Enveil ZeroReveal® Compute Fabric Server v4.6.3 Security Target

Version: 2.1

Date: 11 April 2024

Enveil
6000 Headquarters Drive,
Suite 600
Plano, Texas 75024
www.enveil.com

2400 Research Blvd
Suite 395
Rockville, MD 20850
www.acumensecurity.net

Table Of Contents

Table Of Figures

Table Of Tables

Revision History

Version	Date	Description
1.0	03.15.2023	Initial Draft
1.1	05.15.2023	Added remote DB
1.2	06.15.2023	Vendor mods
1.3	07.24.2023	Added TDs.
1.4	09.14.2023	Added TDs.
1.5	11.10.2023	Addressed ECRs and added CAVP #
1.6	11.29.2023	Addressed ECRs
1.7	01.21.2024	Minor mods
1.8	03.28.2024	Minor mods
1.9	04.08.2024	Addressed Client ECRs.
2.0	04.09.2024	Made mods per review.
2.1	04.11.2024	Addressed TD0822 and TD0833.

1 troduction

The Security Target (ST) serves as the basis for the Common Criteria (CC) evaluation and identifies the Target of Evaluation (TOE), the scope of the evaluation, and the assumptions made throughout. This document will also describe the intended operational environment of the TOE, and the functional and assurance requirements that the TOE meets.

1.1 Security Target and TOE Reference

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

Table 1: TOE and ST Identification

Category	Identifier
ST Title	Enveil ZeroReveal® Compute Fabric Server v4.6.3 Security Target
ST Version	2.1
ST Date	11 April 2024
ST Author	Acumen Security, LLC.
TOE Identifier	Enveil ZeroReveal® Compute Fabric Server v4.6.3
TOE Software Version	4.6.3
TOE Developer	Enveil
Key Words	Enveil, ZeroReveal, APP_PP, TLS_PKG

1.2 TOE Overview

The TOE is the Enveil ZeroReveal Compute Fabric Server (otherwise referred to as the ZeroReveal Server, or the TOE) software application which communicates to one or more instances of the Enveil ZeroReveal Compute Fabric Client software application via REST API over mutually authenticated HTTPS over TLS.

The TOE is a homomorphic encryption engine for database queries. In normal database operation, a query is submitted in plain text, and a plain text answer retrieved for the querier. While the communication between the querier and the database engine itself may be transmitted through a tunnel such as IPsec, TLS, or SSH, the contents of the query are always in plaintext. The ZeroReveal Compute Fabric Client (evaluated separately) takes an authenticated user's database query and encrypts it using Enveil's proprietary homomorphic encryption process. This encrypted query is passed via a mutually authenticated TLS trusted channel from ZeroReveal Client to ZeroReveal Server. The encrypted query is never decrypted during this process, which prevents ZeroReveal Server and its owners/administrators from being able to tell what the query was searching for and what items in the database (if any) matched the query. The output of this process is an encrypted response that is sent back to ZeroReveal Client. In this way, the database itself is not strictly aware of what the query was and no individual point in the chain between the user and the information know what was requested.

The ZeroReveal Server (the TOE) and ZeroReveal Client are evaluated as software applications only and the homomorphic encryption techniques used for the ZeroReveal Client and ZeroReveal Server operations are outside the scope this evaluation.

1.3 TOE Description

1.3.1 Evaluated Configuration

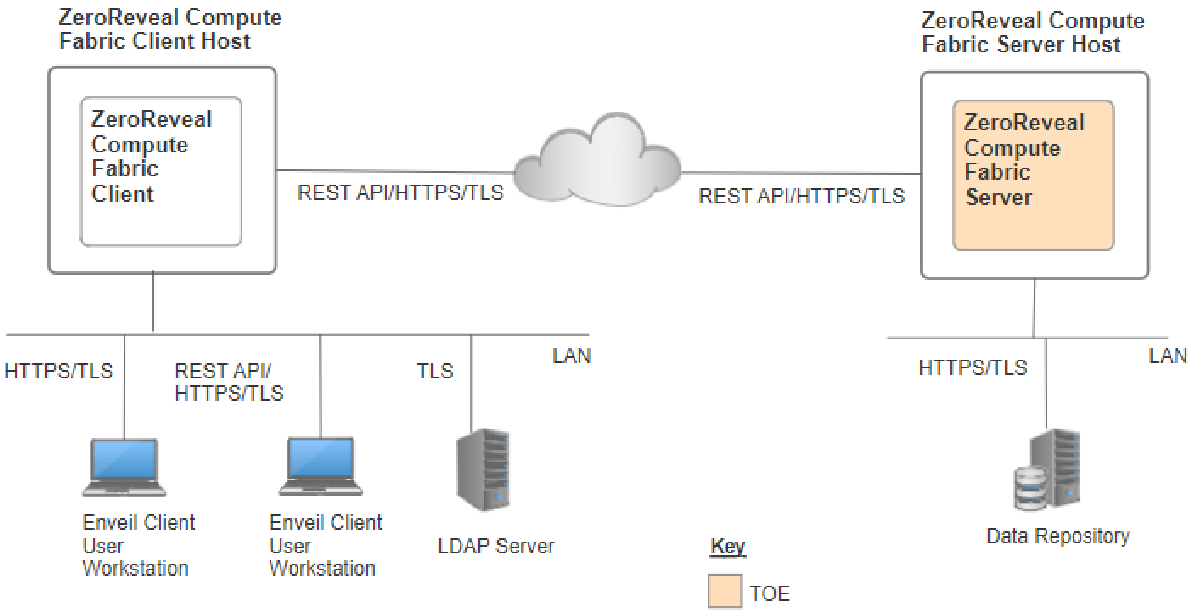
The TOE has been evaluated on the following host platform:

- Rocky Linux 8.7 with SELinux OS on Intel Core i7-10710U (Comet Lake)

1.3.2 Physical Boundary

The diagram below depicts a representative TOE deployment.

Figure 1: Representative TOE Deployment


ZeroReveal Compute Fabric Client Host
ZeroReveal Compute Fabric Client
REST API/HTTPS/TLS
REST API/HTTPS/TLS
ZeroReveal Compute Fabric Server Host
ZeroReveal Compute Fabric Server
HTTPS/TLS
REST API/ HTTPS/TLS
TLS
LAN
HTTPS/TLS
LAN
Enveil Client User Workstation
Enveil Client User Workstation
LDAP Server
Key
TOE
Data Repository

The following items are required for the operational environment.

Table 2: Hardware and Software Environmental Components

Components	Mandatory/ Optional	Description
Hardware		
Enveil ZeroReveal® Compute Fabric Server 4.6.3 Host	Mandatory	The hardware running the TOE. The Server platform must include OpenJDK 8 JRE and Rocky Linux 8.7 with SELinux operating system installed.
Local Access	Mandatory	Local access to the ZeroReveal Server platform that enables an administrator to modify configuration files using a text editor and read log files. Access is via the local keyboard.
Enveil ZeroReveal® Compute Fabric Client 4.6.3 software and host platform	Mandatory	The Enveil ZeroReveal Client application which communicates with the ZeroReveal Server to process data queries. The TOE communicates with the ZeroReveal Client by receiving REST API commands sent using HTTPS over TLS.

Components	Mandatory/ Optional	Description
Remote Data Repository	Mandatory	A remotely installed and configured database containing information against which ZeroReveal queries are executed. The TOE communicates with the remote database using HTTPS over TLS.
Software		
Rocky Linux 8.7 with SELinux OS	Mandatory	The operating system installed on the TOE's host.
OpenJDK 8	Mandatory	Java Platform that includes the Java Runtime Environment (JRE) installed on the TOE's host.

The TOE is the ZeroReveal Compute Fabric Server software that includes the following libraries:

- Java JSSE Library 8
- Bouncy Castle FIPS Provider v1.0.2.3
- Bouncy Castle FIPS TLS Provider v1.0.12.3
- GMP Library v6.2.0
- SEAL Homomorphic Encryption Library v3.7.2.0

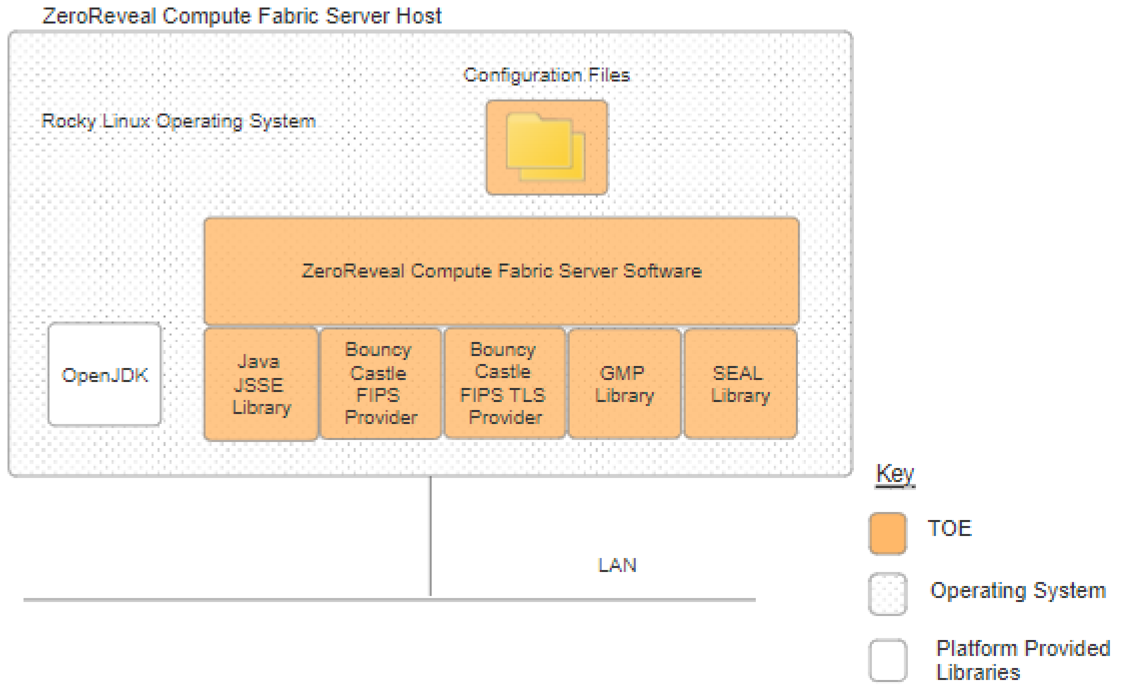
Additionally, the TOE boundary includes configuration files that include key strings that must be completed to configure the TOE in the evaluated configuration. The configuration files are modified by administrators and are accessed using the local keyboard.

