to the LDAP server.

1.6.2.3 Identification and Authentication

The TOE uses and X.509v3 certificates for TLS communications. The certificates are validated by the TOE and used for mutual authentication between the TOE and the Windows server PAS components, and the RHEL server PAS components. The TOE uses a certificate revocation list (CRL) to check the certificate revocation status and will not establish connections to the OE components when the CRL is not available.

1.6.2.4 Security Management

The TOE provides a set of commands for administrators to manage the security functions, configuration, and other features of the TOE and OE components. A TOE administrator manages the TOE from the PVWA TSFI (TOE Security Functional Interface) on the Windows server in the OE.

There is no access to TOE functionality until passwords are created for the built-in Administrator user. During installation, the administrator is prompted to create a password for the for the Administrator user.

1.6.2.5 Privacy

The TOE does not store or transmit any Personally Identifiable Identification (PII).

1.6.2.6 Protection of the TSF

The TOE protects against exploitation as follows:

- Does not map memory to explicit addresses except for OpenSSL functions
- Does not allocate memory regions with write and execute permissions
- Does not write user-modifiable files to directories that contain executable files
- Is compiled with stack-based overflow protection enabled
- Uses standard platform APIs
- Uses only the third-party libraries required for its functionality
- Exceeds the anti-exploitation security features provided by the Windows OS.

The TOE provides integrity for installation and software updates as follows:

- The TOE is packaged in the .msi[11] format and executable files are in the .exe format
- Uninstalling the application removes all traces of it
- The TOE cannot download or make any changes to its binaries
- The TOE version can be verified

[11] msi – Microsoft Installer

- The OS verifies the digital signature of individual executable files in the TOE installation package and software updates before they are installed

The TOE runs a suite of self-tests at power-on and during operation.

1.6.2.7 Trusted Path/Channels

The TOE leverages the platform's functionality to create a trusted channel with the LDAP server. All communications with the LDAP server are encrypted and authenticated over TLS v1.2 sessions using LDAPS (port 636). The TOE provides a trusted path between itself and the PSM, CPM, PVWA, PSMP and OPM PAS components. All communications between the TOE and these components are encrypted and authenticated over TLS v1.2 (port 443) sessions.

1.6.3 Product Physical/Logical Features and Functionality not included in the TOE

Features and Functionality that are not part of the evaluated configuration of the TOE are:

- Functionality provided by CPM
- Functionality provided by OPM
- Functionality provided by PSM
- Functionality provided by PSMP
- Functionality provided by PVWA

1.6.4 Scope of Evaluation

The evaluation is limited in scope to the described in the *Protection Profile for Application Software v1.2; April 22, 2016* (AS PP) and detailed in Section 1.6.2. The TOE is conformant to the AS PP and no interpretations apply to the claims made in this ST.

2. Conformance Claims

This section provides the identification for any CC, PP, Technical Decisions (TD), and EAL package conformance claims. Rationale is provided for any extensions or augmentations to the conformance claims. Rationale for CC and PP conformance claims can be found in Section 9.1.

Table 4 – ApplicableTDs for CC and PP Conformance

Common Criteria (CC) Identification and Conformance	Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 4, September 2012; CC Part 2 extended; CC Part 3 extended; PP claim to the *Protection Profile for Application Software v1.2; April 22, 2016* conformant.
PP Identification	Exact Conformance[12] to the *Protection Profile for Application Software v1.2; April 22, 2016*.
TD Conformance	Conformance to the following TDs is claimed: • 0434 – Windows Desktop Applications Test • 0389 – Handling of SSH EP claim for platform • 0382 – Configuration Storage Options for Apps • 0359 – Buffer Protection • 0358 – Cipher Suites for TLS in SWApp v1.2 • 0327 – Default file permissions for FMT_CFG_EXT.1.2 • 0326 – RSA-based key establishment schemes • 0305 – Handling of TLS connections with and without mutual authentication • 0304 – Update to FCS_TLSC_EXT.1.2 • 0300 – Sensitive Data in FDP_DAR_EXT.1 • 0295 – Update to FPT_AEX_EXT.1.3 Assurance Activities • 0293 – Update to FCS_CKM.1(1) • 0268 – FMT_MEC_EXT.1 Clarification • 0267 – TLSS testing - Empty Certificate Authorities list • 0244 – FCS_TLSC_EXT - TLS Client Curves Allowed • 0241 – Removal of Test 4.1 in FCS_TLSS_EXT.1.1 • 0238 – User-modifiable files FPT_AEX_EXT.1.4 • 0221 – FMT_SMF.1.1 - Assignments moved to Selections • 0217 – Compliance to RFC5759 and RFC5280 for using CRLs • 0192 – Update to FCS_STO_EXT.1 Application Note • 0178 – Integrity for installation tests in APSW PP • 0177 – FCS_TLSS_EXT.1 Application Note Update • 0163 – Update to FCS_TLSC_EXT.1.1 Test 5.4 and FCS_TLSS_EXT.1.1 Test • 0131 – Update to FCS_TLSS_EXT.1.1 Test 4.5 • 0121 – FMT_MEC_EXT.1.1 Configuration Options • 0119 – FCS_STO_EXT.1.1 in PP_APP_v1.2 • 0107 – FCS_CKM - ANSI X9.31-1998, Section 4.1.for Cryptographic Key Generation

[12] Exact Conformance is a type of strict conformance such that the set of SFRs and the SPD/Objectives are exactly as presented within the accepted PP and Extended PP without changes.

3. Security Problem Definition

This section describes the security aspects of the environment in which the TOE will be used and how the TOE is expected to be employed. It provides the statements for the TOE security environment's threats, assumptions, and Organizational Security Policies (OSPs) as identified in the AS PP.

3.1 Threats

Table 5 describes the threats that the TOE is expected to address as defined in the AS PP.

Table 5 – Threats

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

3.2 Assumptions

Table 6 describes the assumptions that are assumed to exist in the TOE's operating environment as defined in the AS PP.

Table 6 – Assumptions

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

3.3 Organizational Security Policies

There are no organizational security policies (OSP) defined in the AS PP.

4. Security Objectives

This section identifies the security objectives for the TOE and its supporting environment.

4.1 Security Objectives for the TOE

Table 7 describes the security objectives that the TOE is required to meet as defined in the AS PP.

Table 7 – Security Objectives for the TOE

Objective	Description
O.INTEGRITY	Conformant TOEs ensure the integrity of their installation and update packages, and also leverage execution environment-based mitigations. Software is seldom if ever shipped without errors, and the ability to deploy patches and updates to fielded software with integrity is critical to enterprise network security. Processor manufacturers, compiler developers, execution environment vendors, and operating system vendors have developed execution environment-based mitigations that increase the cost to attackers by adding complexity to the task of compromising systems. Application software can often take advantage of these mechanisms by using APIs provided by the runtime environment or by enabling the mechanism through compiler or linker options. Addressed by: FDP_DEC_EXT.1, FMT_CFG_EXT.1, FPT_AEX_EXT.1, FPT_TUD_EXT.1
O.MANAGEMENT	To facilitate management by users and the enterprise, conformant TOEs provide consistent and supported interfaces for their security-relevant configuration and maintenance. This includes the deployment of applications and application updates through the use of platform-supported deployment mechanisms and formats, as well as providing mechanisms for configuration. This also includes providing control to the user regarding disclosure of any PII. Addressed by: FMT_SMF.1, FPT_IDV_EXT.1, FPT_TUD_EXT.1.5, FPR_ANO_EXT.1
O.PROTECTED_COMMS	To address both passive (eavesdropping) and active (packet modification) network attack threats, conformant TOEs will use a trusted channel for sensitive data. Sensitive data includes cryptographic keys, passwords, and any other data specific to the application that should not be exposed outside of the application. Addressed by: FTP_DIT_EXT.1, FCS_TLSC_EXT.1, FCS_DTLS_EXT.1, FCS_RBG_EXT.1
O.PROTECTED_STORAGE	To address the issue of loss of confidentiality of user data in the event of loss of physical control of the storage medium, conformant TOEs will use data-at-rest protection. This involves encrypting data and keys stored by the TOE in order to prevent unauthorized access to this data. This also includes unnecessary network communications whose consequence may be the loss of data. Addressed by: FDP_DAR_EXT.1, FDP_DAR_EXT.1, FCS_STO_EXT.1, FCS_RBG_EXT.1
O.QUALITY	To ensure quality of implementation, conformant TOEs leverage services and APIs provided by the runtime environment rather than implementing their own versions of these services and APIs. This is especially important for cryptographic services and other complex operations such as file and media parsing. Leveraging this platform behavior relies upon using only documented and supported APIs. Addressed by: FMT_MEC_EXT.1, FPT_API_EXT.1, FPT_LIB_EXT.1

