Enveil ZeroReveal™ Compute Fabric Security Target

Version 1.0
August 13, 2018

Prepared for:
Enveil

8171 Maple Lawn Blvd, Suite 240
Fulton, MD 20759

Prepared by:
Leidos Inc.

https://www.leidos.com/CC-FIPS140
Common Criteria Testing Laboratory
6841 Benjamin Franklin Drive
Columbia, MD 21046

1. SECURITY TARGET INTRODUCTION 3

1.1 SECURITY TARGET, TOE AND CC IDENTIFICATION 3
1.2 CONFORMANCE CLAIMS 4
1.3 CONVENTIONS 4

2. TOE DESCRIPTION 8

2.1 TOE OVERVIEW 8
2.2 TOE ARCHITECTURE 9
2.3 TOE DOCUMENTATION 11

3. SECURITY PROBLEM DEFINITION 12

4. SECURITY OBJECTIVES 13

4.1 SECURITY OBJECTIVES FOR THE OPERATIONAL ENVIRONMENT 13

5. IT SECURITY REQUIREMENTS 14

5.1 EXTENDED REQUIREMENTS 14
5.2 TOE SECURITY FUNCTIONAL REQUIREMENTS 14
5.3 TOE SECURITY ASSURANCE REQUIREMENTS 20

6. TOE SUMMARY SPECIFICATION 22

6.1 CRYPTOGRAPHIC SUPPORT 22
6.2 USER DATA PROTECTION 24
6.3 IDENTIFICATION AND AUTHENTICATION 24
6.4 SECURITY MANAGEMENT 25
6.5 PRIVACY 27
6.6 PROTECTION OF THE TSF 28
6.7 TRUSTED PATH/CHANNELS 29

7. PROTECTION PROFILE CLAIMS 30

8. RATIONALE 31

8.1 TOE SUMMARY SPECIFICATION RATIONALE 31

9. APPENDIX: LINUX APIS 33

10. APPENDIX: DEPENDENCIES 45

LIST OF TABLES

Table 1 – Evaluation Specific Terminology 5
Table 2 - TOE Security Functional Components 15
Table 3 - Assurance Components 20
Table 4 - SFR Protection Profile Sources 30
Table 5 – Security Functions vs. Requirements Mapping 31

1. Security Target Introduction

This section identifies the Target of Evaluation (TOE) along with identification of the Security Target (ST) itself. The section includes documentation organization, ST conformance claims, and ST conventions.

The TOE is the Enveil ZeroReveal™ Compute Fabric provided by Enveil.

The Enveil ZeroReveal™ Compute Fabric enables data to remain encrypted even while being processed, thereby eliminating the risk of exposure. For example, a search against a dataset can be processed while keeping both the search and the contents of the dataset encrypted at all times. In addition to securing operations over encrypted data, the product also secures operations over unencrypted data. In this manner, the product encrypts operations such as searches or analytics, and processes these encrypted operations over unencrypted data (without ever decrypting the operation), and produces encrypted results. Thus, a user is able to secure operations in untrusted environments such as data aggregators and data lakes in which they do not control the data or its encryption. The ZeroReveal Client and ZeroReveal Server are evaluated as a software application only and the homomorphic encryption techniques used for the ZeroReveal Client and ZeroReveal Server operations are outside the scope of [PP APP SW].

Enveil's ZeroReveal™ Compute Fabric application consists of one ZeroReveal Client Component and one ZeroReveal Server Component. The ZeroReveal Client Component resides within the enterprise and is responsible for encrypting ZeroReveal Compute Fabric operations and decrypting results. The ZeroReveal Server Component resides within the environment of a data repository and is responsible for processing encrypted operations over the data. The ZeroReveal Compute Fabric encrypted data operations and the decrypting of the results are outside the scope of the [PP APP SW] and therefore not included in the evaluation.

The Security Target contains the following additional sections:

- Security Target Introduction (Section 1)
- TOE Description (Section 2)
- Security Problem Definition (Section 3)
- Security Objectives (Section 4)
- IT Security Requirements (Section 5)
- TOE Summary Specification (Section 6)
- Protection Profile Claims (Section 7)
- Rationale (Section 8).
- Appendix: Linux APIs (Section 9).
- Appendix: Dependencies (Section 10)

1.1 Security Target, TOE and CC Identification

ST Title – Enveil ZeroReveal™ Compute Fabric Security Target

ST Version – Version 1.0

ST Date – August 13, 2018

TOE Identification – Enveil ZeroReveal™ Compute Fabric v1.1.1

ZeroReveal Client v1.1.1

ZeroReveal Server v1.1.1

The evaluated configuration includes one ZeroReveal Client Component and one ZeroReveal Server Component in standalone mode.

