CyberArk Software Ltd.

Privileged Access Security – Linux Components

Including Privileged Session Manager SSH Proxy (PSMP) v10.4 and On-Demand Privileges Manager (OPM) 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 – Linux Components, including Privileged Session Manager SSH[1] Proxy (PSMP) v10.4 and On-Demand Privileges Manager (OPM) v10.4, and will hereafter be referred to as the TOE throughout this document. The TOE is a software-based solution that runs on Linux and is a 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. PSMP enables organizations to secure, control, and monitor privileged access to network devices. OPM enables organizations to secure, control, and monitor privileged access to UNIX commands by allowing end users to perform super-user tasks with their own personal account without the need to know super-user credentials.

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.

[1] SSH – Secure Shell

Table 1 – ST and TOE References

ST Title	*CyberArk Software Ltd. Privileged Access Security – Linux Components including Privileged Session Manager SSH Proxy (PSMP) v10.4 and On-Demand Privileges Manager (OPM) v10.4 Security Target*
ST Version	Version 0.17
ST Author	Corsec Security, Inc.
ST Publication Date	September 27, 2019
TOE Reference	CyberArk Privileged Access Security – Linux Components including PSMP v10.4.1-3.x86_64 and OPM v10.04.01.2.x86_64

1.3 Product Overview

The product is the CyberArk's PAS Solution software suite, which enables organizations to secure, provision, control, and monitor all activities associated with the privileged identities used in enterprise systems.[2] 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 components PSMP and OPM provide the functionality to establish SSH connection to remote devices, and to manage and control access to privileged commands. 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.

The other PAS applications provide the functionality to securely store and control access to the privileged accounts, establish secure RDP connections to remote targets, administer and configure PAS, and enforce password policy. The PAS applications below interact with the TOE to provide the complete functionality of the PAS software suite; they are not covered by the evaluation.

EPV is the core component of the PAS software suite. EPV manages the secure storage and access to the privileged account files, and to the administrator and session activity files. The privileged account files are used to connect to target machines.

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[3] 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.

[2] 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 PSMP and OPM.

[3] RDP – Remote Desktop Protocol

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.

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 – Linux Components, which contains the PSMP and OPM software and CyberArk Version Check tool, all of which run on a Linux operating system (OS). PSMP acts as a proxy for SSH-enabled devices by controlling access to privileged sessions and initiating SSH connections to remote devices on behalf of the user without the need to disclose SSH credentials. Unique to the PSMP are the Single Sign-On capabilities, which allow users to connect to target devices without being exposed to the privileged connection credentials. PSMP is able to record text-based sessions that are stored in the Enterprise Password Vault (EPV) in the environment. OPM is a privilege manager that enables organizations to secure, control, and monitor access to privileged Linux commands. Users can perform super-user tasks with their own personal account, all while maintaining the least-privilege concept.

In its evaluated configuration, the TOE is a part of CyberArk's PAS Solution. PSMP and OPM are installed on a single instance of Red Hat Enterprise Linux (RHEL) 7.4. RHEL contains the required OpenSSL and OpenSSH Server packages that are required to secure communications with clients. PSMP contains the required OpenSSH Client package that is installed on the RHEL machine for communicating with remote targets. An instance of EPV is part of the operating environment (OE) and is installed on a standalone Windows 2012 R2[4] server for access control usage by both PSMP and OPM. 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. All components of the TOE and OE are installed on the same isolated network as other management devices. The TOE will need access to the target devices on this network when PSMP connects to them. This isolated network is separate from the production network and does not have access to the internet to ensure that attackers cannot access the TOE or OE from the internet.

TOE Administrators are referred to as administrators in this document. Administrators are responsible to managing the TOE as described in section 8.1.4 below. TOE Users are referred to as users in this document. Users are entities that access the PSMP Client TSFI to connect to a target device or the OPM Client TSFI to run an elevated command. Users will supply their CyberArk Vault account when using these interfaces. Both the administrators and users are part of the personnel that are responsible for maintaining the isolated network where the TOE and environment are installed.

The sections 1.4.1, 1.4.2, 1.4.3, and 1.5 respectively describe the software components of PSMP, OPM, CyberArk Version Check tool, and the components of the TOE environment in detail.

[4] R2 – Release Two

1.4.1 PSMP

The PSMP component allows users to obtain privileged account information through its PSMP Client TSFI[5] and then uses the TOE's OpenSSH Client to log the user onto a target device over a secured SSH connection.

A user connects to PSMP by providing a target device, target user, vault user, and vault password, which are then relayed to the EPV for verification. Once the user is verified, PSMP retrieves the target user's credentials to connect the user to the target device. While a user is connected to a target device and is performing activities in the privileged session, PSMP actively records all of the activities and uploads them to EPV. Communications between PSMP and EPV are conducted over TLS[6] v1.2. PSMP is compiled with the OpenSSL FIPS[7] Object Module[8] v2.0.13 that includes the following CyberArk libraries for its cryptographic functionality: CyberArk PAS Cryptographic Library for Linux v1.0, CyberArk PAS TLS Library for Linux v1.0, and CyberArk PAS Linux SSH Client Cryptographic Module.

1.4.2 OPM

The OPM component is accessible via the OPM Client TSFI, and it 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 command permissions (i.e., Access Control List [ACL]) that define command and access permissions. Each ACL determines the commands that each user or group can issue for a given account or for an entire platform. Command permissions and policies are created and managed in EPV. Once authorized, users can issue privileged commands from their Linux machines according to the ACL defined in EPV and elevate their standard login session to a privileged session that runs privileged commands.