4.2 Security Objectives for the Operational Environment

Table 8 describes the security objectives that the TOE's operating environment is required to meet as defined in the AS PP.

Table 8 – Security Objectives for the Operational Environment

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

4.3 Security Objectives Rationale

Table 9 describes how the assumptions, threats, and organizational security policies map to the security objectives as defined in the AS PP.

Table 9 – Security Objectives Rationale Mapping

Threat, Assumption, or OSP	Security Objectives	Rationale
T.NETWORK_ATTACK	O.PROTECTED_COMMS, O.INTEGRITY, O.MANAGEMENT	The threat T.NETWORK_ATTACK is countered by O.PROTECTED_COMMS as this provides for integrity of transmitted data. The threat T.NETWORK_ATTACK is countered by O.INTEGRITY as this provides for integrity of software that is installed onto the system from the network. The threat T.NETWORK_ATTACK is countered by O.MANAGEMENT as this provides for the ability to configure the application to defend against network attack.
T.NETWORK_EAVESDROP	O.PROTECTED_COMMS, O.QUALITY, O.MANAGEMENT	The threat T.NETWORK_EAVESDROP is countered by O.PROTECTED_COMMS as this provides for confidentiality of transmitted data. The objective O.QUALITY ensures use of mechanisms that provide protection against network-based attack. The threat T.NETWORK_EAVESDROP is countered by O.MANAGEMENT as this provides for the ability to configure the application to protect the confidentiality of its transmitted data.
T.LOCAL_ATTACK	O.QUALITY	The objective O.QUALITY protects against the use of mechanisms that weaken the TOE with regard to attack by other software on the platform.
T.PHYSICAL_ACCESS	O.PROTECTED_STORAGE	The objective O.PROTECTED_STORAGE protects against unauthorized attempts to access physical storage used by the TOE.
A.PLATFORM	OE.PLATFORM	The operational environment objective OE.PLATFORM is realized through A.PLATFORM.
A.PROPER_USER	OE.PROPER_USER	The operational environment Objective OE.PROPER_USER is realized through A.PROPER_USER.
A.PROPER_ADMIN	OE.PROPER_ADMIN	The operational environment Objective OE.PROPER_ADMIN is realized through A.PROPER_ADMIN.

5. Extended Components

This section defines the extended SFRs and extended SARs met by the TOE.

5.1 Extended TOE Security Functional Components

Table 12 in section 7.2 below identifies the extended SFRs implemented by the TOE. These extended SFRs' definitions are taken directly from the AS PP and are not repeated in this ST.

5.2 Extended TOE Security Assurance Components

Table 10 identifies the extended SARs claimed for the TOE. These extended SARs' definitions are taken directly from the AS PP and are not repeated in this ST.

Table 10 – Extended TOE Security Assurance Components

Name	Description
ALC_TSU_EXT.1	Timely Security Updates

6. Security Assurance Requirements

The AS PP identifies the Security Assurance Requirements (SARs) to frame the extent to which the evaluator assesses the documentation applicable for the evaluation and performs independent testing.

This section lists the set of SARs that are required in evaluations against the AS PP. The AS PP is conformant to Parts 2 (extended) and 3 (extended) of CC V3.1, Revision 4.

The general model for evaluation of TOEs against STs written to conform to PPs is as follows: after the ST has been approved for evaluation, the ITSEF[13] will obtain the TOE, supporting environment (if required), and the guidance documentation for the TOE. The ITSEF is expected to perform actions mandated by the Common Evaluation Methodology (CEM) for the ASE and ALC SARs. The ITSEF also performs the Assurance Activities contained within the AS PP. The Assurance Activities that are captured in the AS PP also provide clarification as to what the developer needs to provide to demonstrate the TOE is compliant with the PP.

The TOE security assurance requirements are identified in Table 11.

Table 11 – Security Assurance Requirements

Assurance Requirements	
Security Target (ASE)	Conformance claims (ASE_CCL.1)
	Extended components definition (ASE_ECD.1)
	ST introduction (ASE_INT.1)
	Security objectives (ASE_OBJ.1)
	Security requirements (ASE_REQ.1)
	Security problem definition (ASE_SPD.1)
	TOE summary specification (ASE_TSS.1)
Development (ADV)	Basic functional specification (ADV_FSP.1)
Guidance documents (AGD)	Operational user guidance (AGD_OPE.1)
	Preparative procedures (AGD_PRE.1)
Life Cycle Support (ALC)	Labeling of the TOE (ALC_CMC.1)
	TOE CM coverage (ALC_CMS.1)
	Timely Security Updates (ALC_TSU_EXT.1)
Tests (ATE)	Independent testing – Conformance (ATE_IND.1)
Vulnerability assessment (AVA)	Vulnerability survey (AVA_VAN.1)

[13] ITSEF – Information Technology Security Evaluation Facility

7. Security Functional Requirements

The individual SFRs are specified in the sections below. SFRs in this section are mandatory SFRs that any conformant TOE must meet. Based on selections made in these SFRs, it will also be necessary to include some of the selection-based SFRs in Appendix B. Optional or Objective SFRs may also be adopted from those listed in Appendix A and Appendix C respectively.

The Assurance Activities defined in AS PP describe actions that the evaluator will take in order to determine compliance of a particular TOE with the SFRs. The content of these Assurance Activities will therefore provide more insight into deliverables required from TOE Developers.

7.1 Conventions

The conventions used in descriptions of the SFRs are as follows:

- Refinement: Indicated with bold text (e.g., [**refinement**]).
- Selection: Indicated with underlined text surrounded by brackets (e.g., [selection]).
- Assignment: Indicated with italicized text surrounded by brackets (e.g., [*assignment*]).
- Assignment within a Selection: Indicated with italicized and underlined text surrounded by brackets (e.g., [*assignment within a selection*]).
- Refinement within a Selection: Indicated with bold and underlined text surrounded by brackets (e.g., [**assignment within a selection**]).
- Iteration: Indicated by appending the iteration number in parenthesis, e.g., (1), (2), (3) and/or by adding a string starting with "/".
- Extended SFRs are identified by having a label 'EXT' at the end of the SFR name.

7.2 Security Functional Requirements

This section specifies the SFRs for the TOE and organizes the SFRs by CC class. Table 12 identifies all SFRs implemented by the TOE and indicates the ST operations performed on each requirement. Note that some column headers use the following abbreviations: S=Selection; A=Assignment; R=Refinement; I=Iteration.