The TOE's operational environment requires the TOE platform to have:

- Rocky Linux 8.7 with SELinux installed and running and
- OpenJDK 8 JRE installed.

The following diagram depicts the TOE and the Operational Environment of the ZeroReveal Compute Fabric Server Host.

Figure 2: ZeroReveal Server Host


ZeroReveal Compute Fabric Server Host
Configuration Files
Rocky Linux Operating System
ZeroReveal Compute Fabric Server Software
OpenJDK
Java JSSE Library
Bouncy Castle FIPS Provider
Bouncy Castle FIPS TLS Provider
GMP Library
SEAL Library
LAN
Key
TOE
Operating System
Platform Provided Libraries

1.3.3 Logical Boundary

The TOE provides the security functions required by the *Protection Profile for Application Software*, Version 1.4 and the *Functional Package for Transport Layer Security (TLS)*, Version 1.1.

1.3.3.1 Cryptographic Support

The cryptographic services provided by the TOE are described below.

Table 3: Provided Cryptography

Cryptographic Method	Use within the TOE
AES-GCM	TLS encryption
ECDSA	TLS key generation, signature generation and verification
RSA	TLS key generation, signature generation and verification
HMAC	Message integrity and authentication for TLS
AES-CCM	Storage of credentials
DRBG	Random bit generation for all cryptographic functions

Each of these cryptographic algorithms have been validated for conformance to the requirements specified in their respective standards (refer to Table 6.2).

1.3.3.2 User Data Protection

The ZeroReveal Server network communication is restricted to user-initiated communication for responses to API requests from ZeroReveal Clients and accessing the remote database using TLS. Credentials are stored locally, encrypted using AES algorithm in CCM mode.

1.3.3.3 Identification and Authentication

The ZeroReveal server performs X.509v3 certificate validation functions to authenticate the certificate(s) during the establishment of the TLS trusted channels.

1.3.3.4 Security Management

Administrators manages the TOE via configuration files on each installation platform. The access interface and file editor used to modify the files is outside the scope of the TOE.

The TOE does not include any predefined or default credentials and utilizes the platform recommended storage process for configuration files.

1.3.3.5 Privacy

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

1.3.3.6 Protection of the TSF

The TOE leverages platform provided package management for secure installation and updates. The TOE installation package includes only those third-party libraries necessary for its intended operation. The TOE is designed to utilize compiler-provided anti-exploitation capabilities.

1.3.3.7 Trusted Path/Channels

The TOE communicates to the ZeroReveal® Compute Fabric Client via REST API over mutually authenticated HTTPS over TLS and stores data in a remote database using TLS. Administrators configure the TOE via local access only, making changes to configuration files.

1.3.4 TOE Documentation

- *Enveil ZeroReveal® Compute Fabric Configuration Guide for Common Criteria v3.1*, Version 4.6.3.

1.4 Product Functionality not Included in the Scope of the Evaluation

The TOE is a software application, and as such many of the functions of the application itself are out of scope of a Common Criteria Evaluation. The following functionality is explicitly excluded from the scope of evaluation; it was not evaluated during the common criteria evaluation, and no claims are made regarding the applicability, suitability, or functionality of the following TOE functions:

- The homomorphic encryption process, including the algorithms, uses and the security strength of the resultant ciphertext.
- The user interface to modify the local configuration files.

2 Conformance Claims

2.1 CC Conformance

This TOE is conformant to the following:

- *Common Criteria for Information Technology Security Evaluations Part 2: Security functional components*, Version 3.1, Revision 5, April 2017: Part 2 extended.
- *Common Criteria for Information Technology Security Evaluations Part 3: Security assurance components*, Version 3.1, Revision 5, April 2017: Part 3 conformant.

2.2 Protection Profile Conformance

This ST and the TOE is describes claim exact conformance to the following CC specifications:

- *Protection Profile for Application Software*, Version 1.4, 07 October 2021 [AppPP] with the following optional and selection based SFRs:
    - FCS_CKM.1/AK
    - FCS_CKM.1/SK
    - FCS_CKM.2
    - FCS_COP.1/Hash
    - FCS_COP.1/KeyedHash
    - FCS_COP.1/Sig
    - FCS_COP.1/SKC
    - FCS_HTTPS_EXT.1/Client
    - FCS_HTTPS_EXT.1/Server
    - FCS_HTTPS_EXT.2
    - FCS_RBG_EXT.2
    - FIA_X509_EXT.1
    - FIA_X509_EXT.2
    - FPT_TUD_EXT.2
- *Functional Package for Transport Layer Security (TLS)*, Version 1.1, 01 March 2019 [TLSPkg] with the following optional and selection based SFRs:
    - FCS_TLSC_EXT.1
    - FCS_TLSC_EXT.2
    - FCS_TLSC_EXT.3
    - FCS_TLSC_EXT.5
    - FCS_TLSS_EXT.1
    - FCS_TLSS_EXT.2
    - FCS_TLSS_EXT.3

2.3 Conformance Rationale

This Security Target provides exact conformance to Version 1.4 of the *Protection Profile for Application Software* and Version 1.1 of the *Functional Package for Transport Layer Security (TLS)*. The security problem definition and security objectives in this Security Target are taken from the Protection Profile unmodified. The security requirements in this Security Target are all taken from the Protection Profile and Functional Package performing only operations defined there.

The security requirements in this Security Target are all taken from the Protection Profile and Functional Package performing only operations defined there. All mandatory SFRs are claimed. The [AppPP] and [TLSPkg] Selection-Based SFRs are claimed and are consistent with the selections made in the mandatory SFRs that prompt their inclusion. Additionally, the [AppPP]'s optional FCS_CKM.1/SK SFR is claimed and the [TLSPkg]'s objective SFR FCS_TLSS_EXT.3 is claimed.

2.3.1 Technical Decisions

The following table identifies the NIAP Technical Decisions that apply to the TOE and have been accounted for in the ST development and the conduct of the evaluation or were considered to be non-applicable.

Table 4: Applicable Technical Decisions

Identifier	Applicable	Notes/Exclusion Rationale (if applicable)
App PP		
TD0823 Update to Microsoft Windows Exploit Protection link in FPT_AEX_EXT.1.3	No	The TD applies to Microsoft Windows applications only.
TD0822 Correction to Windows Manifest File for FDP_DEC_EXT.1	No	The TD applies to Microsoft Windows applications only.
TD0815: Addition of Conditional TSS Activity for FPT_AEX_EXT.1.5.	Yes	Applies to AA only.
TD0798: Static Memory Mapping Exceptions	Yes	Applies to Test and AA only.
TD0780: FIA_X509_EXT.1 Test 4 Clarification	Yes	Applies to Test only. Archives TD0669.
TD0756: Update for platform-provided full disk encryption	Yes	Applies to Test only.
TD0747: Configuration Storage Option for Android	No	The TOE does not run on Android systems.
TD0743: FTP_DIT_EXT.1.1 Selection exclusivity	Yes	Archives TD0655.
TD0736: Number of elements for iterations of FCS_HTTPS_EXT.1	Yes	Archives TD0709.
TD0719: ECD for PP APP V1.3 and V1.4	Yes	
TD0717: Format changes for PP_APP_V1.4	Yes	Archives TD0659 and TD0626. Applies to SFRs, Test, and AA.
TD0664: Testing activity for FPT_TUD_EXT.2.2	Yes	Applies to Test only.
TD0650: Conformance claim sections updated to allow for MOD_VPNC_V2.3 and 2.4	No	VPN configuration is not claimed for the TOE.
TD0628: Addition of Container Image to Package Format	Yes	Applies to SFR and Test.
TLS Package		
TD0779: Updated Session Resumption Support in TLS package V1.1.	Yes	Archives TD0588
TD0770: TLSS.2 connection with no client cert	Yes	Applies to SFRs, Tests, and AA.
TD0739: PKG_TLS_V1.1 has 2 different publication dates	Yes	Applies to Test only.