Users who require privileged account permissions to execute a privileged task will invoke the OPM Client TSFI by using the pimsu command, which refers to the local Privileged Identity Manager (PIM) Provider to start the privileged session. The PIM Provider maintains a local cache that contains the access control details that permit each user to invoke the specific privileged commands that they requested and no other commands. The PIM Provider uses a unique PIM Provider account to access EPV when it is retrieving the access control details or storing the session recordings and audit information. The PIM Provider constantly refreshes its cache from EPV, so that it always contains accurate information. The PIM Provider maintains audit logs and session recordings, so that there is complete accountability for each privileged command request by every user, and monitors logs that register PIM Provider activity and status. The PIM Provider maintains a local cache, which is a datafile with records on OPM's host. This datafile contains the access control details that permit each user to invoke the privileged command requested. The local cache eliminates the need to access EPV for every privileged command invocation and provides high availability, regardless of EPV or network availability. This datafile is only accessible by the PIM Provider. No direct network access to the datafile is allowed.

[5] TSFI – TOE Security Function Interface
[6] TLS – Transport Layer Security
[7] FIPS – Federal Information Processing Standard
[8] 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.

OPM communicates with EPV to retrieve and update the ACLs for the privileged accounts on the Linux system. Communications between OPM and EPV are conducted over TLS v1.2. OPM is compiled with the OpenSSL FIPS Object Module v2.0.13 that includes the following CyberArk libraries for its cryptographic functionality: CyberArk PAS Cryptographic Library for Linux v1.0 and CyberArk PAS TLS Library for Linux v1.0.

1.4.3 CyberArk Version Check Tool

The CyberArk Version Check tool is a script used to query the current versions of the installed PAS components and checks if an update is available for the found components. The tool relies on the EPV server to store the file it will use to check for the latest version.

1.5 TOE Environment

It is assumed that there will be no untrusted users or software on the TOE server component. Access to the server's OS must be limited to authorized users and secured with an authentication method. In addition, the TOE server component is 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, PSMP and OPM are installed on the RHEL 7.4 OS. All TOE components require access to the same internal network in which the EPV server is installed in order to interact with it. Note that the platform that the TOE is installed on will not be allowed to have access to the internet and is intended for only intranet use. PSMP will also be used to connect to remote targets using SSH to allow TOE users to securely interact with devices on the internal network. This requires the installation of OpenSSH Client in the TOE and RHEL's OpenSSH Server in the OE. OpenSSH Client is automatically installed when installing PSMP and RHEL's OpenSSH Server is installed following its FIPS Security Policies. RHEL's OpenSSH Server also require RHEL's OpenSSL package to be installed following its FIPS Security Policies. The CyberArk Version Check tool is also downloaded to the host from the CyberArk support vault and relies on the EPV server for storing the file that contains the latest version information.

The TOE communicates with the EPV server in the OE, which is on the same isolated network as the TOE. EPV is installed on a hardened[9] version of Microsoft Windows Server 2012 R2. Access to and from the EPV server is available to only components of the PAS Solution via a secure channel over TLS v1.2. EPV is used by the TOE for authenticating users, reading access control lists, downloading configuration settings, and storing recorded data.

A user can obtain access to PSMP by accessing a Linux Shell system or through a third-party Windows Shell application, such as PuTTY. When the user sends the PSMP command to connect to a target machine through a shell program, they will connect to the OE's OpenSSH Server application. PSMP will record the communications from the OpenSSH Server for auditing and authorization. Audit data and authorization requests are sent to the EPV server by PSMP. If the user provided the correct credentials, the EPV server will return the information PSMP needs to authenticate with the remote target. PSMP will use the authentication information from EPV and the target information from the user to setup a session with the target device using the TOE's OpenSSH Client application. All information returned from the TOE's OpenSSH Client connection is recorded and then routed back to the user through the OE's OpenSSH Server.

[9] Protected by a firewall that only allows remote communications from CyberArk applications via a secure channel.

When the user needs to perform a privileged task, they invoke OPM's pimsu command to access the OPM Client TSFI from their terminal. The OPM Client TSFI checks whether the user has permission to access the account required to perform this task or run this session. If the command permissions in the EPV server gives the user the appropriate permissions to run this task, OPM will automatically open a privileged session on a pseudo terminal without exposing the root password to the user. OPM then runs the privileged command and redirects the input/output of the command to the user's terminal where they can follow the process of the command. OPM records the entire privileged session. When the session has been completed, the recording is uploaded into the EPV server where it can be accessed by authorized users.

Table 2 lists the required non-TOE components for the TOE and describes the requirements for the components found in the TOE environment.

Table 2 – Environmental Components

Component	Requirements
RHEL Server	Operating System: • Red Hat Enterprise Linux 7.4 Hardware: • Intel i7-6700 or Intel Xeon E5 family processor Required Platform Applications: • 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
EPV Server	Operating System: • Microsoft Windows Server 2012 R2 • CyberArk EPV v10.4 Hardware: • Intel i7-6700 or Intel Xeon E5 family processor Required Platform Applications: • .NET Framework 4.5.2
SSH Client Workstation	Any SSH client, such as plink, PuTTY, SecureCrt, etc., that complies with RFCs[10] 4251, 4252, 4253, and 4254.
Remote Target	Any SSH server that complies with RFCs 4251, 4252, 4253, and 4254.