Table 12 – TOE Security Functional Requirements

Name	Description	S	A	R	I
Required SFRs					
FCS_RBG_EXT.1	Random Bit Generation Services	✓			
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	✓	✓		
FMT_CFG_EXT.1	Secure by Default Configuration				

Name	Description	S	A	R	I
FMT_MEC_EXT.1	Supported Configuration Mechanism				
FMT_SMF.1	Specification of Management Functions	✓	✓		
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_LIB_EXT.1	User of Third Party Libraries		✓		
FPT_TUD_EXT.1	Integrity for Installation and Update	✓			
FTP_DIT_EXT.1	Protection of Data in Transit	✓			
Selection-based SFRs					
FCS_CKM.1(1)	Cryptographic Asymmetric Key Generation	✓		✓	✓
FCS_CKM.2	Cryptographic Key Establishment	✓		✓	
FCS_CKM_EXT.1	Cryptographic Key Generation Services	✓			
FCS_COP.1(1)	Cryptographic Operation – Encryption/Decryption	✓			✓
FCS_COP.1(2)	Cryptographic Operation – Hashing	✓			✓
FCS_COP.1(3)	Cryptographic Operation – Signing	✓		✓	✓
FCS_COP.1(4)	Cryptographic Operation – Keyed-Hash Message Authentication	✓	✓		✓
FCS_RBG_EXT.2	Random Bit Generation from Application	✓			
FCS_TLSS_EXT.1	TLS Server Protocol	✓			
FIA_X509_EXT.1	X.509 Certificate Validation	✓			
FIA_X509_EXT.2	X.509 Certificate Authentication	✓			

7.2.1 Class FCS: Cryptographic Support

FCS_CKM.1(1) Cryptographic Asymmetric Key Generation

FCS_CKM.1.1(1)

The application shall [implement functionality] to generate asymmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm

[

- [**ECC schemes**] using [**"NIST[14] curves" P-256, P-384, no other curves**]] that meet the following: [**FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.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:

[

[14] NIST – National Institute of Standards and Technology

- [**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"**],

].

FCS_CKM_EXT.1 Cryptographic Key Generation Services

FCS_CKM_EXT.1.1

The application shall [implement asymmetric key generation].

FCS_COP.1(1) Cryptographic Operation – Encryption/Decryption

FCS_COP.1.1(1)

The application shall perform encryption/decryption in accordance with a specified cryptographic algorithm

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

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

FCS_COP.1(2) Cryptographic Operation – Hashing

FCS_COP.1.1(2)

The application shall perform cryptographic hashing services in accordance with a specified cryptographic algorithm [SHA-256, SHA-384] and message digest sizes [256, 384] bits that meet the following: FIPS Pub 180-4.

FCS_COP.1(3) Cryptographic Operation – Signing

FCS_COP.1.1(3)

The application shall perform cryptographic signature services (generation and verification) in accordance with a specified cryptographic algorithm

[

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

].

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

FCS_COP.1.1(4)

The application shall perform keyed-hash message authentication in accordance with a specified cryptographic algorithm

- HMAC-SHA-256
- and [SHA-384]

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

FCS_RBG_EXT.1 Random Bit Generation Services

FCS_RBG_EXT.1.1

The application shall [implement DRBG functionality] for its cryptographic operations.

FCS_RBG_EXT.2 Random Bit Generation from Application

FCS_RBG_EXT.2.1

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

FCS_RBG_EXT.2.2

The deterministic RBG shall be seeded by an entropy source that accumulates entropy from a platform-based DRBG and [a software-based noise source] with a minimum of [256 bits] of entropy at least equal to the greatest security strength (according to NIST SP 800-57) of the keys and hashes that it will generate.

FCS_STO_EXT.1 Storage of Credentials

FCS_STO_EXT.1.1

The application shall [implement functionality to securely store [*file keys, safe keys, and password verifiers*] to non-volatile memory.

FCS_TLSS_EXT.1 TLS Server Protocol

FCS_TLSS_EXT.1.1

The application shall [implement TLS 1.2 (RFC 5246)] supporting the following cipher suites:

[

- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289]

].

and no other cipher suite.

FCS_TLSS_EXT.1.2

The application shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0, TLS 1.1, and [none].

FCS_TLSS_EXT.1.3

The application shall generate key establishment parameters using [RSA with key size [2048 bits, 3072 bits], ECDHE over NIST curves [secp256r, secp384r] and no other curves, Diffie Hellman parameters of size 2048, [3072 bits]].

FCS_TLSS_EXT.1.4

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

FCS_TLSS_EXT.1.5

The application shall not establish a trusted channel if the peer certificate is invalid.

FCS_TLSS_EXT.1.6

The application shall not establish a trusted channel if the distinguished name (DN) or Subject Alternative Name (SAN) contained in a certificate does not match the expected identifier for the peer.

7.2.2 Class FDP: User Data Protection

FDP_DAR_EXT.1 Encryption of Sensitive Application Data

FDP_DAR_EXT.1.1

The application shall [protect sensitive data in accordance with FCS_STO_EXT.1] in non-volatile memory.

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 [*the firewall and event and system log repositories*].

FDP_NET.EXT.1 Network Communications

FDP_NET_EXT.1.1

The application shall restrict network communication to [user-initiated communication for [*the establishment of TLS sessions with the PAS components and the following functions:*

- *CPM – authenticate to EPV, retrieve and update privileged passwords and password policies*
- *PSM – authenticate to EPV, retrieve privileged accounts, upload privilege session recordings*
- *PVWA – authenticate to EPV, TOE administration*
- *OPM – authenticate to EPV, retrieve and update the privileged accounts and policies*
- *PSMP – authenticate to EPV, retrieve privileged accounts, upload privileged session recordings*],

[*application-initiated TLS communications to the LDAP server for authentication and HTTP connections to the CA server for certification revocation checks*]].

7.2.3 Class FIA: Identification and Authentication

FIA_X509_EXT.1 X.509 Certificate Validation

FIA_X509_EXT.1.1

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

- RFC 5280 certificate validation and certificate path validation.
- The certificate path must terminate with a trusted CA certificate.
- The application shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates.
- The application shall validate the revocation status of the certificate using [*a Certificate Revocation List (CRL) as specified in RFC 5759, a Certificate Revocation List (CRL) as specified in RFC 5280*]
- The application shall validate the extendedKeyUsage field according to the following rules:
    - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
    - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
    - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
    - S/MIME certificates presented for email encryption and signature shall have the Email Protection purpose (id-kp 4 with OID 1.3.6.1.5.5.7.3.4) in the extendedKeyUsage field.
    - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.
    - Server certificates presented for EST[15] shall have the CMC[16] Registration Authority (RA) purpose (id-kpcmcRA with OID 1.3.6.1.5.5.7.3.28) in the extendedKeyUsage field.

FIA_X509_EXT.1.2

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

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 [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].

[15] EST – Enrollment over Secure Transport
[16] CMC– Certificate Management over Cryptographic Message Syntax

7.2.4 Class FMT: Security Management

FMT_CFG_EXT.1 Secure by Default Configuration

FMT_CFG_EXT.1.1

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

FMT_CFG_EXT.1.2

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

FMT_MEC_EXT.1 Supported Configuration Mechanism

FMT_MEC_EXT.1.1

The application shall invoke the mechanisms recommended by the platform vendor for storing and setting configuration options.

FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1

The TSF shall be capable of performing the following management functions [no management functions].

7.2.5 Class FPR: Privacy

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].

7.2.6 Class FPT: Protection of the TSF

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 [

- 0x00000000FB000000
- 0x0000001800000000

].

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 compiled with stack-based buffer overflow protection enabled.