Identifier	Applicable	Notes/Exclusion Rationale (if applicable)
TD0726: Correction to (D) TLSS SFRs in TLS 1.1 FP	Yes	Applies to SFRs only.
TD0513: CA Certificate Loading	Yes	Applies to Test only.
TD0499: Testing with Pinned certificates	Yes	Applies to Test only.
TD0469: Modification of test activity for FCS_TLSS_EXT.1.1 test 4.1	Yes	Applies to Test only.
TD0442: TLS Ciphersuites for TLS Package	Yes	Applies to SFRs only,

3 Security Problem Definition

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

3.1 Threats

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

Table 5: Threats

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

3.2 Assumptions

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

Table 6: Assumptions

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

3.3 Organizational Security Policies

There are no OSPs for the application.

4 Security Objectives

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

4.1 Security Objectives for the TOE

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

Table 7: Security Objectives

ID	TOE Objective
O.INTEGRITY	Conformant TOEs ensure the integrity of their installation and update packages, and also leverage execution environment-based mitigations. Software is seldom, if ever, shipped without errors. The ability to deploy patches and updates to fielded software with integrity is critical to enterprise network security. Processor manufacturers, compiler developers, execution environment vendors, and operating system vendors have developed execution environment-based mitigations that increase the cost to attackers by adding complexity to the task of compromising systems. Application software can often take advantage of these mechanisms by using APIs provided by the runtime environment or by enabling the mechanism through compiler or linker options.
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.
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.
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.
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.

4.2 Security Objectives for the Operational Environment

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

Table 8: Objectives for the Operational Environment

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

5 Security Requirements

This section identifies the Security Functional Requirements (SFRs) for the TOE. The Security Functional Requirements included in this section are derived from Part 2 of the *Common Criteria for Information Technology Security Evaluation*, Version 3.1, Revision 5; the *Protection Profile for Application Software,* Version 1.4; and the *Functional Package for Transport Layer Security (TLS), Version 1.1* and all international interpretations.

5.1 Extended Requirements

All the extended requirements in this ST have been drawn from the [AppPP] or [TLSPkg] and are itemized below. This document defines the extended SFRs; since they have not been redefined in this ST, the [PP_APP] or [PKG_TLS] should be consulted for more information regarding these extensions to CC Parts 2 and 3.

Extended requirements from the [AppPP].

- FCS_CKM_EXT.1
- FCS_HTTPS_EXT.1/Client
- FCS_HTTPS_EXT.1/Server
- FCS_HTTPS_EXT.2
- FCS_RBG_EXT.1
- FCS_RBG_EXT.2
- FCS_STO_EXT.1
- FCS_TLS_EXT.1
- FDP_DAR_EXT.1
- FDP_DEC_EXT.1
- FDP_NET_EXT.1
- FIA_X509_EXT.1
- FIA_X509_EXT.2
- FMT_CFG_EXT.1
- FMT_MEC_EXT.1
- FPR_ANO_EXT.1
- FPT_AEX_EXT.1
- FPT_API_EXT.1
- FPT_IDV_EXT.1
- FPT_LIB_EXT.1
- FPT_TUD_EXT.1
- FPT_TUD_EXT.2
- FTP_DIT_EXT.1

Extended requirements from the [TLSPkg].

- FCS_TLSC_EXT.1
- FCS_TLSC_EXT.2
- FCS_TLSC_EXT.3
- FCS_TLSC_EXT.5
- FCS_TLSS_EXT.1
- FCS_TLSS_EXT.2

- FCS_TLSS_EXT.3

5.2 Conventions

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

- Assignment: Indicated with *italicized* text.
- Selection: Indicated with underlined text.
- Refinement: Indicated with **bold** text for additions and ~~strike-through~~ for deletions.
- Iteration: Indicated by a slash followed by a descriptor for the purpose of the iteration. For example, FCS_HTTPS_EXT.1/Client indicates that the FCS_HTTPS_EXT.1 requirement applies specifically to HTTPS client functionality.
- The ST does not show operations that have been completed by the PP authors, though it does preserve brackets to show where such operations have been made.
- The ST does not retain the formatting conventions of the PPs.

5.3 Security Functional Requirements (SFRs)

Table 9: SFRs

Requirement	Description
FCS_CKM_EXT.1[1]	Cryptographic Key Generation Services
FCS_CKM.1/AK	Cryptographic Asymmetric Key Generation
FCS_CKM.1/SK	Cryptographic Symmetric Key Generation
FCS_CKM.2	Cryptographic Key Establishment
FCS_COP.1/Hash	Cryptographic Operation - Hashing
FCS_COP.1/KeyedHash	Cryptographic Operation - Keyed-Hash Message Authentication
FCS_COP.1/Sig	Cryptographic Operation - Signing
FCS_COP.1/SKC	Cryptographic Operation - Encryption/Decryption
FCS_HTTPS_EXT.1/Client	HTTPS Protocol
FCS_HTTPS_EXT.1/Server	HTTPS Protocol
FCS_HTTPS_EXT.2	HTTPS Protocol with Mutual Authentication
FCS_RBG_EXT.1	Random Bit Generation Services
FCS_RBG_EXT.2	Random Bit Generation from Application
FCS_STO_EXT.1	Storage of Credentials
FCS_TLS_EXT.1	TLS Protocol
FCS_TLSC_EXT.1	TLS Client Protocol
FCS_TLSC_EXT.2	TLS Client Support for Mutual Authentication
FCS_TLSC_EXT.3	TLS Client Support for Signature Algorithms Extension
FCS_TLSC_EXT.5	TLS Client Support for Supported Groups Extension

[1] Applied TD0717 renaming FCS_CKM.1 to FCS_CKM_EXT.1.

Requirement	Description
FCS_TLSS_EXT.1	TLS Server Protocol
FCS_TLSS_EXT.2	TLS Server Support for Mutual Authentication
FCS_TLSS_EXT.3	TLS Server Support for Signature Algorithms Extension
FDP_DAR_EXT.1	Encryption Of Sensitive Application Data
FDP_DEC_EXT.1	Access to Platform Resources
FDP_NET_EXT.1	Network Communications
FIA_X509_EXT.1	X.509 Certificate Validation
FIA_X509_EXT.2	X.509 Certificate Authentication
FMT_CFG_EXT.1	Secure by Default Configuration
FMT_MEC_EXT.1	Supported Configuration Mechanism
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_IDV_EXT.1	Software Identification and Versions
FPT_LIB_EXT.1	Use of Third Party Libraries
FPT_TUD_EXT.1	Integrity for Installation and Update
FPT_TUD_EXT.2	Integrity for Installation and Update
FTP_DIT_EXT.1	Protection of Data in Transit

5.3.1 Cryptographic Support (FCS)

FCS_CKM_EXT.1[2] Cryptographic Key Generation Services

FCS_CKM_EXT.1.1[3]
The application shall [implement asymmetric key generation].

FCS_CKM.1/AK Cryptographic Asymmetric Key Generation

FCS_CKM.1.1/AK[4]
The application shall [implement functionality] to generate asymmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm [

- [RSA schemes] using cryptographic key sizes of [2048-bit and 3072-bits[5] or greater] that meet the following: [FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3],
- [ECC schemes] using ["NIST curves" P-384 and [P-256]] that meet the following: [FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4]

].

[2] Applied TD0717.
[3] Applied TD0717.
[4] Applied TD0717.
[5] Applied CSfC selection (v11.3.2022)

FCS_CKM.1/SK Cryptographic Symmetric Key Generation

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

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_COP.1/Hash Cryptographic Operation - Hashing

FCS_COP.1.1/Hash[6]
The application shall perform [cryptographic hashing services] in accordance with a specified cryptographic algorithm [

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

] and message digest sizes [

- 256,
- 384,
- 512

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

FCS_COP.1/KeyedHash Cryptographic Operation - Keyed-Hash Message Authentication

FCS_COP.1.1/KeyedHash[7]
The application shall perform [keyed-hash message authentication] in accordance with a specified cryptographic algorithm [

- HMAC-SHA-256
- HMAC-SHA-384
- HMAC-SHA-512

] and [

- no other algorithm

] with key sizes [*256 bits, 384 bits, 512 bits*] and message digest sizes [256, 384, 512] and [no other size] bits that meet the following: [FIPS Pub 198-1, 'The Keyed-Hash Message Authentication Code' and FIPS Pub 180-4 'Secure Hash Standard'].

[6] Applied TD0717.

[7] Applied TD0717.

FCS_COP.1/Sig Cryptographic Operation - Signing

FCS_COP.1.1/Sig[8]
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 and 3072 bits or greater[9]] that meet the following: [FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5],
- ECDSA schemes using ["NIST curves" P-256, P-384 and [no other curves]] that meet the following: [FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6]

] .

FCS_COP.1/SKC Cryptographic Operation - Encryption/Decryption

FCS_COP.1.1/SKC[10]
The application shall perform [encryption/decryption] in accordance with a specified cryptographic algorithm [

- AES-CCM (as defined in NIST SP 800-38C) mode,
- AES-GCM (as defined in NIST SP 800-38D) mode

] and cryptographic key sizes [256-bit].

FCS_HTTPS_EXT.1/Client HTTPS Protocol

FCS_HTTPS_EXT.1.1/Client
The application shall implement the HTTPS protocol that complies with RFC 2818.