1.6 TOE Description

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

[10] RFC – Request for Comments

1.6.1 Physical Scope

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

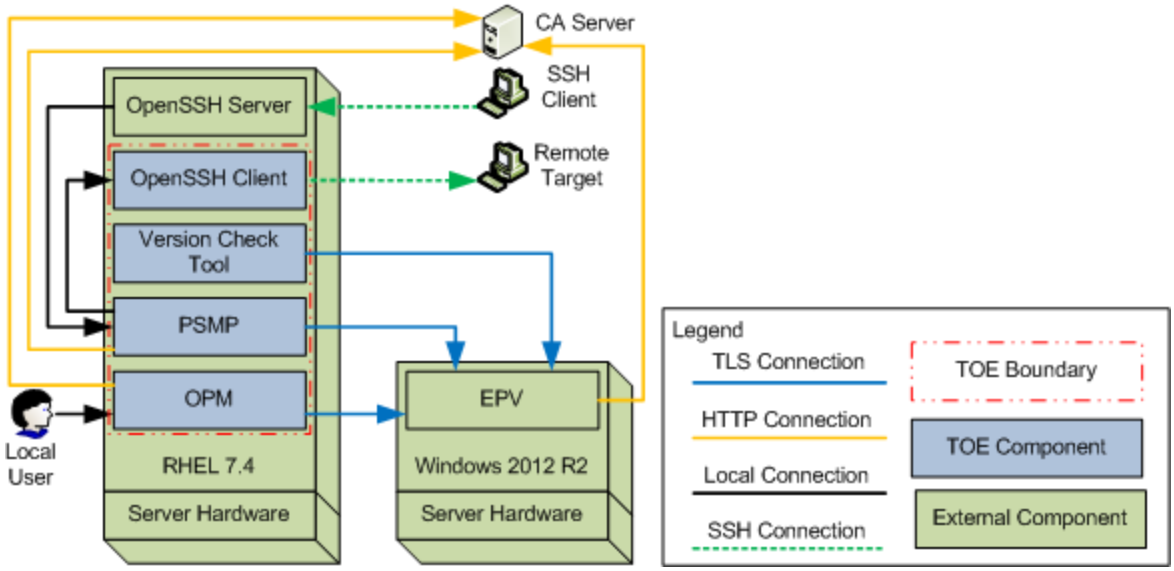

Figure 1 – TOE Boundary

The TOE Boundary includes all the CyberArk developed parts of PSMP, OPM, and the and Version Check tool.

1.6.1.1 TOE Software

The TOE is a software-only TOE and is comprised of the PSMP, OPM, and Version Check tool software. For the evaluated configuration, the following TOE software must be installed on the RHEL machine in the environment:

- CyberArk PSMP v10.4.1-3.x86_64
  - Including PSMP's OpenSSH Client
- CyberArk OPM v10.04.01.2.x86_64
- 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; PASINS-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	

Document Name	Description
CyberArk Software Ltd.; Privileged Access Security – Linux Components; Guidance Documentation Supplement; Document 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.

1.6.2.1 Cryptographic Support

The TOE uses CAVP-validated cryptographic algorithm provided by its own CyberArk cryptographic libraries. The library is used to support the establishment of trusted channels to protect data in transit. In the evaluated configuration, the TOE's cryptographic library is used by the OpenSSH Client to remote targets and the TLS client connection to the EPV server. The TOE provides the cryptographic functionality listed in Table 15 below.

1.6.2.2 User Data Protection

The TOE stores sensitive information in the form of encrypted passwords in non-volatile memory. The TOE will limit its access to only network connectivity when accessing the platform's hardware resources. The network connection is used for communications between the TOE to the EPV server, the TOE to the target devices, and the user to the TOE. The TOE will also access the EPV server's sensitive information repository (safes) when it needs to authenticate users or request root credentials.

1.6.2.3 Identification and Authentication

To validate the EPV server's certificate during the TLS handshake, the TOE implements functionality to validate X.509 certificates. The TOE uses a CRL[11] to check certificate revocation status and will not establish a connection to the EPV server when the CRL is unavailable.

1.6.2.4 Security Management

The TOE is configured with default file permissions already in place and does not provide default credentials for user authentication. The TOE relies on the platform for storing and setting configuration options within its config files. Administrators are able to configure the basic PSMP or OPM configuration parameters and restart the related service.

1.6.2.5 Privacy

The TOE does not store or transmit any personally identifiable information (PII).

1.6.2.6 Protection of the TOE Security Functionality (TSF)

The TOE protects against exploitation by implementing address space layout randomization (ASLR) except for vsyscall and not allocating memory with both writing and execution. The TOE is also compatible with SELinux and is compiled with stack-based buffer overflow protection. It also stores user-modifiable files to directories that do not contain executable files.

[11] CRL – Certificate Revocation List

The TOE uses standard platform APIs[12] and includes only the third-party libraries it needs to perform its functionality.

The TOE version can be checked using commands provided by the platform. Checking for updates to the TOE is reliant on the platform's functionally. Any update downloaded for the TOE must be installed using the platform's package manager. An administrator will install a public key from CyberArk that is used by the package manager to verify the integrity of any updates to the TOE.

1.6.2.7 Trusted Path/Channels

The TOE provides a trusted channel between itself and target devices over SSH using OpenSSH Client. The SSH software used by the TOE follows the Extended Package for Secure Shell. A trusted TLS channel is used between itself and the EPV server.

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:

- Using PSMP for anything other than SCP[13] commands
- Functionality provided by EPV.
- Functionality provided by PVWA.
- Functionality provided by CPM.
- Functionality provided by PSM.

1.6.4 Scope of Evaluation

The evaluation is limited in scope to the secure features described in the *Protection Profile for Application Software v1.2*; April 22, 2016 (AS PP), *Extended Package for Secure Shell (SSH) v1.0*; 2016-02-19 (SSH EP), and detailed in Section 1.6. The TOE is conformant to the AS PP and SSH EP and no interpretations apply to the claims made in this ST.

[12] API – Application Programming Interface
[13] SCP – Secure Copy

2. Conformance Claims

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

Table 4 – 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 claim to the *Extended Package for Secure Shell (SSH) v1.0; 2016-02-19* conformant.
PP Identification	Exact Conformance[14] to the *Protection Profile for Application Software v1.2;* April 22, 2016 and the *Extended Package for Secure Shell (SSH) v1.0*; 2016-02-19.
TD Conformance	Conformance to the following AS PP TDs is claimed: • 0435 – Alternative to SELinux for FPT_AEX_EXT.1.3 • 0392 – FCS_TLSC_EXT.1.2 Wildcard Checking • 0389 – Handling of SSH EP claim for platform • 0382 – Configuration Storage Options for Apps • 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 • 0304 – Update to FCS_TLSC_EXT.1.2 • 0300 – Sensitive Data in FDP_DAR_EXT.1 • 0293 – Update to FCS_CKM.1(1) • 0268 – FMT_MEC_EXT.1 Clarification • 0244 – FCS_TLSC_EXT – TLS Client Curves Allowed • 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 • 0174 – Optional Ciphersuites for TLS • 0163 – Update to FCS_TLSC_EXT.1.1 Test 5.4 and FCS_TLSS_EXT.1.1 Test • 0119 – FCS_STO_EXT.1.1 in PP_APP_v1.2 • 0107 – FCS_CKM – ANSI[15] X9.31-1998, Section 4.1.for Cryptographic Key Generation Conformance to the following SSH EP TDs is claimed: • 0420 – Conflict in FCS_SSHC_EXT.1.1 and FCS_SSHS_EXT.1.1 • 0331 – SSH Rekey Testing • 0240 – FCS_COP.1.1(1) Platform provided crypto for encryption/decryption

[14] 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.