FPT_API_EXT.1 Use of Supported Services and APIs

FPT_API_EXT.1.1

The application shall use only documented platform APIs.

FPT_LIB_EXT.1 User of Third-Party Libraries

FPT_LIB_EXT.1.1

The application shall be packaged with only [*the third-party libraries listed in* Table 13 *below*].

Table 13 – TOE's Third-Party Libraries

Third-Party Library	Version	Vendor
AutoMapper.dll	1.1.0.118	Jimmy Bogard
Castle.Core.dll	3.3.3.58	Castle Project
Castle.Windsor.dll	3.3.0.51	Castle Project
Dapper.StrongName.dll	1.50.2.0	Sam Saffron
FluentNHibernate.dll	1.3.0.0	James Gregory and contributors (Paul Batum, Hudson Akridge et al)
lesi.Collections.dll	3.3.0.4000	Aidant Systems
libmysql.dll	5.6.15	MySQL
libmysql_x64.dll	6.1.2.0	MySQL
msvcp120.dll	12.0.21005.1	Microsoft Corporation
msvcr71.dll	7.10.6030.0	Microsoft Corporation
MySql.Data.dll	6.4.4.0	Oracle
MySql.Data.dll	6.9.9.0	Oracle
NHibernate.dll	3.3.0.4000	LGPL
NHibernate.XmlSerializers.dll	3.3.0.4000	LGPL
PowerCollections.dll	N/A	Wintellect
python27.dll	2.7.6150.1013	Python Software
System.Data.SQLite.dll	1.0.66.0	SQLite
Xalan-C_1_11.dll	1.11.0.1	Apache Software Foundation
Xalan-C_1_11_x64.dll	1.11.0.1	Apache Software Foundation
XalanMessages_1_11.dll	1.11	Apache Software Foundation
XalanMessages_1_11_x64.dll	1.11	Apache Software Foundation
xerces-c_3_1_4.dll	3.1.4.0	Apache Software Foundation
xerces-c_3_1_4_x64.dll	3.1.4.0	Apache Software Foundation

FPT_TUD_EXT.1 Integrity for Installation and Update

FPT_TUD_EXT.1.1

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

FPT_TUD_EXT.1.2

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

FPT_TUD_EXT.1.3

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.1.4

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

FPT_TUD_EXT.1.5

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

FPT_TUD_EXT.1.6

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

7.2.7 Class FTP: Trusted Path/Channel

FTP_DIT_EXT.1 Protection of Data in Transit

FTP_DIT_EXT.1.1

The application shall [

- encrypt all transmitted sensitive data with [TLS]
- invoke platform-provided functionality to encrypt all transmitted sensitive data with [TLS]

] between itself and another trusted IT product.

8. TOE Summary Specification

This section presents information to detail how the TOE meets the functional requirements described in previous sections of this ST.

8.1 TOE Security Functionality

Each of the security requirements and the associated descriptions correspond to the security functions. Hence, each function is described by how it specifically satisfies each of its related requirements. This serves to both describe the security functions and rationalize that the security functions satisfy the necessary requirements.

Table 14 – Mapping of TOE Security Functionality to Security Functional Requirements

TOE Security Function	SFR ID	Description
Cryptographic Support	FCS_CKM.1(1)	Cryptographic Asymmetric Key Generation
	FCS_CKM.2	Cryptographic Key Establishment
	FCS_CKM_EXT.1	Cryptographic Key Generation Services
	FCS_COP.1(1)	Cryptographic Operation - Encryption/Decryption
	FCS_COP.1(2)	Cryptographic Operation - Hashing
	FCS_COP.1(3)	Cryptographic Operation - Signing
	FCS_COP.1(4)	Cryptographic Operation - Keyed-Hash Message
	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_TLSS_EXT.1	TLS Server Protocol
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
Identification and Authentication	FIA_X509_EXT.1	Certificate Validation
	FIA_X509_EXT.2	Certificate Authentication
Security Management	FMT_CFG_EXT.1	Secure by Default Configuration
	FMT_MEC_EXT.1	Supported Configuration Mechanism
	FMT_SMF.1	Specification of Management Functions
Privacy	FPR_ANO_EXT.1	User Consent for Transmission of Personally Identifiable Information
Protection of the TSF	FPT_AEX_EXT.1	Anti-Exploitation Capabilities
	FPT_API_EXT.1	Use of Supported Services and APIs
	FPT_LIB_EXT.1	User of Third Party Libraries
	FPT_TUD_EXT.1	Integrity for Installation and Update

TOE Security Function	SFR ID	Description
Trusted Path / Channels	FTP_DIT_EXT.1	Protection of Data in Transit

8.1.1 Cryptographic Support

The TOE implements the OpenSSL FIPS Object Module v2.0.14 with the CyberArk libraries to provide the required algorithms for all cryptographic operations. Each of the cryptographic algorithms supported by the TOE have been tested and certified by the CAVP. See Table 15 below for the cryptographic operations implemented by the TOE.

Table 15 – Cryptographic Algorithms and Key Sizes

Cryptographic Operation	Usage	Algorithm	Key Sizes (bits)	Certificate
Encryption/Decryption	Secure Storage	AES-CBC	256	CAVP 5487 and C1088
	TLS	AES-CBC and GCM	128, 256	
Key Generation	Safe	AES CTR-DRBG	256	CAVP 5487 and C1088
Signature Generation Signature Verification	TLS	RSA	2048, 3072	CAVP 2948 and C1088
Key Exchange /Establishment	TLS	ECDHE	256, 384	CAVP 1945 and C1088
Message Digest	TLS	SHA-256, SHA-384	256, 384	CAVP 4404 and C1088
Message Authentication	TLS	HMAC- SHA-256, SHA-384	256, 384	CAVP 3645 and C1088
Random Number Generation	TOE DRBG	CTR DRBG (AES)	N/A[17]	CAVP 2162 and C1088

FCS_CKM.1(1) / FCS_CKM_EXT.1 / FCS_CKM.2

Table 15 above lists all the key sizes used for the ECC asymmetric key generation scheme and its usage. Table 15 also lists the key establishment and key exchange schemes used by the TOE. The TOE uses ECDHE key establishment/exchange for TLS. The use of asymmetric encryption is needed for the TLS protocol used by the TOE. The key generation methods follow the requirements within FIPS PUB 186-4. The key establishment methods follow the requirements within NIST Special Publication 800-56A and NIST Special Publication 800-56B

FCS_COP.1(1)

AES 128-bit and 256-bit symmetric keys are used to encrypt/decrypt the TLS communications between the TOE and PAS components. AES is supported in CBC and GCM modes and used with 128-bit and 256-bit keys.

AES256-CBC is used for the encryption/decryption of sensitive data stored in non-volatile memory.

FCS_COP.1(2) / COP.1(4)

Table 16 lists all the key sizes used for SHA hashing and message digests within the TOE. Usages of SHA is limited to TLS and for SRP. The SHA256 and SHA384 hash function are used in HMAC for TLS message integrity and authentication. The TOE's implementation of SHA follows the requirements within FIPS Pub 180-4.

[17] N/A – Not Applicable

Table 16 – HMAC Properties

Hash Function	Block Size	Key Length	Output (Digest)
SHA256	512	256	256
SHA384	1024	384	384

FCS_COP.1(3)
Table 15 lists all the key sizes used for signature generation and verification for TLS and the key sizes used to verify TOE file signatures. The TOE's implementation of signature generation and verification follow the requirements within FIPS PUB 186-4.