FCS_HTTPS_EXT.1.2/Client
The application shall implement HTTPS using TLS as defined in the TLS package.

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

FCS_HTTPS_EXT.1/Server HTTPS Protocol

FCS_HTTPS_EXT.1.1/Server
The application shall implement the HTTPS protocol that complies with RFC 2818.

FCS_HTTPS_EXT.1.2/Server
The application shall implement HTTPS using TLS as defined in the TLS package.

FCS_HTTPS_EXT.1.3/Server [11]
The application shall [establish or not establish the connection based on an administrative or user setting] if the peer certificate is deemed invalid.

[8] Applied TD0717.

[9] Applied CSfC selections.

[10] Applied TD0717.

[11] Applied TD0736.

FCS_HTTPS_EXT.2 HTTPS Protocol with Mutual Authentication

FCS_HTTPS_EXT.2.1
The application shall [establish or not establish the connection based on an administrative or user setting[12]] if the peer certificate is deemed invalid.

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 [HMAC_DRBG (any)].

FCS_RBG_EXT.2.2
The deterministic RBG shall be seeded by an entropy source that accumulates entropy from a platform-based DRBG and [

- no other noise source

] with a minimum of [

- 256 bits

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

FCS_STO_EXT.1 Storage of Credentials

FCS_STO_EXT.1.1
The application shall [

- implement functionality to securely store *[TLS server and client certificates and private keys]* according to [FCS_COP.1/SKC]

] to non-volatile memory.

FCS_TLS_EXT.1 TLS Protocol

FCS_TLS_EXT.1.1
The product shall implement [

- TLS as a client,
- TLS as a server

].

[12] Added per CSfC TLS Software Application selections document.

FCS_TLSC_EXT.1 TLS Client Protocol

FCS_TLSC_EXT.1.1[13]
The product shall implement TLS 1.2 (RFC 5246) and [no earlier TLS versions] as a client that supports the cipher suites [

- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

] and also supports functionality for [mutual authentication].

FCS_TLSC_EXT.1.2
The product shall verify that the presented identifier matches the reference identifier according to RFC 6125.

FCS_TLSC_EXT.1.3
The product shall not establish a trusted channel if the server certificate is invalid [with no exceptions].

FCS_TLSC_EXT.2 TLS Client Support for Mutual Authentication

FCS_TLSC_EXT.2.1

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

FCS_TLSC_EXT.3 TLS Client Support for Signature Algorithms Extension

FCS_TLSC_EXT.3.1

The product shall present the signature_algorithms extension in the Client Hello with the supported_signature_algorithms value containing the following hash algorithms: [SHA384, SHA512] and no other hash algorithms.

FCS_TLSC_EXT.5 TLS Client Support for Supported Groups Extension

FCS_TLSC_EXT.5.1
The product shall present the Supported Groups Extension in the Client Hello with the supported groups [secp384r1].

FCS_TLSS_EXT.1 TLS Server Protocol

FCS_TLSS_EXT.1.1[14] [15]
The product shall implement TLS 1.2 (RFC 5246) and [no earlier TLS versions] as a server that supports the cipher suites [

- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

] and no other cipher suites, and also supports functionality for [

- mutual authentication,
- session resumption based on session tickets according to RFC 4346 (TLS1.1) or RFC 5246 (TLS 1.2)

[13] Applied TD0442.

[14] Applied TD0442.

[15] Applied TD0779.

].

FCS_TLSS_EXT.1.2
The product shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0 and [TLS 1.1].

FCS_TLSS_EXT.1.3[16]
The product shall perform key establishment for TLS using [ECDHE parameters using elliptic curves [secp384r1] and no other curves].

FCS_TLSS_EXT.2 TLS Server Support for Mutual Authentication

FCS_TLSS_EXT.2.1
The product shall support authentication of TLS clients using X.509v3 certificates.

FCS_TLSS_EXT.2.2[17]
The product shall not [not establish a trusted channel] establish a trusted channel if the client certificate is invalid.

FCS_TLSS_EXT.2.3
The product shall not establish a trusted channel if the Distinguished Name (DN) or Subject Alternative Name (SAN) contained in a certificate does not match one of the expected identifiers for the client.

FCS_TLSS_EXT.3 TLS Server Support for Signature Algorithms Extension

FCS_TLSS_EXT.3.1
The product shall present the HashAlgorithm enumeration in supported_signature_algorithms in the Certificate Request with the following hash algorithms: [SHA256, SHA384] and no other hash algorithms.

5.3.2 User Data Protection (FDP)

FDP_DAR_EXT.1 Encryption Of Sensitive Application Data

FDP_DAR_EXT.1.1
The application shall [

- leverage platform-provided functionality to encrypt sensitive data,
- 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 [*[no sensitive information repositories]*].

FDP_NET_EXT.1 Network Communications

FDP_NET_EXT.1.1
The application shall restrict network communication to [

[16] Applied TD0726.

[17] Applied TD0770.

- respond to *[REST API requests sent via HTTPS over TLS from Enveil ZeroReveal Compute Fabric Clients]*
- *[initiate communication to remote database via HTTPS over TLS]*

].

5.3.3 Identification and Authentication (FIA)

FIA_X509_EXT.1 X.509 Certificate Validation

FIA_X509_EXT.1.1
The application shall [*implement functionality*] to validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation.
- The certificate path must terminate with a trusted CA certificate.
- The application shall validate a certificate path by ensuring the presence of the basicConstraints extension, that the CA flag is set to TRUE for all CA certificates, and that any path constraints are met.
- The application shall validate that any CA certificate includes caSigning purpose in the key usage field.
- The application shall validate the revocation status of the certificate using [CRL as specified in RFC 8603].
- The application shall validate the extendedKeyUsage (EKU) field according to the following rules:
  - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing Purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
  - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the EKU field.
  - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the EKU field.
  - S/MIME certificates presented for email encryption and signature shall have the Email Protection purpose (id-kp 4 with OID 1.3.6.1.5.5.7.3.4) in the EKU field.
  - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-dp 9 with OID 1.3.6.1.5.5.7.3.9) in the EKU field.
  - Server certificates presented for EST shall have the CMC Registration Authority (RA) purpose (id-kp-cmcRA with OID 1.3.6.1.5.5.7.3.28) in the EKU field.

FIA_X509_EXT.1.2
The application shall treat a certificate as a CA certificate only if the basicConstraints extension is present and the CA flag is set to TRUE.

FIA_X509_EXT.2 X.509 Certificate Authentication

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

FIA_X509_EXT.2.2
When the application cannot establish a connection to determine the validity of a certificate, the application shall [allow the administrator to choose whether to accept the certificate in these cases].

5.3.4 Security Management (FMT)

FMT_CFG_EXT.1 Secure by Default Configuration

FMT_CFG_EXT.1.1
The application shall provide only enough functionality to set new credentials when configured with default credentials or no credentials.

FMT_CFG_EXT.1.2
The application shall be configured by default with file permissions which protect the application binaries and data files from modification by normal unprivileged users.

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].

5.3.5 Privacy (FPR)

FPR_ANO_EXT.1 User Consent for Transmission of Personally Identifiable Information

FPR_ANO_EXT.1.1
The application shall [not transmit PII over a network].

5.3.6 Protection of TSF (FPT)

FPT_AEX_EXT.1 Anti-Exploitation Capabilities

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

FPT_AEX_EXT.1.2
The application shall [allocate memory regions with write and execute permissions for only *[Java runtime performing just-in-time compilation]*].

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

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

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

FPT_API_EXT.1 Use of Supported Services and APIs

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

FPT_IDV_EXT.1 Software Identification and Versions

FPT_IDV_EXT.1.1
The application shall be versioned with [*[major_version.minor_version.patch_number]*].

FPT_LIB_EXT.1 Use of Third Party Libraries

FPT_LIB_EXT.1.1
The application shall be packaged with only [

- *Java JSSE Library v8*
- *Bouncy Castle FIPS Provider v1.0.2.3*
- *Bouncy Castle FIPS TLS Provider v1.0.12.3*
- *GMP Library v6.2.0*
- *SEAL Homomorphic Encryption Library v3.7.2.0*

].

FPT_TUD_EXT.1 Integrity for Installation and Update

FPT_TUD_EXT.1.1
The application shall [leverage the platform] to check for updates and patches to the application software.

FPT_TUD_EXT.1.2
The application shall [leverage the platform] to query the current version of the application software.

FPT_TUD_EXT.1.3
The application shall not download, modify, replace or update its own binary code.

FPT_TUD_EXT.1.4
Application updates shall be digitally signed such that the application platform can cryptographically verify them prior to installation.

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

FPT_TUD_EXT.2 Integrity for Installation and Update

FPT_TUD_EXT.2.1[18]
The application shall be distributed using [the format of the platform-supported package manager].

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

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

[18] Applied TD0628.

5.3.7 Trusted Path/Channel (FTP)

FTP_DIT_EXT.1 Protection of Data in Transit

FTP_DIT_EXT.1.1[19]
The application shall [

- encrypt all transmitted [data] with [
    - HTTPS as a server using mutual authentication in accordance with FCS_HTTPS_EXT.2 for [*communication with the Enveil ZeroReveal Client sending REST API commands*],
    - TLS as a server as defined in the Functional Package for TLS and also supports functionality for [*communication with the Enveil ZeroReveal Client*],
    - HTTPS as a client in accordance with FCS_HTTPS_EXT.1/Client [*communicating with the remote database*],
    - TLS as a client as defined in the Functional Package for TLS [*communicating with the remote database*]

  ]

] between itself and another trusted IT product.