[15] ANSI – American National Standards Institute

3. Security Problem Definition

This section describes the security aspects of the environment in which the TOE will be used and the manner in which 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 OSPs 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	The threat T.NETWORK_ATTACK is countered by O.PROTECTED_COMMS as this provides for integrity of transmitted data.
	O.INTEGRITY	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.
	O.MANAGEMENT	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	The threat T.NETWORK_EAVESDROP is countered by O.PROTECTED_COMMS as this provides for confidentiality of transmitted data.
	O.QUALITY	The objective O.QUALITY ensures use of mechanisms that provide protection against network-based attack.
	O.MANAGEMENT	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 not repeated in this ST, but they are taken directly from the AS PP and SSH EP.

5.2 Extended TOE Security Assurance Components

Table 10 identifies the extended SARs claimed for the TOE. The 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[16] 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 for the operational environment (ASE_OBJ.1)
	Stated 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[17] 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)

[16] ITSEF – Information Technology Security Evaluation Facility
[17] CM – Configuration Management

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.

The Assurance Activities defined in AS PP and SSH EP 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_COP.1(5)	Cryptographic Operation – Encryption/Decryption (Refined) (SSH)	✓			✓
FCS_RBG_EXT.1	Random Bit Generation Services	✓			
FCS_SSH_EXT.1	SSH Protocol	✓			
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	✓	✓		

Name	Description	S	A	R	I
FMT_CFG_EXT.1	Secure by Default Configuration				
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_SSHC_EXT.1	SSH Protocol – Client	✓	✓		
FCS_TLSC_EXT.1	TLS Client Protocol	✓			
FCS_TLSC_EXT.4	TLS Client 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[18] schemes]** using [**"NIST[19] curves" P-256, P-384 and [no other curves]**] that meet the following: [**FIPS PUB[20] 186-4, "Digital Signature Standard (DSS)", Appendix B.4**]

].

[18] ECC – Elliptic Curve Cryptography
[19] NIST – National Institute of Standards and Technology
[20] PUB – Publication

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"**]

].

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[21]-CBC[22] (as defined in NIST SP[23] 800-38A) mode;
- and [AES-GCM[24] (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[25]-256, SHA-384, SHA-512] and message digest sizes [256, 384, 512] bits that meet the following: FIPS Pub 180-4.

FCS_COP.1(3) Cryptographic Operation – Signing

FCS_COP.1.1(3)

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

- [**RSA[26] schemes**] using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 4,

[21] AES – Advanced Encryption Standard
[22] CBC – Cipher Block Chaining
[23] SP – Special Publication
[24] GCM – Galois Counter Mode
[25] SHA – Secure Hash Algorithm
[26] RSA – Rivest, Shamir, Adleman

- [**ECDSA[27] schemes**] using "NIST curves" P-256, P-384 and [no other curves] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5

].

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[28]-SHA-256
- and [SHA-384, SHA-512]

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

FCS_COP.1(5) Cryptographic Operation – Encryption/Decryption (Refined)

FCS_COP.1.1(5)

The SSH software shall [perform] encryption/decryption services for data in accordance with a specified cryptographic algorithm AES-CTR[29] (as defined in NIST SP 800-38A) mode and cryptographic key sizes [128-bit, 256-bit].

FCS_RBG_EXT.1 Random Bit Generation Services

FCS_RBG_EXT.1.1

The application shall [implement DRBG[30] 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[31] 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.

[27] ECDSA – Elliptic Curve Digital Signature Algorithm
[28] HMAC – Hash-based Message Authentication Code
[29] CTR – Counter Mode
[30] DRBG – Deterministic Random Bit Generator
[31] RBG – Random Bit Generation

FCS_SSH_EXT.1 SSH Protocol

FCS_SSH_EXT.1.1

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

FCS_SSHC_EXT.1 SSH Protocol – Client

FCS_SSHC_EXT.1.1

The SSH client shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, and [none].

FCS_SSHC_EXT.1.2

The SSH client shall ensure that, as described in RFC 4253, packets greater than [*35,000*] bytes in an SSH transport connection are dropped.

FCS_SSHC_EXT.1.3

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

FCS_SSHC_EXT.1.4

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

FCS_SSHC_EXT.1.5

The SSH client shall ensure that the SSH transport implementation uses [hmac-sha2-256, hmac-sha2-512] and [no other MAC[32] algorithms] as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHC_EXT.1.6

The SSH client shall ensure that [ecdh-sha2-nistp256] and [ecdh-sha2-nistp384] are the only allowed key exchange methods used for the SSH protocol.

FCS_SSHC_EXT.1.7

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

FCS_SSHC_EXT.1.8

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

[32] MAC – Message Authentication Code

FCS_STO_EXT.1 Storage of Credentials

FCS_STO_EXT.1.1

The application shall [implement functionality to securely store [*passwords to credentials for the psmpappuser, psmpgwuser, and opmuser accounts*]] to non-volatile memory.

FCS_TLSC_EXT.1 TLS Client Protocol

FCS_TLSC_EXT.1.1

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

- TLS_ECDHE[33]_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
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

].

FCS_TLSC_EXT.1.2

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

FCS_TLSC_EXT.1.3

The application shall establish a trusted channel only if the peer certificate is valid.