FCS_RBG_EXT.1/FCS_RBG_EXT.2
The TOE implements the Approved SP 800-90 Approved AES256-CTR DRBG to generate random bits for key generation. When the TOE starts up, the DRBG is seeded with 256 bits of entropy from the Windows Entropy Pool by calling the OpenSSL RAND_seed function for the CryptGenRandom function and for Crypto API (CAPI). The platform system time and tick count noise sources are added to the Windows OS Entropy Pool after initialization. On an ongoing basis the TOE seeds the DRBG with 256 bits of entropy by calling the RAND_seed function for the BCryptGenRandom function and for the CNG (Crypto Next Generation) API. More information about the entropy process is described in the proprietary Entropy Rationale document.

FCS_STO_EXT.1
The TOE secures sensitive data stored in non-volatile memory using its algorithms for AES256-CBC encryption with a 256-bit key. Sensitive data includes the file key sent with a file from a PAS component, the safe key used to encrypt the file key, and the verifier associated with a CyberArk password (for CyberArk authentication).

The privileged account file sent by a PAS client is encrypted by the PAS client. The encrypted file is sent to the TOE and the file key is sent along securely in encrypted form over TLS. The TOE decrypts the file key, then encrypts the file key with the safe's unique AES 256-bit key using AES256-CBC encryption. The safe key is encrypted by the unique AES 256-bit Server key and stored within the safe. A safe key is generated automatically using the DRBG when a safe is created.

An administrator creates the initial password for a CyberArk (local) account. When the administrator creates the initial password, or a user changes it, the password is concatenated and manipulated using hash and exponential functions to derive a password verifier. The password verifier is stored in the MySQL DB. In the MySQL DB, the column containing the verifier is encrypted with the Server key using AES256-CBC encryption. Anytime a local user authenticates, the password verifier is derived and authenticated against the value stored in the DB.

The Server key is unique to the TOE and is stored in volatile memory. The Server key is used to encrypt the safe keys and the sensitive data stored within the DB. A safe key is used to encrypt one or more files within a safe.

The Administrator user, and the PAS component users listed below, authenticate to the TOE using CyberArk authentication.

- CPM – PasswordManager
- PVWA – PVWAAppUser, PVWAGWUser
- PSM – PSMAPPUSer, PSMGWUser
- PSMP – PSMPAppUser, PSMPGWUser

- OPM – OPMUser

FCC_TLSS_EXT.1
The TOE is a server to the OE PAS component clients. The TOE uses the OpenSSL FIPS Object Module v2.0.14 with the CyberArk libraries for the cryptographic services required to support TLS communications with the PAS component clients. X.509 certificates are used for mutual authentication between the TOE and PAS clients and when establishing the TLS session. The TOE supports the cipher suites listed below.

- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

The TOE does not accept any connections requests using SSL or a TLS version other than TLSv1.2. The TOE checks that the presented identifier matches the reference identifier, either the IP or DNS name, and only establishes a trusted channel if the identifier is a match and on the client's certificate is validated. The TOE supports certificate pinning.

TOE Security Functional Requirements Satisfied: FCS_CKM.1(1), FCS_CKM.2, FCS_CKM_EXT.1, FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_RBG_EXT.1, FCS_RBG_EXT.2, FCS_STO_EXT.1, FCS_TLSC_EXT.1, FCS_TLSC_EXT.4, FCS_TLSS_EXT.1

8.1.2 User Data Protection

FDP_DAR_EXT.1
Except for cryptographic key destruction for keys stored in volatile memory, the TOE does not depend on platform provided cryptographic functionality to provide its cryptographic services; it is included with CyberArk Crypto Library, which provides all cryptographic services including encryption/decryption of data stored in safes and in the MySQL DB. The TOE protects sensitive data by using AES256-CBC to encrypt the data before storing it in non-volatile memory and restricting access to the data. Sensitive data includes the file key used to encrypt a file sent by a PAS client, the safe key used to encrypt the file key, and the password verifier used for SRP authentication to the TOE. The file key is encrypted by the safe key of the safe where it is stored. Both the client file and its encrypted file key are stored within the safe. All sensitive data stored within a safe is protected by that safe's key. The safes are stored in non-volatile memory at c:\Private\Safes. The safe key is encrypted with the 256-bit Server key using AES256-CBC encryption. The Server key is stored in volatile memory. The Administrator user and PAS components use SRP authentication. PAS client password verifiers are encrypted with the Server key using AES256-CBC encryption and stored in the MySQL DB.

The sensitive data within a safe is protected by the combination of the Vault Access Control Policy, which is configured by the installation process, and the Safe Access Control Policy. Access to the Vault and safes is enforced by user account authorizations and permissions. The Vault Access Control Policy controls EPV user access to the Vault. The Vault Access Control Policy only allows access to the Vault for those EPV users that are defined in the Vault.ini file. The Safe Access Control Policy controls safe member permissions to view or create safes and their permissions on the files within the safes. An operator attempting to access the Vault or safes with the incorrect authorizations and permissions is denied access.

FDP_DEC_EXT.1 and FDP_NET_EXT.1
The TOE limits its access to only network connectivity when accessing the platform's hardware resources. The TOE requires network access to the CA server, LDAP server, the Windows server PAS components, and the RHEL server PAS components. The TOE limits access to only network connectivity between the PAS component clients and the TOE over TLS on port 443 and between the TOE (as the client) and LDAP server over LDAPS on port 636. The TOE will also use port 80 for HTTP connections to the CA server for certification revocation checks.

The TOE limits access to the platform's firewall services and audit mechanism. The TOE hardening process closes all ports and removes services not required by the TOE. The TOE accesses the platform's firewall to take control over the firewall services and change the firewall information flow control rules. The TOE also accesses the platform's audit mechanism to write event and system logs.

TOE Security Functional Requirements Satisfied: FDP_DAR_EXT.1, FDP_DEC_EXT.1, FDP_NET_EXT.1

8.1.3 Identification and Authentication

FIA_X509_EXT.1
The TOE provides its own implementation of TLS to perform certificate validation. The Windows server and RHEL server are clients to the TOE. The TOE validates each of the clients' X.509 certificates during TLS authentication. The TOE ensures that the X.509 certificate adheres to RFC 5280 (certificate validation and certificate path validation) and that the certificate path terminates with a trusted CA certificate. The TOE treats a certificate as a CA certificate when the certificate includes the basicConstraints extension and verifies that the CA flag is set to "TRUE" for all CA certificates. The TOE validates the revocation status of each client's TLS certificate according to RFC 5759 using a CRL when establishing the TLS connection. The CRL is downloaded from the CA server in the operating environment. The path to the CRL is read from the certificate's CRL Distribution Point (CDP) field. The TOE checks each of the client's certificate against the downloaded CRL and automatically rejects the certificate if it is found to be invalid. When a TLS v1.2 connection cannot be established because the validity check of a certificate fails, the connection is aborted. The TOE validates that the Windows server and RHEL server certificates presented for TLS have the Client Authentication purpose in the extended key usage field.

The TOE does not accept OCSP, S/MIME[18] or EST certificates. The TOE supports a maximum trust depth of two nodes.

FIA_X509_EXT.2
The TOE uses X.509v3 certificates as defined by RFC 5280 to support authentication. It will read the location of the certificate from the dbparm.ini files using the ServerCertificateFile parameter.

The TOE validates X.509v3 certificates from the Windows server and RHEL server for TLS authentication. The path to the CRL is read from the certificate's CDP field. The TOE will reject a certificate if it is found to be invalid according to the requirements in FIA_X509.1. If the TOE cannot establish a connection to validate a certificate, the TOE does not establish a connection. The connection from the TOE to the CDP is conducted over HTTP as per the RFC.