5.4 TOE SFR Dependencies Rationale for SFRs

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

5.5 Security Assurance Requirements

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

Table 10: Security Assurance Requirements

Assurance Class	Components	Components Description
Development	ADV_FSP.1	Basic functional specification
Guidance documents	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative procedures
Life-cycle support	ALC_CMC.1	Labeling of the TOE
	ALC_CMS.1	TOE CM coverage
Security Target evaluation	ASE_CCL.1	Conformance claims
	ASE_ECD.1	Extended components definition
	ASE_INT.1	ST introduction
	ASE_OBJ.1	Security objectives for the operational environment
	ASE_REQ.1	Stated security requirements
	ASE_SPD.1	Security problem definition

[19] Applied TD0743.

Assurance Class	Components	Components Description
	ASE_TSS.1	TOE summary specification
Tests	ATE_IND.1	Independent testing – conformance
Vulnerability assessment	AVA_VAN.1	Vulnerability survey

5.6 Rationale for Security Assurance Requirements

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

6 TOE Summary

6.1 TOE Summary Specification

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

Table 11: TOE Summary Specification

SFR	Rationale
FCS_CKM_EXT.1 FCS_CKM.1/AK FCS_CKM.2 FCS_COP.1/Sig	The TOE implements ECDSA Key Generation, Signature Generation, and Signature Verification as part of TLS trusted channel establishment. NIST curves P-256 and P-384 are supported. The TOE implements RSA Key Generation, Signature Generation and Signature Verification as part of TLS trusted channel establishment. Key sizes of 2048-bits and 3072-bits and greater are supported. Key establishment for TLS is performed using Elliptic Curve Diffie-Hellman with NIST curves P-256 and P-384.
FCS_CKM.1/SK	The TOE generates symmetric AES 256-bit keys for use in AES-GCM as part of TLS and for use in AES-CCM for protection of stored credentials. Refer to the ancillary Entropy Assessment Report for information about entropy details.
FCS_COP.1/SKC	The TOE performs encryption and decryption using AES-GCM for use in TLS trusted channels and using AES-CCM for use as part of protecting stored credentials.
FCS_COP.1/Hash FCS_COP.1/KeyedHash	The TOE performs hashing and HMAC using: • SHA-256, using 256-bit message digest size as part of digital signatures. • SHA2-384 using 384-bit message digest size as part of TLS and digital signatures. • SHA2-512 using 512-bit message digest size as part of the authentication function used in key store and certificate formatting, and as the underlying DRBG function.
FCS_HTTPS_EXT.1/Client	The TOE acts as an HTTPS Client when communicating to the remote database. The TOE implements the HTTPS protocol according to RFC 2818 by implementing all SHALL, MUST, and SHOULD statements and by not implementing any SHALL NOT, MUST NOT, or SHOULD NOT statements. HTTPS is implemented using TLS 1.2 (RFC 5246). The TOE's HTTPS interfaces to the remote database rejects a connection when a server's certificate is invalid. If a Server's certificate is deemed

SFR	Rationale
	invalid, the TOE will notify the user by writing into the `/var/log/enveil/zeroreveal-server/server.log` file.
FCS_HTTPS_EXT.1/Server FCS_HTTPS_EXT.2	The TOE acts as an HTTPS Server when receiving REST API commands from the ZeroReveal Client. The TOE implements the HTTPS protocol according to RFC 2818 by implementing all SHALL, MUST, and SHOULD statements and by not implementing any SHALL NOT, MUST NOT, or SHOULD NOT statements. HTTPS is implemented using TLS 1.2 (RFC 5246). The TOE's REST interface rejects a connection if a ZeroReveal Client's certificate is invalid (mutual authentication) based on an administrator configurable parameter.
FCS_RBG_EXT.1 FCS_RBG_EXT.2	The TOE implements HMAC_DRBG functionality to generate random bits for use in the cryptographic functions. The TOE utilizes a platform based DRBG as its noise source and seeds with a minimum of 256 bits of entropy. This is achieved using the SecureRandom Java class which is configured to use the `/dev/random` system device. The CAVP details are given in Table 12. Additional information related to entropy functionality of the TOE can be reviewed in the Entropy Assessment Report (EAR) provided as an ancillary document.
FCS_STO_EXT.1	The TOE implements secure storage of TLS certificates and private keys used as part of establishing the TLS trusted channel with the Enveil ZeroReveal Client and the remote database by encrypting them with AES-CCM and storing them in `/etc/enveil/zeroreveal-server/certs`. The TOE uses its Bouncy Castle cryptographic library to encrypt/decrypt. The files are further encrypted/decrypted by the platform implanting LUKS encryption.
FCS_TLS_EXT.1 FCS_TLSC_EXT.1 FCS_TLSC_EXT.2 FCS_COP.1/Hash	The TOE acts as a TLS client when establishing connection to the remote database. When acting as a TLS client, the TOE supports mutual authentication using X.509v3 certificates. The TOE's certificate must contain the hostname or the IP address of the TOE's host machine as a Subject Alternative Name (SAN). The TOE validates the presented identifier in accordance with RFC 6125, and permits the reference identifier to be the CN, DN, or SAN-DNS. Where present, the SAN-DNS identifier supersedes the DN or CN values. Wildcards are supported, only in the leftmost label of the DNS identifier (i.e., "*.example.server.com" but not "example.*.server.com"). The TOE does not support certificate pinning. When acting as a TLS client, the TOE implements TLSv1.2 and rejects all older TLS and SSL versions, and supports the following cipher suites: • TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

SFR	Rationale
	• TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289 The TOE supports Elliptic Curves Extension in the Client Hello with the secp384r1 NIST curve. The supported curves are hardcoded and there are no configuration options. The TOE supports SHA384 and SHA512 signature hash algorithms. The TOE performs X.509v3 certification validation. The TOE will reject trusted channel establishment if the certificate is invalid.
FCS_TLSC_EXT.3	The TOE presents the signature_algorithm extension in the client_Hello message with a supported_signature_algorithms value containing only the SHA-384 and SHA-512 hash algorithms.
FCS_TLSC_EXT.5	The TOE implements the supported Groups extension with groups secp384r1 and no others.
FCS_TLS_EXT.1 FCS_TLSS_EXT.1 FCS_TLSS_EXT.2 FCS_TLSS_EXT.3	The TOE acts as a TLS server when accepting HTTPS connection requests from an Enveil ZeroReveal Client. The data sent is REST API commands. When acting as a TLS server, the TOE supports mutual authentication using X.509v3 certificates. The TOE validates the presented reference identifier in accordance with RFC 6125, and permits the reference identifier to be the CN, DN, or SAN-DNS. Where present, the SAN-DNS identifier supersedes the DN or CN values. When acting as a server, the TOE does not accept wildcards. When acting as a TLS server, the TOE performs ECDH key establishment using secp384r1 elliptic curves. The TOE supports certificate pinning to keep track of permissions for each ZeroReveal Client it has authorized to run queries. The TOE does not support session tickets. The TOE supports session resumption based on session IDs. When acting as a TLS server, the TOE implements TLSv1.2 and rejects all older versions of TLS and SSL, and supports the following cipher suites: • TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289 • TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289 The TOE supports Elliptic Curves Extension in the Client Hello with the secp384r1 NIST curves. The TOE supports SHA384 and SHA512 signature hash algorithms. The TOE performs X.509v3 certification validation. The TOE will reject trusted channel establishment if the certificate is invalid.
FDP_DAR_EXT.1	The TOE protects application log files and configuration data (stored in `/var/log/enveil/zeroreveal-server/server.log`, `/var/log/enveil/zeroreveal-server/stacks.log`, and

SFR	Rationale
	/etc/enveil/zeroreveal-server/server.conf) using Linux filesystem encryption (the platform implements LUKS to encrypt.decrypt). The log files are considered sensitive data because the files are very verbose and include certificate information. The configuration file includes passwords that enable the TOE to decrypt the files encrypted by the Linux file system (LUKS). The TOE implements secure storage of TLS certificates and private keys (stored in /etc/enveil/zeroreveal-server/certs) in accordance with FCS_STO_EXT.1 which uses the TOE's Bouncy Castle cryptographic library to encrypt with AES-CCM. The TLS certificates and private keys are encrypted again by the Linux platform provided encryption/decryption functions (LUKS).
FDP_DEC_EXT.1	The TOE does not utilize any platform resources except network functionality. The TOE does not access sensitive information repositories except for those which it has been explicitly configured to connect to by the administrator to respond to ZeroReveal Queries. The guidance documentation identifies when the TOE requires network connectivity.
FDP_NET_EXT.1	The TOE responds to authenticated REST API requests from a ZeroReveal Compute Fabric Client over TCP port 19443. The TOE initiates connections to the remote database.
FIA_X509_EXT.1	The TOE uses X.509v3 certificates to authenticate network endpoints for the HTTPS/TLS trusted channel communications. The TOE complies with RFC 5280 by implementing all SHALL, SHOULD, and MUST statements and not implementing any SHALL NOT, SHOULD NOT, or MUST NOT statements. The TOE uses the Java PKIX and Bouncy Castle validation tools. The notBefore and notAfter dates included in certificates will be checked to be before and after the current time respectively. Certificates received as part of TLS connections are checked for a valid path up to the certificate authority roots (which must have the X509v3 Basic Constraint CA: True). The TOE performs all of the required checks on trust path requirements, CA validity, key usages, and extended key usages. In the process, it ensures certificates presented for client authentication have the digitalSignature keyUsage and TLS Client extendedKeyUsage. CRL checking as specified in RFC 8603 revocation checking will be attempted on certificates that have listed distribution points. It is a configuration option for administrators to decide if failure to determine a certificate's status (if that certificate lists an endpoint and the endpoint is unreachable) should result in certificate rejection. Enveil enables this platform-provided functionality by adding the java.security.cert.PKIXRevocationChecker class to the chain of X509 TrustManagers associated with TLS contexts used to form connections.