FCS_TLSC_EXT.4 TLS Client Protocol

FCS_TLSC_EXT.4.1

The application shall present the supported Elliptic Curves Extension in the Client Hello with the following NIST curves: [secp256r1, secp384r1].

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].

[33] ECDHE – Elliptic Curve Diffie Hellman Ephemeral

FDP_DEC_EXT.1.2

The application shall restrict its access to [[*SELinux logs*]].

FDP_NET.EXT.1 Network Communications

FDP_NET_EXT.1.1

The application shall restrict network communication to [user-initiated communication for [*SSH connections to PSMP from the OpenSSH Server component*], [*application-initiated SSH connections to targets, TLS connections to the EPV server to download ACLs and authentication users, 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[34] 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 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 [35] 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[36] 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 [37] 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.

[34] CA – Certificate Authority
[35] OID – Object Identifier
[36] MIME – Multipurpose Internet Mail Extensions
[37] OCSP – Online Certificate Status Protocol

- Server certificates presented for EST[38] shall have the CMC[39] 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].

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

].

[38] EST – Enrollment over Secure Transport

[39] CMC – Certificate Management over Cryptographic Message Syntax

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 [*vsyscall*].

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 proprietary libraries listed in Table 13 and no third-party libraries*]

Table 13 – Third-Party Libraries

Components	Libraries	Vender
PSMP	libcauserprovisioning.so	CyberArk Software Ltd.
OPM	Libopmpreloader.so	CyberArk Software Ltd.
	Libcpasswordsdk.so	CyberArk Software Ltd.
	Libcpasswordsdk32.so	CyberArk Software Ltd.

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, SSH as conforming to the Extended Package for Secure Shell]

] between itself and another trusted IT[40] product.

[40] IT – Information Technology

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[41]	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_COP.1(5)	Cryptographic Operation – Encryption/Decryption (Refined)
	FCS_RBG_EXT.1	Random Bit Generation Services
	FCS_RBG_EXT.2	Random Bit Generation from Application
	FCS_SSH_EXT.1	SSH Protocol
	FCS_SSHC_EXT.1	SSH Protocol – Client
	FCS_STO_EXT.1	Storage of Credentials
	FCS_TLSC_EXT.1	TLS Client Protocol
	FCS_TLSC_EXT.4	TLS Client 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

[41] ID – Identification

TOE Security Function	SFR ID[41]	Description
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
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 v2.0.13 that includes the following CyberArk libraries to provide the required algorithms for all cryptographic operations used within the TOE: CyberArk PAS Cryptographic Library for Linux v1.0, CyberArk PAS TLS Library for Linux v1.0, and CyberArk PAS Linux SSH Client Cryptographic Module. Table 15 lists required information about the TOE's and OE's usage of cryptography.

Table 15 – Cryptographic Algorithm and Key Sizes for PSMP and OPM

Cryptographic Operation	Algorithm	Key Sizes / Curves	Usage	Certificate
Encryption/Decryption	AES – CBC and GCM	128, 256	TLS	CAVP 5486 and C1087
	AES – CBC and CTR	128, 256	SSH (client)	
	AES – CBC	256	Inside *.cred files	
Signature Generation Signature Verification	RSA	2048, 3072	TLS	CAVP 2947 and C1087
Key Pair Generation Public Key Verification Signature Generation Signature Verification	ECDSA – P256 and P384	256, 384	TLS and SSH (client)	CAVP 1472 and C1087
Key Exchange/Establishment	ECDHE	256, 384	TLS	CAVP 1942 and C1087
	ECDH	256, 384	SSH (client)	
Message Digest / Hashing	SHA-256, SHA-384	256, 384	TLS	CAVP 4403 and C1087
	SHA-256, SHA-512	256, 512	SSH (client)	
Message Authentication	HMAC-SHA-256, SHA-384	256, 384	TLS	CAVP 3644 and C1087
	HMAC-SHA-256, SHA-512	256, 512	SSH (client)	
Random Number Generation	CTR DRBG (with AES)	N/A[42]	TOE DRBG	CAVP 2161 and C1087

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

Table 15 above lists all the key sizes used for ECC asymmetric key generation schemes and the usage of each key. Table 15 also lists the key establishment and key exchange schemes used by the TOE. The TOE uses key generation and establishment/exchange with the TLS and SSH protocols. The use of asymmetric encryption is needed for the TLS and SSH protocols used by the TOE. The key generation methods follow the requirements within FIPS PUB

[42] N/A – Not Applicable

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)/FCS_COP.1(5)
Table 15 above lists all the key sizes used for AES encryption and decryption within the TOE. Encryption and decryption operations are limited to being used in TLS, SSH, and protecting passwords in credential files. The TOE uses AES-CBC and AES-GCM in its TLS connections. AES is included in the TOE's cryptographic libraries that are statically linked to their components. The cryptographic algorithm follows NIST SP 800-38A (CBC and CTR) and NIST SP 800-38D (GCM). The cryptographic key sizes are 128-bit and 256-bit for all modes.

FCS_COP.1(2)
Table 15 above lists all the key sizes used for SHA hashing and message digests within the TOE. Usages of SHA is limited to TLS and SSH connections. The TOE's implementations of SHA follow the requirements within FIPS Pub 180-4.

FCS_COP.1(3)
Table 15 above lists all the key sizes used for signature generation and verification within the TOE. Signature generation is used in TLS and SSH connections. Signature verification is used in TLS and SSH connections. The TOE's implementations of signature generation and verification follow the requirements within FIPS PUB 186-4.

FCS_COP.1(4)
Table 15 above lists all the key sizes used for HMAC message authentication within the TOE. Usages of HMAC is limited to TLS and SSH connections. The TOE's implementations of HMAC follows the requirements within FIPS Pub 180-4.

FCS_RBG_EXT.1/FCS_RBG_EXT.2
The TOE's CTR_DRBG functionality is implemented within its statically linked cryptographic library. This implementation conforms to the NIST Special Publication 800-90A requirements. The TOE's implementation of CTR_DRBG is AES-256. The DRBG 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.