[18] S/MIME – Secure Multipurpose Internet Mail Extensions

TOE Security Functional Requirements Satisfied: FIA_X509_EXT.1, FIA_X509_EXT.2

8.1.4 Security Management

FMT_CFG_EXT.1
Physical access is required for installation of the TOE. The TOE provides only enough functionality to enter credentials for the Administrator and Master users during installation. There are no default credentials for these users and no other default credentials stored on the TOE. Only an authorized administrator can install the TOE and set the credentials. During installation the TOE is configured by default to protect the application's files from unauthorized access. The files are set with permissions that do not allow the "Users" group to modify them.

FMT_MEC_EXT.1
The TOE uses OS functionality for storing and setting configuration options. The storage location of configuration files is maintained in the Windows Registry. The Server Windows Registry entries are located in the following file: HKLM\Software\CyberArk\PrivateArk\Server\<version>. The Client Windows Registry settings are found at HKEY_LOCAL_MACHINE\SOFTWARE\CyberArk\PrivateArk\Client. The *CyberArk Privileged Access Security Reference Guide v10.4* contains detailed information about the registry and configuration file settings.

The TOE contains local configuration files that are created during installation, but the information is read-only and never written to by the TOE. The Administrator user or users in the Administrators group have Full control, modify, Read & Execute, Read, and Write permissions for the configuration files. The configuration files are located in the "C:\Program Files (x86)\PrivateArk\Server\" folder. The DBParm.ini configuration file contains the general parameters for the Vault database. It contains parameters for cryptographic algorithms, key locations, certificate settings, groups and users, and the TOE's listening port. The Passparm.ini file contains the password complexity settings.
FMT_SMF.1
The TOE does not provide any direct management functionality. Any management operations must be performed by an authorized administrator using PVWA in the environment.

TOE Security Functional Requirements Satisfied: FMT_CFG_EXT.1, FMT_MEC_EXT.1, FMT_SMF.1

8.1.5 Privacy

FPR_ANO_EXT.1
The TOE does not transmit PII. Usernames were considered and determined to not be PII as this information is owned and generated by the company that implements the TOE. This means that the users would not be able to pick their own personal username that can link to their personal identity.

TOE Security Functional Requirements Satisfied: FPR_ANO_EXT.1

8.1.6 Protection of the TSF

FPT_AEX_EXT.1
The TOE provides anti-exploitation protections. By default, ASLR[19] protection is enabled on the Windows 2012 R2 server. The TOE is compiled using the /NXCOMPAT flag to enable Data Execution Protection (DEP) and the /GS flag to enable stack-based buffer overflow protection.

In FIPS mode, during self-check, the OpenSSL *.dll[20] files, calibeay64102k.dll and cassleay64102k.dll, are written to their respective memory addresses: 0x00000000FB000000 and 0x0000001800000000. This mapped memory area has execute permissions but no write permission once the files are loaded.

The TOE does not write user-modifiable files to directories that contain executable files. Executable files are stored in ...\PrivateArk\Server\. User-modifiable files are written to ...\PrivateArk\Server\Conf and ...\PrivateArk\Server\Logs.

The TOE is installed on a hardened operating system based on Microsoft Bastion Host server recommendations. The TOE hardening is part of the installation and results in disablement of many operating system services. The hardening process also strips the permissions from existing and built-in Windows users (except the user that runs the installation). For more information about the hardening process, please refer to the CyberArk *Installation Guide* and the script used to perform the hardening.

Pre-Installation and OS hardening:

- The Windows OS is installed on a dedicated physical Windows server 2012 R2 platform.
- The OS is configured for the specific needs of the TOE software.
- The server is configured in a workgroup; it is not a Windows domain member.
- The server is assigned a static IP.
- No third-party software other than platform applications required for the TOE are installed on the server.

While EMET is not enabled on the platform, the following OS changes are made during the TOE installation and hardening procedure. Please see the hardening script referenced above for the complete list of all settings and their values.

- Operating System services that are not required for the operation of the TOE are deactivated
- The Windows Firewall is configured to block all traffic except for allowed protocols and ports
- Password complexity settings are set
- Privilege rights are set
- Registry values are changed to enable the FIPSAlgorithmPolicy, enable Authenticode, ForceKeyProtection, and many others
- General service values are specified

FPT_API_EXT.1
The TOE uses only the standard platform APIs listed in Appendix A.

[19] ASLR – Address Space Layout Randomization.
[20] dll – Dynamic Link Library

FPT_LIB_EXT.1.
The TOE is packaged with the third-party libraries required for its functionality. See Table 13 in Section 7.2.6 above for the list of libraries.

FPT_TUD_EXT.1
The CyberArk Version Check tool is downloaded to the platform as part of the TOE and is used for checking updates to the TOE. It relies on a file uploaded to the Vault server that contains all the current version information for the CyberArk PAS suite. The TOE administrator will need to upload this file once per version of PAS and can be used for all components of PAS. For local storage purposes, the TOE administrator will also need to upload the update packages to the vault to allow for an internal update repository. An email notification from CyberArk will be sent to the TOE administrator when a new version is available. The TOE administration that receives the email notification is responsible for uploading the files to the Vault server. The information in the email will contain the links to the appropriate download locations and the release notes related to the update. Since multiple components of the PAS solution check for updates against this central location, the administrator that uploads the files to the Vault server is responsible for maintaining accuracy of all component versions.

The TOE administration must run the Version Check tool whenever they need to check for a new version. This can be done periodically or when notified.

If an update is available for the TOE, the TOE administrator will download the latest version of the TOE software from the Safe on the Vault server. The package will contain the required executable (.exe) files for the TOE's platform. The current version of TOE software is returned after running the script. To determine the currently installed version without running the above script, the administrator can check for the TOE software in the Programs and Features manager. The TOE will not automatically download or apply new packages that would replace or update its code.

The TOE installation and configuration files are all packaged into a zip file that is digitally signed by CyberArk. To verify the digital signature of a TOE package, users must complete the following:

1. Download the TOE installation package from CyberArk.
2. Download and install the Java Development Kit (JDK) from Oracle.
3. Download and install the JCE Unlimited Strength Jurisdiction Policy Files.
4. Run the following command without quotes, "%JDK_Home%\jarsigner.exe -verify -verbose -certs <filename>.zip". More information about the jarsigner's options can be found at https://docs.oracle.com/javase/7/docs/technotes/tools/windows/jarsigner.html#CCHFIDAB.

Individual TOE files are signed using the Windows OS package manager MS[21] Sign tool. To verify the integrity of the TOE installation file, complete the following:

1. Extract the files from the archive file.
2. Navigate to the *setup.exe* file.
3. Right-click the file, then Click on **Properties > Digitals Signatures**.
4. The "**CyberArk Software Ltd.**" signer should be selected. Click on **Details**, and then verify the signature details.

[21] MS – Microsoft

The TOE relies on the platform's package manager to make changes to the binary code. Installation of the updates is performed by an administrator while using the executable file (.exe) extracted from the archive file (.zip). The TOE software can be removed from the platform using the platform's Programs and Features manager. Uninstallation of the TOE will remove all traces of the application except for configuration settings, output files, and audit/log events.

TOE Security Functional Requirements Satisfied: FPT_AEX_EXT.1, FPT_API_EXT.1, FPT_API_EXT.2, FPT_LIB_EXT.1, FPT_TUD_EXT.1

8.1.7 Trusted Path/Channels

The TOE protects data in transit by providing trusted paths and channels using the cryptographic functions within the TOE's CyberArk PAS Cryptographic libraries.

The TOE leverages the Windows library wldap32.dll to create a trusted path between itself and the LDAP server. Communications between the TOE and LDAP server are protected by LDAPS. The TOE is a client to the LDAP server and there is a single path between them using TCP port 443, which is specified during installation.