TOE Developer – Enveil

Evaluation Sponsor – Enveil

CC Identification – *Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 4, September 2012*

1.2 Conformance Claims

This TOE is conformant to the following CC specifications:

- This ST is conformant to the *Protection Profile for Application Software*, Version 1.2, 22 April 2016 [PP APP SW].

  The following NIAP Technical Decisions apply to evaluation assurance activities.

  - TD0327: Default file permissions for FMT_CFG_EXT.1.2
  - TD0326: RSA-based key establishment schemes
  - TD0305: Handling of TLS connections with and without mutual authentication
  - TD0304: Update to FCS_TLSC_EXT.1.2
  - TD0300: Sensitive Data in FDP_DAR_EXT.1
  - TD0296: Update to FCS_HTTPS_EXT.1.3
  - TD0283: Cipher Suites for TLS in SWApp v1.2
  - TD0268 - FMT_MEC_EXT.1 Clarification
  - TD0267 - TLSS testing - Empty Certificate Authorities list
  - TD0244 - FCS_TLSC_EXT - TLS Client Curves Allowed
  - TD0241- Removal of Test 4.1 in FCS_TLSS_EXT.1.1
  - TD0238 – User-modifiable files FTP_AEX_EXT.1.4
  - TD0221: FMT_SMF.1.1 - Assignments moved to Selections
  - TD0217 – Compliance to RFC5759 and RFC5280 for using CRLs
  - TD0215: Update to FCS_HTTPS_EXT.1.2
  - TD0192 – Update to FCS_STO_EXT.1 Application Note
  - TD0163 – Update to FCS_TLSC_EXT.1.1 Test 5.4 and FCS_TLSS_EXT.1.1 Test
  - TD0131 – Update to FCS_TLSS_EXT.1.1 Test 4.5
  - TD0121: FMT_MEC_EXT.1.1 Configuration Options
  - TD0119: FCS_STO_EXT.1.1 in PP_APP_v1.2
- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 4, September 2012.
  - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1 Revision 4, September 2012.
  - Part 3 Extended

1.3 Conventions

The following conventions have been applied in this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
  - Iteration: allows a component to be used more than once with varying operations. In the ST, iteration is indicated by a number in parentheses placed at the end of the component. For example,

FDP_ACC.1(1) and FDP_ACC.1(2) indicate that the ST includes two iterations of the FDP_ACC.1 requirement, (1) and (2).

- Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]). Note that an assignment within a selection would be identified in italics and with embedded bold brackets (e.g., [***[selected-assignment]***]).
- Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., [***selection***]).
- Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …"). Note that 'cases' that are not applicable in a given SFR have simply been removed without any explicit identification.

- The [PP APP SW] uses an additional convention – the 'case' – which defines parts of an SFR that apply only when corresponding selections are made or some other identified conditions exist. Only the applicable cases are identified in this ST and they are identified using **bold** text.
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

1.3.1 Evaluation Specific Terminology

Table 1 – Evaluation Specific Terminology

Term	Definition
Application	The word application used in Sections 4 and 5 refers to the TOE (e.g. both ZeroReveal Client Component and ZeroReveal Server Component).
arbitrary-precision arithmetic	In computer science, arbitrary-precision arithmetic, also called bignum arithmetic, multiple-precision arithmetic, or sometimes infinite-precision arithmetic, indicates that calculations are performed on numbers whose digits of precision are limited only by the available memory of the host system. This contrasts with the faster fixed-precision arithmetic found in most arithmetic logic unit (ALU) hardware, which typically offers between 8 and 64 bits of precision.
Commodity Hardware	Commodity hardware is a term for affordable devices that are generally compatible with other such devices. In a process called commodity computing or commodity cluster computing, these devices are often networked to provide more processing power when those who own them cannot afford to purchase more elaborate supercomputers, or want to maximize savings in IT design. In many cases, commodity hardware setups involve low-cost desktop computers or workstations that are IBM-compatible and can run operating systems like Microsoft Windows, Linux and DOS without additional software or adaptations.
Computational Platform	Computational platforms are well integrated software stacks capable of performing a useful research calculation
Data Aggregation	Data aggregation is a type of data and information mining process where data is searched, gathered and presented in a report-based, summarized format to achieve specific business objectives or processes and/or conduct human analysis.
Data Lake	A data lake is a storage repository that holds a vast amount of raw data in its native format until it is needed. While a hierarchical