SFR	Rationale
FIA_X509_EXT.2	The TOE uses X.509v3 certificates for TLS mutual authentication with REST API clients and the remote database. An administrator sets the certificate to be used for each distinct purpose in the TOE configuration file. When presented with an invalid certificate, the connections are accepted or rejected based on an administrator parameter.
FMT_CFG_EXT.1	The TOE is not installed with default credentials. The TOE installer package makes sure all configuration and data directories are configured with appropriate permissions to restrict modification by unprivileged users. Once the TOE has been installed, the following configuration steps must be completed: • Set up TLS for the TOE and install all necessary X.509v3 certificates in support of TLS. • Configure at least one ZeroReveal Compute Fabric Client connection. • Configure the connection to the remote database. The TOE does not provide any functionality until an administrator provides configuration files.
FMT_MEC_EXT.1	Configuration files (modifiable by a text editor) are used to manage TOE configuration. Non-functional configuration file templates are put in place by the installer package. Global configuration options are stored in the `/etc/enveil/zeroreveal-server` directory. The TOE invokes the mechanisms recommended by the platform vendor for storing and setting configuration options. The following parameters are required to be configured: Refer to Appendix B.
FMT_SMF.1	An administrator manages the TOE via configuration files on the platform. These files are included in the TOE boundary. There is no management CLI, GUI, or interface included within the scope of the evaluation.
FPR_ANO_EXT.1	The TOE does not collect or transmit PII over a network.
FPT_AEX_EXT.1	The main TOE application code is written in Java which places calls out to native C/C++ binaries. The Java binaries rely on the JRE for memory and stack protection, which are compiled into the JRE used in the OE by the JRE vendor. The two native code libraries in the TOE: SEAL and GMP. GMP and SEAL are compiled using GCC with the required compiler flags for ASLR (GCC CFLAG –fPIC, “Generate position-independent code”) and stack protection (-fstackprotector-all).

SFR	Rationale
	The memory protections for the GMP and SEAL native code portion were verified through static analysis. The TOE allocates memory regions with write and execute permissions for OpenJDK Java runtime performing just-in-time compilation. The TOE installs data and library files to `/usr/local/enveil/*` and configuration files to `/etc/enveil/*`. By default, the installed directories containing user-modifiable files do not have executables in them.
FPT_API_EXT.1	Enveil only uses public APIs in the TOE. The TOE uses the Linux APIs identified in Appendix A.
FPT_IDV_EXT.1	The TOE is versioned with version information published in the AGD. The TOE versioning methodology is ”Major Version”.”Minor Version”.”Patch Level”. The TOE’s version is major version 4, minor version 6, and patch level 3.
FPT_LIB_EXT.1	The TOE is packaged with the following libraries. No other third-party libraries are included with the TOE. • Java JSSE Library 8 • Bouncy Castle FIPS Provider v1.0.2.3 • Bouncy Castle FIPS TLS Provider v1.0.12.3 • GMP Library v6.2.0 • SEAL Homomorphic Encryption Library v3.7.2.0
FPT_TUD_EXT.1 FPT_TUD_EXT.2	Enveil will publish yum repositories for updates and patches to the TOE. The TOE relies on yum to periodically poll the repositories for updates and notify the administrator. The TOE does not check for or apply updates on its own. The TOE relies on the platform to secure communication with the Enveil repositories. If Enveil's repository server is not accessible over the network from the location of the TOE (for example, if the TOE has been installed on a machine without internet access), the enterprise will need to mirror the repositories locally and perform periodic queries of the Enveil website for announcements of important updates. The TOE supports packages running on Red Hat and Red Hat derivatives in RPM format. Official Enveil RPMs are signed using Enveil’s private signing key. When using yum to install Enveil TOE packages, the GPG signatures on the RPM files will automatically be checked. If they are missing a signature or signed with the wrong GPG key, then an error indicating that the GPG keys for the repository do not match the package will be displayed and the install will automatically abort. These checks are also run during the installation of every update. The TOE records its version in the RPM package file. An administrator can determine the current version by running the command `yum info zeroreveal-server`. The update/install packages include the required information so that the package manager will perform removal and deletion of all traces of the

SFR	Rationale
	application when an uninstall command is issued through that package manager. The TOE is updated using the platform package manager. When Enveil developers finish a new version of any component, they sign then upload it to the package repositories, which make it available to users. Updates are initiated by users via the package manager; the TOE will never download, modify, replace or update its own binary code. Enveil provides a changelog as part of the documentation accompanying every update. This changelog communicates any changes to security properties or configuration that occurred as part of the update. Enveil provides a public-facing e-mail address (bugs@enveil.com) that users can use to report security vulnerabilities involving any part of the TOE. This address is communicated to users in the ZeroReveal Platform guide and the Enveil website. A public PGP key is provided on the website at `https://enveil.com/bugs`, which can be used to encrypt reports sent to this e-mail.
FTP_DIT_EXT.1	The TOE encrypts all transmitted data. Communication between the TOE and a ZeroReveal Compute Fabric Client is via REST API over HTTPS over TLS using with mutual authentication enabled. Communication between the TOE and the remote database is via HTTPS over TLS.

6.2 CAVP Certificates

Table 12: CAVP Algorithm Testing

Algorithm	Standard	Modes Supported	CAVP Certificate
Cryptographic Asymmetric Key Generation (FCS_CKM.1/AK)			
RSA KeyGen	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3	2048 bits and 3072 bits and greater	A4651
ECC KeyGen	FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.4	Curves P-256 and P-384	A4651
Cryptographic Key Establishment (FCS_CKM.2)			
ECDHE Key Establishment	NIST SP 800-56A, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”	Curves P-256 and P-384	A4651

Algorithm	**Standard**	**Modes Supported**	**CAVP Certificate**
Cryptographic Operation – Hashing (FCS_COP.1/Hash)			
SHA2-256	FIPS Pub 180-4	Digest size 256 bits	A4651
SHA2-384	FIPS Pub 180-4	Digest size 384 bits	A4651
SHA2-512	FIPS Pub 180-4	Digest size 512 bits	A4651
Cryptographic Operation – Keyed-Hash Message Authentication (FCS_COP.1/KeyedHash)			
HMAC-SHA2-256	FIPS Pub 198-1, 'The Keyed-Hash Message Authentication Code' and FIPS Pub 180-4 'Secure Hash Standard'	Key size 256 bits, block size 512 bits, digest size 256 bits	A4651
HMAC-SHA2-384	FIPS Pub 198-1, 'The Keyed-Hash Message Authentication Code' and FIPS Pub 180-4 'Secure Hash Standard'	Key size 384 bits, block size 512 bits, digest size 384 bits	A4651
HMAC-SHA2-512	FIPS Pub 198-1, 'The Keyed-Hash Message Authentication Code' and FIPS Pub 180-4 'Secure Hash Standard'	Key size 512 bits, block size 512 bits, digest size 512 bits	A4651
Cryptographic Operation – Signing (FCS_COP.1/Sig)			
RSA	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.	2048-bit or greater	A4651
ECDSA	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6.	P-256, P-384,	A4651
Cryptographic Operation - Encryption/Decryption (FCS_COP.1/SKC)			
AES-CCM	NIST SP 800-38C	256 bits	A4651
AES-GCM	NIST SP 800-38D	256 bits	A4651
Random Bit Generation from Application (FCS_RBG_EXT.2)			
HMAC_DRBG	NIST SP 800-90A	AES-256	A4651

7 Acronyms and Abbreviations

The acronyms and abbreviations used in this document are defined below.