Communications between the TOE and Windows server's CPM, PSM, PVWA PAS components and between the TOE and RHEL server's PSMP and OPM PAS components are protected by TLS. The TOE is a server to the Windows server's CPM, PSM, PVWA components and to the RHEL server's PSMP and OPM PAS components. There is a single channel between the components and the TOE using TCP port 443.

TOE Security Functional Requirements Satisfied: FTP_DIT_EXT.1

8.1.8 Timely Security Updates

Upon discovery of a security vulnerability in any of CyberArk's products, underlying systems, or embedded 3rd-party libraries, a vulnerability assessment process commences and may vary depending on the vulnerability characteristics.

CyberArk reviews all OS updates to determine if they are applicable to the TOE. Because the TOE platform is hardened, CyberArk reviews all OS updates to determine if they are applicable to the TOE and notifies customers with update instructions as needed. Likewise, CyberArk has no control over third-party patches or updates but will incorporate any necessary third-party updates into a TOE update and notify the customer.

Typical activities resulting from the vulnerability assessment may include (depending on their severity):

- Release of a software patch that addresses the vulnerability.
- Issue a Security Bulletin or other notice to affected customers that discloses the vulnerability and mitigation information.
- Applying necessary security enhancement to the product roadmap.

The following table outlines the steps of the vulnerability assessment process. Some of these steps may take place in parallel:

1. Severity Review: Assessing the vulnerability's severity ranking.
    a. For 3rd-party libraries, review publicly available security rankings and analyses.
2. Mitigation Analysis: Evaluate whether there is a mitigation option (even temporary) that could reduce the severity of the vulnerability until it is permanently fixed.
3. Fix Assessment: Provide time and effort estimation for suggested fix.
4. Vulnerability Addressed: Addressing the vulnerability according to its Service Level Agreement (SLA).

CyberArk addresses the identified vulnerabilities within the following SLA in correlation with the severity and business risk rating:

- Critical
    - Response Time: Immediate (from time of analysis completion). Dependent on fix complexity (may take up to 90 days)
    - Covered Versions: All effected versions within their End of Development period
- High
    - Response Time: Next planned release cadence
    - Covered Versions: Latest version
- Medium / Low
    - Response Time: Added to roadmap and addressed within one of the next releases
    - Covered Versions: Latest version

Security issues can be reported to this CyberArk website: https://www.cyberark.com/product-security/. Anyone reporting a security issue will be given a set of keys for encrypting the data for transfer. Current security bulletins may also be viewed from the same URL.

Customers will receive emails related to available updates that contain the link to download the latest software for the TOE. Updates may also be downloaded from the CyberArk Support Vault website: https://support.cyberark.com/.

9. Rationale

9.1 Conformance Claims Rationale

This Security Target extends Part 2 and extends to Part 3 of the Common Criteria Standard for Information Technology Security Evaluations, Version 3.1 Revision 4. This ST conforms to the AS PP.

9.1.1 Variance Between the PP and this ST

There is no variance between the AS PP and this ST.

9.1.2 Security Assurance Requirements Rationale

The assumptions, threats, OSPs, and objectives defined in this ST are those specified in the AS PP. This ST maintains exact conformance to the AS PP, including the assurance requirements listed in Section 5 of the AS PP. The TOE is a standalone application that runs on a Windows Server platform and is applicable to the AS PP.

10. Acronyms and Terms

This section describes the acronyms and terms used throughout the document.

10.1 Acronyms

Table 17 defines the acronyms used throughout this document.

Table 17 – Acronyms

Acronym	Definition
AD	Active Directory
ADV	Development Documentation
AD CS	Active Directory Certificate Services
AES	Advanced Encryption Standard
AGD	Guidance Documents
ANSI	American National Standards Institute
API	Application Programming Interface
AS PP	Protection Profile for Application Software v1.2; April 22, 2016
CA	Certificate Authority
CBC	Cipher Block Chaining
CC	Common Criteria
CDP	Certificate Revocation List Distribution Point
CEM	Common Evaluation Methodology
CM	Configuration Management
CMC	Certificate Management over Cryptographic Message Syntax
CPM	Central Password Manager
CRL	Certificate Revocation List
CTR	Counter Mode
DEP	Data Execution Protection
DH	Diffie-Hellman
DHE	Diffie-Hellman Ephemeral
DNS	Domain Name System
DRBG	Deterministic Random Bit Generator
DSA	Digital Signature Algorithm
DSS	Digital Signature Standard
EAL	Evaluation Assurance Level
EC	Elliptic Curve

Acronym	Definition
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie-Hellman
ECDHE	Elliptic Curve Diffie-Hellman Ephemeral
EPV	Enterprise Password Vault
FFC	Finite Field Cryptography
FIPS	Federal Information Processing Standard
FSP	Functional Specification
GCM	Galois Counter Mode
GUI	Graphical User Interface
HMAC	Hash-based Message Authentication Code
HTTPS	Hypertext Transfer Protocol Secure
ICU	International Components for Update
ID	Identification
IIS	Internet Information Services
IP	Internet Protocol
IT	Information Technology
LDAP	Lightweight Directory Access Protocol
LDAPS	Lightweight Directory Access Protocol over TLS
ITSEF	Information Technology Security Evaluation Facility
MAC	Media Access Control
MIME	Multipurpose Internet Mail Extensions
MS	Microsoft
N/A	Not Applicable
NIST	National Institute of Standards and Technology
OCSP	Online Certificate Status Protocol
OID	Object Identifier
OE	Operating Environment
OPM	On-Demand Privilege Manager
OS	Operating System
OSP	Organizational Security Policy
PAS	Privileged Access Security
PCRE	Perl Compatible Regular Expressions
PII	Personally Identifiable Information
PP	Protection Profile
PSM	Privileged Session Manager

Acronym	Definition
PSMP	Privileged Session Manager Proxy
PUB	Publication
PVWA	Private Vault Web Access
R2	Release 2
RA	Registration Authority
RBG	Random Bit Generator
RDP	Remote Desktop Protocol
RDS	Remote Desktop Services
RFC	Request for Comments
RHEL	Red Hat Enterprise Linux
RSA	Rivest, Shamir, Adelman
SAN	Subject Alternative Name
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm
SNMP	Simple Network Management Protocol
SP	Service Pack
SP	Service Publication
SPD	Security Policy Database
SQL	Structured Query Language
SRP	Secure Remote Protocol
SSL	Secure Sockets Layer
ST	Security Target
SW	Software
S/MIME	Secure/Multipurpose Internet Mail Extensions
TCP	Transport Control Protocol
TD	Technical Decision
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSFI	TOE Security Function Interface
TSS	TOE Summary Specification
URI	Uniform Resource Identifier

10.2 Terms

Table 18 defines the terms used throughout this document.

Table 18 – Terms

Name	Definition
Administrator/User	Human or IT entity interacting with the TOE from outside of the TOE boundary.
Assurance Activities	Actions that the evaluator will take in to determine compliance of a particular TOE with the SFRs
Common Criteria	Common Criteria for Information Technology Security Evaluation.
Common Evaluation Methodology	Common Evaluation Methodology for Information Technology Security Evaluation.
Protection Profile	An implementation-independent set of security requirements for a category of products.
Security Target	A set of implementation-dependent security requirements for a specific product.
Target of Evaluation	The product under evaluation. In this case, application software and its supporting documentation.
TOE Security Functionality	The security functionality of the product under evaluation.
TOE Summary Specification	A description of how a TOE satisfies the SFRs in a ST.
Security Functional Requirement	A requirement for security enforcement by the TOE.
Security Assurance Requirement	A requirement to assure the security of the TOE.