Term	Definition
	data warehouse stores data in files or folders, a data lake uses a flat architecture to store data.
GNOME Keyring	GNOME Keyring is a collection of components in GNOME that store secrets, passwords, keys, certificates and make them available to applications. The data is encrypted and stored in a file in the user's home directory. The default keyring uses the login password for encryption.
Homomorphic Encryption	A form of encryption that allows computation on ciphertexts, generating an encrypted result which, when decrypted, matches the result of the operations as if they had been performed on the plaintext.
JDBC	The Java Database Connectivity (JDBC) API provides universal data access from the Java programming language. Using the JDBC API, you can access virtually any data source, from relational databases to spreadsheets and flat files. JDBC technology also provides a common base on which tools and alternate interfaces can be built.
Keystore	A key store is an encrypted file containing certificates and, optionally, corresponding private keys. Key stores require a password in order to decrypt and read the contents. Enveil components support key stores in two formats: PKCS12 (.p12) and Java Key Store (.jks).
TLS Key Store	A TLS key store is a key store containing the certificate and private key the application will use when communicating over TLS. If the application is a TLS server, the certificate should be a TLS server certificate; if it is a TLS client, the certificate should be a TLS client certificate.
TLS Trust Store	A TLS trust store is a key store containing the CA certificates trusted by the application. When a TLS connection is initiated, the application will reject its peer's certificate if it has not been signed by one of the CA certificates in the trust store.
Truststore	The purpose of a truststore is to verify credentials used in the TLS handshake. A TrustStore stores certificates from a third party or certificates signed by a CA (certificate authorities like Verisign, Thawte, Geotrust or GoDaddy) which can be used to identify a third party.
REST	A RESTful API is an application program interface (API) that uses HTTP requests to GET, PUT, POST and DELETE data.

1.3.2 Abbreviations and Acronyms

This section identifies abbreviations and acronyms used in this ST.

API	Application programming interface
ASLR	Address space layout randomization
CC	Common Criteria
CLI	Command line interface
GMP	GNU Multiple Precision Library

GPG	GNU Privacy Guard
GUI	Graphical user interface
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP Secure
IP	Internet Protocol
JAR	Java Archive
JDBC	Java Database Connectivity
JRE	Java Runtime Environment
LDAP	Lightweight Directory Access Protocol
NIAP	National Information Assurance Partnership
PII	Personally Identifiable Information
REST	Representational state transfer
RPM	RPM Package Manager
SAR	Security assurance requirement
SFR	Security functional requirement
TLS	Transport Layer Security

2. TOE Description

This section describes the Enveil ZeroReveal™ Compute Fabric, which is the Target of Evaluation (TOE). The description covers TOE architecture, logical boundaries, and physical boundaries.

2.1 TOE Overview

The ZeroReveal Client and ZeroReveal Server communicate via REST over mutually authenticated TLS. The REST APIs may be used over HTTPS/TLS, and require no additional software to be installed on the system communicating with the Client. Note, neither the ZeroReveal Client nor the ZeroReveal Server Components provide a graphical user interface (GUI) or a command line interface (CLI). Users and applications communicate to the ZeroReveal Client through the REST APIs to retrieve data.[1] An administrator communicates with and manages the ZeroReveal Client Component and ZeroReveal Server Component via configuration files (modifiable by a text editor) on each platform.

Figure 1 shows the ZeroReveal Client and ZeroReveal Server in its environment.