When the DRBG requires entropy bits, it uses the RAND_seed() function to fetch entropic bits from the blocking /dev/random device driver. The number of bits requested is compared to the entropy counter value. If there is less entropy in the Output Entropy Pool than requested by RAND_seed, then the DRBG attempts to transfer bits from the Input Entropy Pool to the Output Entropy Pool. If the Input Entropy Pool has sufficient entropy, then entropy is extracted from the Input Entropy Pool and input to the Output Entropy Pool. If there is not enough entropy available in the Input Entropy Pool to provide to the Output Entropy Pool, then the DRBG waits until sufficient entropy is available.

The SP 800-90A CTR_DRBG requires a minimum entropy input length of 256 bits for instantiation and reseeding. The 256 bits of entropy are requested during the DRBG's instantiation using the get_entropy function. Entropy is only extracted from /dev/random during the instantiation and reseed operations for the DRBG. Subsequent random number requests continue to use the output of the properly seeded and instantiated DRBG. The DRBG does require a reseed at an interval, which, as specified in SP 800-90A, must be <= 2^32 or 2^48 (for higher strength algorithms). Therefore, even at the lower value of 2^32, 4.2 billion DRBG generate operations may occur

before a reseed is required. The amount of time between the required reseed operations provides more than ample time for enough entropy to be gathered for the next /dev/random call.

FCS_SSH_EXT.1
The TOE uses the OpenSSH Client in the TOE boundary to implement the SSH protocol according to RFCs 4251, 4252, 4253, 4254, 5656, and 6668.

FCS_SSHC_EXT.1
The OpenSSH Client component implements public key-based authentication as described in RFC 4252. Note that password-based authentication is not enabled for the OpenSSH Client component. The OpenSSH Client uses the TOE's cryptographic library for ECDSA-SHA2-NISTP256 and ECDSA-SHA2-NISTP384 as its public key algorithms and rejects all other public key algorithms. The TOE's encryption algorithms used by the OpenSSH Client for SSH transport include AES128-CTR, AES256-CTR, AES128-CBC, and AES256-CBC. All other encryption algorithms are rejected. No other optional characteristics are used for the encryption and public key algorithms. The OpenSSH Client uses the TOE's HMAC-SHA2-256, and HMAC-SHA2-512 as its data integrity MAC algorithms used in SSH transport. All other MAC algorithms are rejected. The key exchange algorithms used by the OpenSSH Client include ECDH-SHA2-NISTP256 and ECDH-SHA2-NISTP384. No other key exchange methods can be used for the SSH protocol. The OpenSSH Client underwent component validation testing and was assigned the CAVP certificate 1943 and C1089.

The OpenSSH Client component ensures that packets greater than 35,000 bytes are dropped from the SSH transport connection. The max uncompressed payload length is 32,768 bytes for a max packet size of 35,000 bytes (including "packet_length", "padding_length", "payload", "random padding", and "mac"). The OpenSSH Client also ensures that the SSH connection is rekeyed after 1 gigabyte of data has been transmitted or 1 hour of connection time has passed, whichever happens first.

The OpenSSH Client component ensures that the identity of the SSH server is authenticated using a local database that associates each host name with its corresponding public key.

FCS_STO_EXT.1
The TOE securely stores passwords in non-volatile memory for the following accounts:

- PSMPappuser – This is the EPV account that will run the PSMP application. It is located in the "/etc/opt/CARKpsmp/Vault/" folder in the psmpappuser.cred file.
- PSMPgwuser – This is the EPV account that will run the PSMP gateway. It is located in the "/etc/opt/CARKpsmp/Vault/" folder in the psmpgwuser.cred file.
- OPMUser – This is the EPV account that will run the OPM application and is used to access the specified safe on the EPV server. It is located in the "/etc/opt/CARKaim/vault" folder in the opmuser.cred file.

Each password for the above accounts is encrypted using AES256-CBC and saved in the noted *.cred file.

FCS_TLSC_EXT.1
The TOE implements a TLS v1.2 client according to RFC 5246 using its CyberArk PAS TLS Library for Linux. This functionality is only used to communicate to the EPV server over TLS. Only the following cipher suites are allowed by the TOE:

- 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
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384

As part of establishing a TLS connection as a client, the TOE will verify that the presented identifier in the server certificate is a valid reference identifier according to RFC 6125. The reference identifier is established by the PSMP and OPM applications. The acceptable reference identifiers are a Common Name or IP[43] address for the Subject Name field. The Common Name field of the EPV server's certificate may contain its IP address because the EPV server is on a hardened machine that is not necessarily accessible via DNS[44]. The use of wildcards in the Subject Name is not supported. Certificate pinning is also not supported. The TOE will only establish a TLS connection if the peer certificate is valid. The TLS library underwent component validation testing and was assigned the CAVP certificate 1944 and C1091.

FCS_TLSC_EXT.4
The TOE implements TLS v1.2 with support for EC[45] algorithms. The TOE supports the secp256r1 and secp384r1 EC Extensions to protect its communications with the EPV server. The support of these curves is enabled by default and no additional configuration is required.

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_COP.1(5), FCS_RBG_EXT.1, FCS_RBG_EXT.2, FCS_SSH_EXT.1, FCS_SSHC_EXT.1, FCS_STO_EXT.1, FCS_TLSC_EXT.1, FCS_TLSC_EXT.4

8.1.2 User Data Protection

FDP_DAR_EXT.1
Sensitive data within the TOE is limited to the passwords for service accounts. The TOE limits the storing of sensitive data in non-volatile memory to only passwords for the psmpappuser, psmpgwuser, and opmuser service accounts. Each account's password is encrypted using AES256-CBC before it is saved. Other passwords that are used to authenticate with the EPV server are never stored by the TOE and transferred to and from the TOE over secure connections.