Appendix A: APIs

Table 19 lists all the APIs used by the TOE.

Table 19 – APIs Used by the TOE

Platform APIs used by the TOE		
AddAccessAllowedAce	GetModuleFileNameA	PathRemoveExtensionA
AddAccessDeniedAce	GetModuleFileNameW	PathRemoveFileSpecA
CallNamedPipeA	GetModuleHandleA	PathRemoveFileSpecW
CharLowerA	GetModuleHandleW	PathRenameExtensionA
CharUpperA	GetNumberFormatW	PathStripToRootA
CloseEventLog	GetNumberOfEventLogRecords	PathStripToRootW
CloseHandle	GetOEMCP	PostQueuedCompletionStatus
CloseServiceHandle	GetOldestEventLogRecord	QueryPerformanceCounter
CoCreateInstance	GetOverlappedResult	QueryPerformanceFrequency
CoInitializeEx	GetProcAddress	QueryServiceStatus
CompareFileTime	GetProcessHeap	RaiseException
CompareStringA	GetProcessTimes	ReadEventLogA
CompareStringW	GetQueuedCompletionStatus	ReadFile
ConnectNamedPipe	GetStartupInfoA	ReadProcessMemory
ControlService	GetStdHandle	RegCloseKey
CopyFileA	GetStringTypeA	RegEnumKeyExA
CopyFileW	GetStringTypeW	RegisterEventSourceA
CreateDirectoryA	GetSystemInfo	RegisterServiceCtrlHandlerA
CreateDirectoryW	GetSystemTime	RegisterServiceCtrlHandlerExA
CreateEventA	GetSystemTimeAsFileTime	RegOpenKeyA
CreateFileA	GetTempPathA	RegOpenKeyExA
CreateFileMappingA	GetThreadContext	RegQueryValueExA
CreateFileW	GetThreadLocale	ReleaseMutex
CreateIoCompletionPort	GetTickCount	ReleaseSemaphore
CreateMutexA	GetTimeFormatA	RemoveDirectoryA
CreateNamedPipeA	GetTimeFormatW	RemoveDirectoryW
CreateProcessA	GetTimeZoneInformation	ReportEventA
CreateSemaphoreA	GetTokenInformation	ResetEvent
CreateThread	GetUserDefaultLCID	ResumeThread
CreateWellKnownSid	GetUserNameA	RevertToSelf
DebugBreak	GetVersion	RpcStringFreeA
DecodePointer	GetVersionExA	RtlCaptureContext
DeleteCriticalSection	GetWindowsDirectoryA	RtlLookupFunctionEntry
DeleteFileA	GetWindowsDirectoryW	RtlPcToFileHeader
DeleteFileW	GlobalAlloc	RtlUnwindEx
DeregisterEventSource	GlobalFree	RtlVirtualUnwind
DeviceIoControl	GlobalLock	SetConsoleCtrlHandler
DisconnectNamedPipe	GlobalMemoryStatusEx	SetCurrentDirectoryA
DuplicateHandle	GlobalUnlock	SetEndOfFile
EncodePointer	HeapAlloc	SetEnvironmentVariableA
EnterCriticalSection	HeapCreate	SetEvent
EnumSystemLocalesA	HeapDestroy	SetFileAttributesA
ExitProcess	HeapFree	SetFileAttributesW
ExitThread	HeapQueryInformation	SetFilePointer
ExpandEnvironmentStringsA	HeapReAlloc	SetFilePointerEx
FatalAppExitA	HeapSetInformation	SetFileTime
FileTimeToLocalFileTime	HeapSize	SetHandleCount

Platform APIs used by the TOE		
FileTimeToSystemTime	HeapValidate	SetLastError
FindClose	InitializeAcl	SetNamedPipeHandleState
FindFirstFileA	InitializeCriticalSection	SetProcessShutdownParameters
FindFirstFileW	InitializeCriticalSectionAndSpinCount	SetSecurityDescriptorDacl
FindNextFileA	InitializeSecurityDescriptor	SetServiceStatus
FindNextFileW	IsBadReadPtr	SetStdHandle
FlsAlloc	IsBadStringPtrA	SetThreadPriority
FlsFree	IsBadWritePtr	SetThreadToken
FlsGetValue	IsDBCSLeadByteEx	SetUnhandledExceptionFilter
FlsSetValue	IsDebuggerPresent	Sleep
FlushFileBuffers	IsValidCodePage	StartServiceA
FormatMessageA	IsValidLocale	StartServiceCtrlDispatcherA
freeaddrinfo	LCMapStringA	StrCmpIW
FreeEnvironmentStringsA	LCMapStringW	StrCmpW
FreeEnvironmentStringsW	LeaveCriticalSection	SuspendThread
FreeLibrary	LoadLibraryA	SystemTimeToFileTime
GetACP	LoadLibraryExA	SystemTimeToTzSpecificLocalTime
GetActiveWindow	LoadLibraryW	TerminateProcess
getaddrinfo	LocalAlloc	TerminateThread
GetCommandLineA	LocalFree	TlsAlloc
GetConsoleCP	LockFile	TlsFree
GetConsoleMode	LockFileEx	TlsGetValue
GetConsoleOutputCP	LookupAccountSidA	TlsSetValue
GetCPInfo	lstrcatA	TryEnterCriticalSection
GetCurrencyFormatW	lstrcmpA	UnhandledExceptionFilter
GetCurrentDirectoryA	lstrcmpW	UnlockFile
GetCurrentDirectoryW	lstrcpyA	UnlockFileEx
GetCurrentProcess	lstrlenA	UnmapViewOfFile
GetCurrentProcessId	lstrlenW	UrlUnescapeW
GetCurrentThread	MapViewOfFile	UuidCreate
GetCurrentThreadId	MessageBoxA	UuidToStringA
GetDateFormatA	MoveFileA	VerQueryValueA
GetDateFormatW	MoveFileExA	VirtualAlloc
GetDiskFreeSpaceA	MoveFileExW	VirtualQuery
GetDiskFreeSpaceW	MoveFileW	WaitForMultipleObjects
GetDriveTypeA	MultiByteToWideChar	WaitForSingleObject
GetEnvironmentStrings	OpenEventA	WideCharToMultiByte
GetEnvironmentStringsW	OpenEventLogA	WriteConsoleA
GetEnvironmentVariableA	OpenFileMappingA	WriteConsoleW
GetExitCodeProcess	OpenMutexA	WriteFile
GetExitCodeThread	OpenProcessToken	WsCall
GetFileAttributesA	OpenSCManagerA	WsCloseServiceProxy
GetFileAttributesW	OpenServiceA	WsCreateError
GetFileSize	OpenThreadToken	WsCreateHeap
GetFileSizeEx	OutputDebugStringA	WsCreateServiceEndpointFromTemplate
GetFileType	OutputDebugStringW	WsCreateServiceProxyFromTemplate
GetFileVersionInfoA	PathFileExistsA	WsFreeError
GetFileVersionInfoSizeA	PathFileExistsW	WsFreeHeap
GetFullPathNameA	PathFindExtensionA	WsFreeServiceProxy
GetFullPathNameW	PathFindFileNameA	WsGetErrorString
GetLastError	PathIsDirectoryA	WsOpenServiceProxy
GetLocaleInfoA	PathIsDirectoryW	wsprintfA
GetLocaleInfoW	PathMatchSpecA	wvsprintfA
GetLogicalDriveStringsA		

Prepared by:
Corsec Security, Inc.

13921 Park Center Road, Suite 460
Herndon, VA 20171
United States of America

Phone: +1 703 267 6050
Email: info@corsec.com
http://www.corsec.com