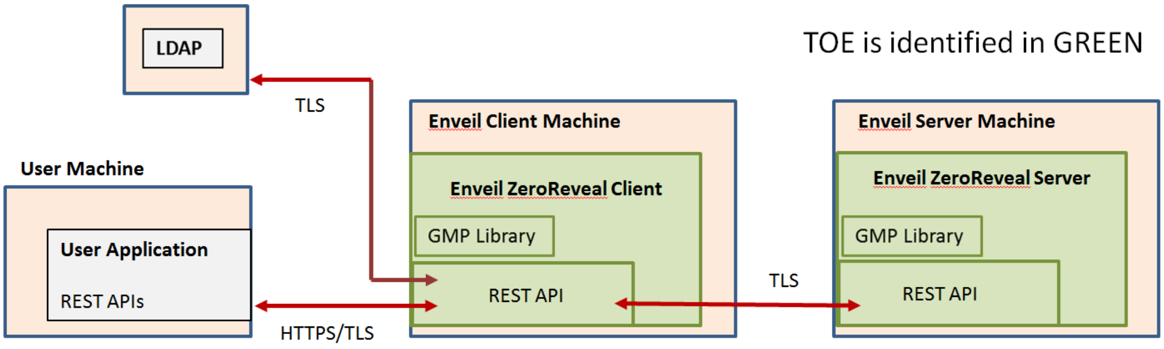

Figure 1 – Simplified Network Connection

The ZeroReveal Client and ZeroReveal Server Components store passwords in a GNOME keyring in non-volatile memory. Both components require their passwords to decrypt keystores and truststores containing TLS certificate and private keys to be stored by an administrator in a GNOME keyring. The ZeroReveal Client Component also requires its LDAP user password to be in stored in the GNOME keyring. Both components must be launched manually by an administrator, who must enter the password to unlock the keyring for the ZeroReveal Client and ZeroReveal Server components. Keyring passwords must be stored with an "enveil_handle" attribute, which the Client and Server Components use to locate the password in the keyring.

The ZeroReveal Client Component does not provide any functionality until an administrator provides configuration files. The configuration files contain the ZeroReveal Client LDAP username and LDAP “bind” credentials, paths to PKI certificates and paths to private keys on the file system and the value of the "enveil_handle" attribute for locating passwords in the keyring. The components store configuration files in the /etc and home directories.

Users and applications authenticate to a ZeroReveal Client Component in order to request ZeroReveal Compute Fabric operations. Enveil uses an external LDAP directory to authenticate users and applications. The LDAP credentials can consist of a username and password, or an X509 certificate trusted by the LDAP root authority.

The ZeroReveal Client Component relays information presented by a user or application to the LDAP directory and the directory provides the client with decision to either accept or deny, which the client enforces. The ZeroReveal Client and ZeroReveal Server components rely on certificate validation functions provided by the platform. The ZeroReveal Server has an explicit list of client certificates that are allowed to connect to it, and what their permissions are, and will reject any connections not using those certificates.

The ZeroReveal Client and ZeroReveal Server components only support OCSP and CRL checking if the peer certificate includes endpoint (for example, CRL Distribution Points field). The ZeroReveal Client and ZeroReveal

[1] Enveil provides for convenience a JDBC Driver library that can be used to communicate with ZeroReveal Client instead of using the Client’s REST API directly. It is not part of the TOE.

Server do not permit OCSP stapling. The ZeroReveal Client and ZeroReveal Server components treat a certificate as a CA certificate only if the basicConstraints extension is present and the CA flag is set to TRUE. The ZeroReveal Server Component does not provide any functionality until an administrator provides configuration files and verifies the certificate is valid.

The ZeroReveal Client acts as a TLS client when establishing connection to LDAP directory for authentication and when establishing connection to ZeroReveal Server Component for operation requests and responses. When acting as a TLS client, the ZeroReveal Client supports mutual authentication using X.509v3 certificates. The ZeroReveal Client Component requires HTTPS/TLS for connections to the REST interface. The ZeroReveal Client Component and ZeroReveal Server Component communicate via REST over mutually authenticated TLS.

The ZeroReveal Client Component acts as a TLS server when accepting user connections at its REST interfaces. For user connections to ZeroReveal Client Component, the client relies on the LDAP directory for the expected identifier. The ZeroReveal Client verifies that the presented identifier matches the reference identifier according to RFC 6125 and only establish a trusted channel if the peer certificate is valid. The ZeroReveal Client Component supports wildcards.

The ZeroReveal Server Component acts as a TLS server. For ZeroReveal Client Component connections to the ZeroReveal Server Component, the server validates the client’s X.509v3 certificate, extracts the client public key, maps client public key to user names in a configuration file, and compares the identified user name to DN or SAN. The ZeroReveal Server will not accept an invalid certificate. The ZeroReveal Server Component also supports wildcards.

The ZeroReveal Client and the ZeroReveal Server components invoke the platform-provided TLS 1.2 from the SunJCE cryptographic provider and supports the following ciphersuites:

- 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

ZeroReveal Client and ZeroReveal Server components do not check for or apply updates on their own, they rely on the operating system's package manager to poll the Enveil repository for updates, and to notify the user when updates are available. An enterprise may mirror the Enveil repository to the client platform when operating in an environment without network connectivity.