FDP_DEC_EXT.1 and FDP_NET_EXT.1
The TOE will limit its access to only network connectivity when accessing the platform's hardware resources. The TOE requires network access to the SSH client, EPV server, and target devices. The user will initiate the connection from their SSH client to the PSMP Client TSFI over port 22 when they want to SSH to a target machine, requiring the TOE to use the OpenSSH Server to use network resources. The TOE will connect to the EPV server and send the authentication information over TLS on port 443 for verification. If the authentication passes, PSMP then connects to the target device with the supplied credentials using OpenSSH Client over an SSH connection on port 22. OPM connects to the EPV server over TLS on port 443 when a user uses the OPM Client TSFI to execute a Linux command that requires elevated permissions. The TOE will use port 80 for HTTP connections to the CA server for

[43] IP – Internet Protocol
[44] DNS – Domain Name System
[45] EC – Elliptic Curve

certification revocation checks from each component. The TOE also connections to the EPV server periodically over TLS on port 443 to download ALCs and configuration information.

The TOE will limit its access to sensitive information repository, which includes the SELinux logs. The SELinux logs are accessed when the TOE needs to store related event data.

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 TOE's PSMP and OPM components are clients to the EPV server and each validates the EPV server's X.509v3 certificate during TLS authentication. The components ensure that the X.509v3 certificate adheres to RFC 5280 (certificate validation and certificate path validation) and that the certificate path terminates with a trusted CA certificate. The components treat 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. Each of the components validates the revocation status of the EPV'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 PSMP and OPM components each check the EPV certificate against the downloaded CRL and automatically reject 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 PSMP and OPM components each validate that the EPV's server certificate presented for TLS has the Server Authentication purpose in the extended key usage field.

The TOE does not accept S/MIME, OCSP 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. Both components of the TOE will read the location of their certificates from their vault.ini files using the ClientCertificate parameter.

The TOE validates X.509v3 certificates from the EPV server for TLS authentication. This functionality is enabled by default. The path to the CRL is read from the certificate's CRL Distribution Point (CDP) field. The TOE's implementation of TLS will automatically reject a certificate if it is found to be invalid according to the requirements in FIA_X509_EXT.1. A certificate with an unknown revocation status due to the inability to establish a connection to the CDP will be rejected. The connection from the TOE to the CDP is conducted over HTTP as per the RFC.

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

8.1.4 Security Management

FMT_CFG_EXT.1
No default user credentials are provided by the TOE. Once the TOE is installed, it is connected to the EPV server for all authentication needs. The files created during installation are set with default permissions. The Linux

permissions for “other” are set to “0” on all files except the ones used to launch the TOE, which have read and execute permissions, to protect the files from modifications made by unprivileged user.

FMT_MEC_EXT.1
The TOE does not write or set any configuration options using local configuration files. All configuration settings are stored in a safe on the EPV server and can be configured using the Password Vault Web Access interfaces. The TOE contains local configuration files that are created during installation, but the information is read-only and never written to by the TOE. Local configuration information related to the PSMP component is stored in “/etc/opt/CARKpsmp/conf/basic_psmpserver.conf” file. Local configuration information related to the OPM component is stored in “/etc/opt/CARKaim/conf/basic_opm.conf” file.

FMT_SMF.1
The TOE does not provide any methods of management through PSMP and OPM because these components read their configurations from the EPV server.

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 collect PII for administrators or users. Therefore, there is no case in which the TOE will transmit this data over the network.

TOE Security Functional Requirements Satisfied: FPR_ANO_EXT.1

8.1.6 Protection of the TSF

FPT_AEX_EXT.1
The TOE does not request memory mappings at explicit addresses except for the usage of vsyscall. Vsyscall is a legacy memory segment that was added as a way to execute specific system calls which do not need any real level of privilege to run, such as gettimeofday. It has a fixed address ffffffffff600000 and only has read and execute permissions. When the TOE is being complied, it uses the “-fPIC” flag to enable ASLR. The TOE does not allocate any memory region with both write and execute permissions. No just-in-time compilations are performed by the TOE. The TOE is also complied using the “-fstackprotector=all” flag to enable stack-based buffer overflow protection.

The TOE is compatible with SELinux and use SELinux profile files that are applied during installation.

The TOE does not write user-modifiable files to directories that contain executable files. User modifiable files are written to the “/etc/opt/CARKpsmp/”, “/etc/opt/CARKaim/”, “/var/opt/CARKpsmp/”, “/var/opt/CARKaim/”, and “/var/tmp/opm-install-logs/” folders. Executable files are stored in the “/opt/CARKpsmp/” and “/opt/CARKaim/”.

FPT_API_EXT.1
The only TOE uses supported platform APIs in order to function. The below list includes all the platform APIs used by the PSMP component. To read more about each system call used by PSMP, please refer to its reference in the Linux manual pages (also known as man pages).

- accept
- access
- arch_prctl
- bind
- brk
- chdir
- chmod
- chown
- clone
- close
- connect
- detached
- dup
- dup2
- dup3
- execve
- write
- exit_group
- fchown
- fcntl
- fstat
- futex
- getcwd
- getdents
- getegid
- geteuid
- getgid
- getpeername
- getpgrp
- getpid
- getppid
- getrlimit
- getsockname
- getsockopt
- getuid
- getxattr
- ioctl
- lgetxattr
- listen
- lseek
- lstat
- madvise
- mkdir
- mknod
- mmap
- mprotect
- munmap
- nanosleep
- open
- openat
- pipe
- pipe2
- poll
- ptrace
- read
- readlink
- recvfrom
- recvmsg
- rename
- restart_syscall
- rmdir
- rt_sigaction
- rt_sigprocmask
- rt_sigreturn
- select
- sendto
- setgroups
- setresgid
- setresuid
- set_robust_list
- setsid
- setsockopt
- set_tid_address
- shutdown
- socket
- stat
- statfs
- tgkill
- times
- umask
- uname
- unlink
- wait4
- with

The below list includes all the platform APIs used by the OPM component.