Table 13: Acronyms and abbreviations

Acronym	Definition
AppPP	*Protection Profile for Application Software*, Version 1.4
AES	Advanced Encryption Standard
API	Application Programming Interface
CA	Certification Authority
CAVP	Cryptographic Algorithm Validation Program
CC	Common Criteria
CM	Configuration Management
CN	Common Name
CTR	Counter-Mode
DH	Diffie-Hellman
DRBG	Deterministic Random Bit Generator
DSA	Digital Signature Algorithm
EC	Elliptic Curve
ECC	Elliptic Curve Cryptography
EP	Extended Package
FFC	Finite Field Cryptography
FIPS	Federal Information Processing Standard
FQDN	Fully Qualified Domain Name
GCM	Galois Counter Mode
GMP	GNU Multiple Precision Arithmetic
GNU	GNU's Not Unix
HMAC	Hashed Message Authentication Code
HTTP	Hyper-Text Transfer Protocol
HTTPS	Hyper-Text Transfer Protocol Security
IP	Internet Protocol
ISO	International Standards Organization
JDK	Java Development Kit
JRE	Java Runtime Environment
JSSE	Java Secure Socket Extension
LAN	Local Area Network
LUKS	Linux Unified Key Setup
MAC	Message Authentication Code
NIAP	National Information Assurance Partnership
NIST	National Institute of Standards and Technology
OCSP	Online Certificate Status Protocol
OID	Object Identifier
OS	Operating System

Acronym	Definition
PP	Protection Profile
REST	Representational State Transfer
RFC	Request For Comments
RSA	Rivest, Shamir & Adleman
SAN	Subject Alternate Name
SAR	Security Assurance Requirement
SEAL	Simple Encrypted Arithmetic Library
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm
SSH	Secure Shell Protocol
ST	Security Target
TD	Technical Decision
TLS	Transport Layer Security
TLSPkg	*Functional Package for Transport Layer Security (TLS)*, Version 1.1
TOE	Target of Evaluation
YUM	Yellowdog Updater Modified

Appendix A

Enveil only uses public APIs in the TOE. The TOE uses the following Linux APIs:

java.awt.geom.Point2D
java.beans.PropertyVetoException
java.io.BufferedInputStream
java.io.BufferedOutputStream
java.io.BufferedReader
java.io.BufferedWriter
java.io.ByteArrayInputStream
java.io.ByteArrayOutputStream
java.io.DataInputStream
java.io.DataOutputStream
java.io.EOFException
java.io.Externalizable
java.io.File
java.io.FileInputStream
java.io.FileNotFoundException
java.io.FileOutputStream
java.io.FileReader
java.io.FileWriter
java.io.FilterInputStream
java.io.IOException
java.io.InputStream
java.io.InputStreamReader
java.io.ObjectInput
java.io.ObjectInputStream
java.io.ObjectOutput
java.io.ObjectOutputStream
java.io.ObjectStreamException
java.io.OutputStream
java.io.OutputStreamWriter
java.io.PipedInputStream
java.io.PipedOutputStream
java.io.PrintWriter
java.io.Reader
java.io.SequenceInputStream
java.io.Serializable
java.io.StringWriter
java.io.UnsupportedEncodingException
java.io.Writer
java.lang.annotation.Annotation
java.lang.annotation.ElementType
java.lang.annotation.Inherited
java.lang.annotation.Repeatable
java.lang.annotation.Retention
java.lang.annotation.RetentionPolicy

java.lang.annotation.Target
java.lang.ref.WeakReference
java.lang.reflect.Array
java.lang.reflect.Field
java.lang.reflect.InvocationTargetException
java.lang.reflect.Method
java.lang.reflect.ParameterizedType
java.lang.reflect.Proxy
java.lang.reflect.Type
java.math.BigDecimal
java.math.BigInteger
java.net.ConnectException
java.net.InetAddress
java.net.MalformedURLException
java.net.Proxy
java.net.Socket
java.net.SocketException
java.net.URI
java.net.URISyntaxException
java.net.URL
java.net.UnknownHostException
java.nio.ByteBuffer
java.nio.ByteOrder
java.nio.charset.Charset
java.nio.charset.StandardCharsets
java.nio.file.DirectoryStream
java.nio.file.FileAlreadyExistsException
java.nio.file.FileStore
java.nio.file.FileSystem
java.nio.file.FileSystems
java.nio.file.Files
java.nio.file.NoSuchFileException
java.nio.file.Path
java.nio.file.Paths
java.nio.file.StandardOpenOption
java.nio.file.attribute.PosixFilePermission
java.nio.file.attribute.PosixFilePermissions
java.security.DigestInputStream
java.security.DigestOutputStream
java.security.GeneralSecurityException
java.security.InvalidAlgorithmParameterException
java.security.InvalidKeyException
java.security.InvalidParameterException
java.security.Key
java.security.KeyFactory
java.security.KeyManagementException
java.security.KeyPair
java.security.KeyPairGenerator

java.security.KeyStore
java.security.KeyStoreException
java.security.MessageDigest
java.security.NoSuchAlgorithmException
java.security.NoSuchProviderException
java.security.Principal
java.security.PrivateKey
java.security.Provider
java.security.PublicKey
java.security.SecureRandom
java.security.Security
java.security.SignatureException
java.security.UnrecoverableEntryException
java.security.UnrecoverableKeyException
java.security.cert.CertPathBuilder
java.security.cert.Certificate
java.security.cert.CertificateEncodingException
java.security.cert.CertificateException
java.security.cert.CertificateExpiredException
java.security.cert.CertificateFactory
java.security.cert.CertificateNotYetValidException
java.security.cert.CertificateParsingException
java.security.cert.PKIXBuilderParameters
java.security.cert.PKIXRevocationChecker
java.security.cert.X509CertSelector
java.security.cert.X509Certificate
java.security.interfaces.ECPrivateKey
java.security.interfaces.ECPublicKey
java.security.interfaces.RSAPrivateCrtKey
java.security.interfaces.RSAPrivateKey
java.security.interfaces.RSAPublicKey
java.security.spec.ECGenParameterSpec
java.security.spec.ECParameterSpec
java.security.spec.InvalidKeySpecException
java.security.spec.PKCS8EncodedKeySpec
java.security.spec.RSAKeyGenParameterSpec
java.sql.Connection
java.sql.DatabaseMetaData
java.sql.Date
java.sql.DriverManager
java.sql.PreparedStatement
java.sql.ResultSet
java.sql.SQLException
java.sql.Timestamp
java.sql.Types
java.text.DecimalFormat
java.text.Normalizer
java.text.NumberFormat

java.text.ParseException
java.text.SimpleDateFormat
java.time.Duration
java.time.Instant
java.time.LocalDateTime
java.time.ZoneId
java.time.ZoneOffset
java.time.ZonedDateTime
java.time.format.DateTimeFormatter
java.time.temporal.ChronoUnit
java.util.AbstractList
java.util.AbstractMap
java.util.ArrayList
java.util.Arrays
java.util.Base64
java.util.Calendar
java.util.Collection
java.util.Collections
java.util.Comparator
java.util.Date
java.util.EnumSet
java.util.Enumeration
java.util.Formatter
java.util.HashMap
java.util.HashSet
java.util.Iterator
java.util.LinkedHashMap
java.util.LinkedHashSet
java.util.LinkedList
java.util.List
java.util.Locale
java.util.Map
java.util.NoSuchElementException
java.util.Objects
java.util.Optional
java.util.OptionalInt
java.util.PrimitiveIterator
java.util.Properties
java.util.Random
java.util.ResourceBundle
java.util.ServiceLoader
java.util.Set
java.util.Spliterator
java.util.Spliterators
java.util.Stack
java.util.TimeZone
java.util.TimerTask
java.util.TreeMap

java.util.TreeSet
java.util.UUID
java.util.concurrent.ArrayBlockingQueue
java.util.concurrent.BlockingQueue
java.util.concurrent.Callable
java.util.concurrent.CancellationException
java.util.concurrent.ConcurrentHashMap
java.util.concurrent.ConcurrentLinkedQueue
java.util.concurrent.ConcurrentMap
java.util.concurrent.CountDownLatch
java.util.concurrent.ExecutionException
java.util.concurrent.ExecutorService
java.util.concurrent.Executors
java.util.concurrent.Future
java.util.concurrent.LinkedBlockingQueue
java.util.concurrent.ScheduledExecutorService
java.util.concurrent.ScheduledFuture
java.util.concurrent.ThreadFactory
java.util.concurrent.ThreadLocalRandom
java.util.concurrent.TimeUnit
java.util.concurrent.TimeoutException
java.util.concurrent.atomic.AtomicBoolean
java.util.concurrent.atomic.AtomicInteger
java.util.concurrent.atomic.AtomicLong
java.util.concurrent.atomic.AtomicReference
java.util.concurrent.locks.ReadWriteLock
java.util.concurrent.locks.ReentrantLock
java.util.concurrent.locks.ReentrantReadWriteLock
java.util.function.BiConsumer
java.util.function.BiFunction
java.util.function.BiPredicate
java.util.function.Consumer
java.util.function.DoubleConsumer
java.util.function.Function
java.util.function.Predicate
java.util.function.Supplier
java.util.logging.Level
java.util.logging.LogManager
java.util.logging.LogRecord
java.util.logging.Logger
java.util.regex.Matcher
java.util.regex.Pattern
java.util.stream.Collectors
java.util.stream.IntStream
java.util.stream.Stream
java.util.stream.StreamSupport
java.util.zip.GZIPInputStream
java.util.zip.GZIPOutputStream

javax.annotation.Generated
javax.annotation.processing.AbstractProcessor
javax.annotation.processing.RoundEnvironment
javax.annotation.processing.SupportedAnnotationTypes
javax.annotation.processing.SupportedSourceVersion
javax.crypto.BadPaddingException
javax.crypto.Cipher
javax.crypto.IllegalBlockSizeException
javax.crypto.KeyGenerator
javax.crypto.Mac
javax.crypto.NoSuchPaddingException
javax.crypto.SecretKey
javax.crypto.SecretKeyFactory
javax.crypto.spec.IvParameterSpec
javax.crypto.spec.PBEKeySpec
javax.crypto.spec.SecretKeySpec
javax.inject.Inject
javax.inject.Provider
javax.inject.Singleton
javax.jms.ConnectionFactory
javax.lang.model.SourceVersion
javax.lang.model.element.Element
javax.lang.model.element.ElementKind
javax.lang.model.element.TypeElement
javax.naming.AuthenticationNotSupportedException
javax.naming.CommunicationException
javax.naming.NamingEnumeration
javax.naming.NamingException
javax.naming.directory.Attribute
javax.naming.directory.SearchControls
javax.naming.directory.SearchResult
javax.naming.ldap.LdapContext
javax.net.SocketFactory
javax.net.ssl.CertPathTrustManagerParameters
javax.net.ssl.KeyManager
javax.net.ssl.KeyManagerFactory
javax.net.ssl.SSLContext
javax.net.ssl.SSLEngine
javax.net.ssl.SSLException
javax.net.ssl.SSLPeerUnverifiedException
javax.net.ssl.SSLServerSocket
javax.net.ssl.SSLSession
javax.net.ssl.SSLSessionContext
javax.net.ssl.SSLSocket
javax.net.ssl.SSLSocketFactory
javax.net.ssl.TrustManager
javax.net.ssl.TrustManagerFactory
javax.net.ssl.X509TrustManager

javax.persistence.AttributeConverter
javax.persistence.CascadeType
javax.persistence.Column
javax.persistence.Converter
javax.persistence.DiscriminatorColumn
javax.persistence.DiscriminatorValue
javax.persistence.ElementCollection
javax.persistence.Embeddable
javax.persistence.Embedded
javax.persistence.EmbeddedId
javax.persistence.Entity
javax.persistence.EnumType
javax.persistence.Enumerated
javax.persistence.FetchType
javax.persistence.GeneratedValue
javax.persistence.Id
javax.persistence.Index
javax.persistence.Inheritance
javax.persistence.InheritanceType
javax.persistence.JoinColumn
javax.persistence.JoinTable
javax.persistence.Lob
javax.persistence.ManyToMany
javax.persistence.ManyToOne
javax.persistence.OneToMany
javax.persistence.OneToOne
javax.persistence.OrderBy
javax.persistence.OrderColumn
javax.persistence.Query
javax.persistence.Table
javax.persistence.Transient
javax.persistence.TypedQuery
javax.persistence.criteria.CriteriaBuilder
javax.persistence.criteria.CriteriaQuery
javax.persistence.criteria.Expression
javax.persistence.criteria.Predicate
javax.persistence.criteria.Root
javax.persistence.criteria.Selection
javax.security.auth.login.Configuration
javax.security.auth.x500.X500Principal
javax.servlet.ServletConfig
javax.servlet.ServletContext
javax.servlet.http.HttpServletRequest
javax.sql.DataSource
javax.tools.Diagnostic
javax.tools.FileObject
javax.tools.StandardLocation
javax.validation.ValidationException

javax.validation.constraints.Min
javax.validation.constraints.NotNull
javax.validation.constraints.Size
javax.ws.rs.Consumes
javax.ws.rs.DELETE
javax.ws.rs.DefaultValue
javax.ws.rs.ForbiddenException
javax.ws.rs.GET
javax.ws.rs.HeaderParam
javax.ws.rs.InternalServerErrorException
javax.ws.rs.NotAllowedException
javax.ws.rs.NotFoundException
javax.ws.rs.POST
javax.ws.rs.PUT
javax.ws.rs.Path
javax.ws.rs.PathParam
javax.ws.rs.ProcessingException
javax.ws.rs.Produces
javax.ws.rs.QueryParam
javax.ws.rs.WebApplicationException
javax.ws.rs.client.Client
javax.ws.rs.client.WebTarget
javax.ws.rs.container.ContainerRequestContext
javax.ws.rs.container.ContainerRequestFilter
javax.ws.rs.container.ContainerResponseContext
javax.ws.rs.container.ContainerResponseFilter
javax.ws.rs.container.PreMatching
javax.ws.rs.core.Application
javax.ws.rs.core.Context
javax.ws.rs.core.Cookie
javax.ws.rs.core.Feature
javax.ws.rs.core.FeatureContext
javax.ws.rs.core.HttpHeaders
javax.ws.rs.core.MediaType
javax.ws.rs.core.MultivaluedMap
javax.ws.rs.core.NewCookie
javax.ws.rs.core.Request
javax.ws.rs.core.Response
javax.ws.rs.core.Response.Status
javax.ws.rs.core.SecurityContext
javax.ws.rs.core.StreamingOutput
javax.ws.rs.core.UriBuilder
javax.ws.rs.core.UriInfo
javax.ws.rs.ext.ExceptionMapper
javax.ws.rs.ext.MessageBodyReader
javax.ws.rs.ext.MessageBodyWriter
javax.ws.rs.ext.ParamConverter
javax.ws.rs.ext.ParamConverterProvider

javax.ws.rs.ext.Provider
javax.ws.rs.ext.Providers
javax.ws.rs.ext.ReaderInterceptor
javax.ws.rs.ext.ReaderInterceptorContext
javax.xml.XMLConstants
javax.xml.bind.DatatypeConverter
javax.xml.parsers.DocumentBuilder
javax.xml.parsers.DocumentBuilderFactory
javax.xml.parsers.ParserConfigurationException
javax.xml.transform.OutputKeys
javax.xml.transform.Source
javax.xml.transform.Transformer
javax.xml.transform.TransformerException
javax.xml.transform.TransformerFactory
javax.xml.transform.dom.DOMSource
javax.xml.transform.stream.StreamResult
javax.xml.transform.stream.StreamSource
javax.xml.validation.Schema
javax.xml.validation.SchemaFactory
javax.xml.validation.Validator
sun.security.provider.certpath.BuildStep
sun.security.provider.certpath.SunCertPathBuilderException

Appendix B

The following parameters are required to be configured to put the TOE in the evaluated configuration (FMT_MEC_EXT.1).

`enveil.security.tls.keystore.path`
Path to the key store on ZeroReveal Server's local disk.

`enveil.security.tls.keystore.type`
Type of the key store (possible options are jks, pkcs12, or bcfks).

`enveil.security.tls.keystore.password`
The key store's password.

`enveil.security.tls.truststore.path`
Path to the trust store on ZeroReveal Server's local disk.

`enveil.security.tls.keystore.type`
Type of the key store (possible options are jks or bcfks).

`enveil.security.tls.truststore.password`
The trust store's password.

If the certificate keys are generated using Elliptic Curve Cryptography, ensure that the curve used is either secp256r1 or secp384r1. If RSA keys are used, they must be 2048, 3072 bits, or 4096 bits.

Ensure that all TLS key stores and TLS trust stores are stored in `/etc/enveil/zeroreveal-client/certs/` and are readable only by the enveil user.

`enveil.common.niap.enforce`
(boolean) Enforces that the server is configured to meet the NIAP requirements.
Must be set to `true`.

`enveil.client.auth.mechanisms`
Comma-separated list of authentication mechanisms to use.
Must be set to `[certificate]`.

`enveil.client.auth.require.user.cert`
(boolean) Whether to require users to present valid TLS client certificates.
Must be set to `true`.

`enveil.client.auth.certificate.user.source.mechanisms`
(string) A comma-separated list of user stores for use with certificate authentication.
Must be set to `[ldap]` .

`enveil.client.auth.certificate.ldap.ssl.enabled`
(boolean) Whether to connect to the LDAP server under TLS for certificate `enveil.client.auth`.
Must be set to `true`.

`enveil.client.auth.certificate.ldap.connect.with.sasl.external`
(boolean) Whether to authenticate to the LDAP server using a TLS client certificate or not for certificate auth.
Must be set to `true`.

`enveil.client.gateway.specification.dir`
(path) Path to a directory containing specifications for ZeroReveal Servers to connect to. Each ZeroReveal Server is represented by a separate properties file.
Must be set to `/etc/enveil/zeroreveal-client/gateways`.

`enveil.security.tls.conscrypt.aes.enabled`

(boolean) true enables the use of native AES ciphers from a bundled BoringSSL implementation. false will disable the native ciphers and use default Java implementation.
Must be set to `false`.

`enveil.security.tls.keystore.check`

(boolean) Validates the key store on startup.
Must be set to `true`.

`enveil.security.tls.strict`

(string) If true, requires TLSv1.2 and an AES-256 cipher suite for all connections. If false, accepts any valid TLS protocol and cipher suite available in the local Java installation.
Must be set to `true`.

`enveil.security.tls.client.certificate.check`

(boolean) Whether to check the validity of a certificate presented by any TLS client (currently only ZeroReveal Client).
Must be set to `true`.

`enveil.security.random.blockingDevice`

(boolean) Whether to use a blocking device for random number generation. That is, wait for enough entropy to be available before generating random numbers.
Must be set to `true`.

`enveil.security.tls.niap.signature.algorithms`

(boolean) Only used NIAP-approved signature algorithms.
Must be set to `true`.

`enveil.security.cert.revocation.check.mode`

(string) Whether to check for certificate revocation using any provided CRL endpoint. Defaults to NONE.
Must be set to “HARD_FAIL”.

ZeroReveal Client automatically restricts all TLS connections to TLS version 1.2, denying all other TLS versions. No further configuration is required to configure the cryptographic engine beyond the parameters described above.