- accept
- access
- alarm
- arch_prctl
- bind
- brk
- chdir
- chmod
- chown
- clone
- close
- connect
- dup
- dup2
- dup3
- execve
- _exit
- exit_group
- faccessat
- fadvise64
- write
- fchdir
- fchmod
- fchown
- fcntl
- fstat
- fstatfs
- fsync
- futex
- getcwd
- getdents
- getegid
- geteuid
- getgid
- getgroups
- getpeername
- getpgid
- getpgrp
- getppid
- getrlimit
- getsid
- getsockname
- getsockopt
- gettid
- getuid
- getxattr
- ioctl
- kill
- lgetxattr
- listen
- lseek
- lstat
- madvise
- mkdir
- mmap
- mprotect
- munmap
- nanosleep
- newfstatat
- open
- openat
- pipe
- pipe2
- poll
- pread
- read
- readlink
- recvfrom
- recvmsg
- rename
- restart_syscall
- rmdir
- rt_sigaction
- rt_sigprocmask
- rt_sigreturn
- select
- sendmsg
- sendto
- setgroups
- setpgid
- setresgid
- setresuid
- set_robust_list
- setsid
- setsockopt
- set_tid_address
- setuid
- sigaltstack
- socket
- stat
- statfs
- symlink
- sysinfo
- tgkill
- times
- umask
- uname
- unlink
- unlinkat
- utimensat
- wait4
- with

FPT_LIB_EXT.1

The TOE is packaged with the third-party libraries listed in Table 13 above and requires these libraries in order to properly function.

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 components. 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 contains the required *.rpm 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 run the following commands "rpm -q CARKpsmp" and "rpm -q CARKaim" for PSMP and OPM respectfully. The TOE will not automatically download or apply new packages that would replace or update its code.

The installation packages are digitally signed to protect them from alteration after publication. To verify the digital signature of a TOE package, users must complete the following:

1. Import the RPM-GPG-KEY-CyberArk public key that is provided with the installation package by running the "rpm --import RPM-GPG-KEY-CyberArk" command.
2. Verify the signature of the package, by running the "rpm -K -v <package_name.rpm>" command.

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 (.rpm) extracted from the archive file (.zip). The TOE software can be removed from the platform using the "rpm -e CARKpsmp" and "rpm -e CARKaim" commands for PSMP and OPM respectfully. 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_IDV_EXT.1, FPT_LIB_EXT.1, FPT_TUD_EXT.1

8.1.7 Trusted Path/Channels

FTP_DIT_EXT.1
The TOE protects data in transit by providing trusted paths and channels using the cryptographic functions within the TOE's cryptographic libraries. The TOE provides a trusted channel between itself and target devices over SSH.

The TOE uses OpenSSH Client to create this SSH connection, which provides support using AES-128-CTR, AES-256-CTR, AES-128-CBC, or AES-256-CBC for encrypting its traffic. The SSH connection used by the TOE follows the Extended Package for Secure Shell.

The TOE provides a trusted TLS channel between itself and the EPV server. Both PSMP and OPM will act as a TLS client to connect to the EPV server over TLS when access safes that are stored in EPV.

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 for Information Technology Security Evaluations,* Version 3.1, Revision 4, September 2012. This ST conforms to the AS PP and SSH EP.

9.1.1 Variance Between the PP and this ST

There is no variance between the AS PP, SSH EP, 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 and SSH EP, including the assurance requirements listed in Section 5 of the AS PP. The TOE is a standalone application that runs on a Linux platform and is applicable to the AS PP and SSH EP.

10. Acronyms and Terms

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

10.1 Acronyms

Table 16 defines the acronyms used throughout this document.

Table 16 – Acronyms

Acronym	Definition
ACL	Access Control List
AD	Active Directory
AES	Advanced Encryption Standard
AIM	Application Identity Manager
ANSI	American National Standards Institute
API	Application Programming Interface
AS PP	Protection Profile for Application Software v1.2; April 22, 2016
ASLR	Address Space Layout Randomization
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
CRL	Certificate Revocation List
CTR	Counter Mode
DH	Diffie-Hellman
DHE	Diffie Hellman Ephemeral
DNS	Domain Name System
DRBG	Deterministic Random Bit Generator
DSS	Digital Signature Standard
EAL	Evaluation Assurance Level
EC	Elliptic Curve
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie-Hellman
ECDHE	Elliptic Curve Diffie Hellman Ephemeral

Acronym	Definition
ECDSA	Elliptic Curve Digital Signature Algorithm
EPV	Enterprise Password Vault
EST	Enrollment over Secure Transport
FIPS	Federal Information Processing Standard
GCM	Galois Counter Mode
HMAC	Hash-based Message Authentication Code
ICU	International Components for Unicode
ID	Identification
IP	Internet Protocol
IT	Information Technology
ITSEF	Information Technology Security Evaluation Facility
MAC	Message Authentication Code
MIME	Multipurpose Internet Mail Extensions
N/A	Not Applicable
NIST	National Institute of Standards and Technology
OCSP	Online Certificate Status Protocol
OE	Operating Environment
OID	Object Identifier
OPM	On-Demand Privileges Manager
OS	Operating System
OSP	Organizational Security Policy
PAS	Privileged Access Security
PCRE	Perl Compatible Regular Expressions
PII	Personally Identifiable Information
PIM	Privileged Identity Manager
PP	Protection Profile
PSMP	Privileged Session Manager SSH Proxy
PUB	Publication
R2	Release Two
RA	Registration Authority
RBG	Random Bit Generation
RFC	Request for Comments
RHEL	Red Hat Enterprise Linux
RSA	Rivest, Shamir, Adleman

Acronym	Definition
SAR	Security Assurance Requirement
SCP	Secure Copy
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm
SP	Service Pack
SP	Special Publication
SSH	Secure Shell
SSH EP	Extended Package for Secure Shell (SSH) v1.0; 2016-02-19
ST	Security Target
TD	Technical Decisions
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSFI	TOE Security Function Interface

10.2 Terms

Table 17 defines the terms used throughout this document.

Table 17 – 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 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.

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