2.2 TOE Architecture

This section describes the TOE architecture including physical and logical boundaries. Figure shows the relationship of the ZeroReveal Compute Fabric Component to its operational environment along with the TOE boundary. The security functional requirements identify the libraries included in the ZeroReveal Client and ZeroReveal Server components.

The ZeroReveal Client and ZeroReveal Server rely on the Java Cryptographic Architecture for cryptographic services and on the GNU Multiple Precision Arithmetic Library (GMP) for arbitrary-precision arithmetic. The ZeroReveal Client and ZeroReveal Server components are packaged with the GMP. The ZeroReveal Client Component and ZeroReveal Server Component rely on Linux tools (for example, Yum) in support of trusted update. The underlying operating system is Linux (CentOS 7.4 with SELinux) and the Java Runtime is the Oracle Java 8 JRE with the Unlimited Strength Jurisdiction Policy installed.

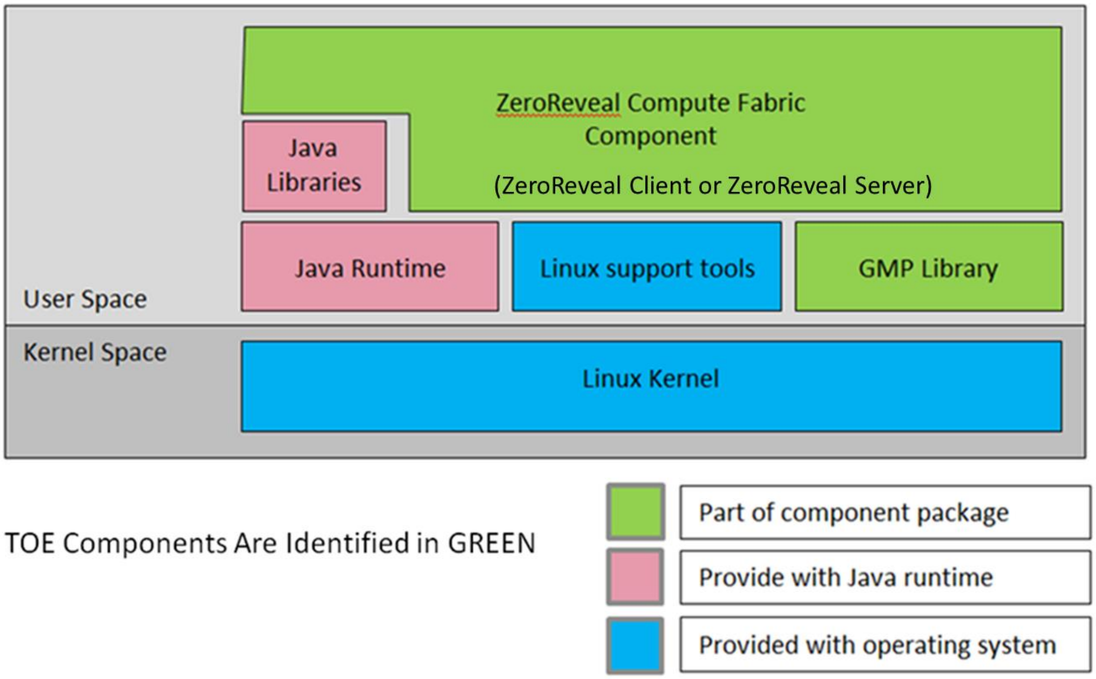
ZeroReveal Compute Fabric Component
(ZeroReveal Client or ZeroReveal Server)
Java Libraries
Java Runtime
Linux support tools
GMP Library
User Space
Kernel Space
Linux Kernel
TOE Components Are Identified in GREEN
Part of component package
Provide with Java runtime
Provided with operating system

Figure 2 - Platform for ZeroReveal Compute Fabric Components

GMP is native platform code. Enveil ensures the GMP library is compiled with anti-exploitation capabilities enabled.

2.2.1 Physical Boundaries

The ZeroReveal Compute Fabric is composed of the ZeroReveal Client Component and the ZeroReveal Server Component. The client's representational state transfer (REST) interfaces are within the scope of evaluation. The configuration files on each platform of the ZeroReveal Client and ZeroReveal Server Components are considered part of the TOE.

2.2.1.1 Software Requirements

The ZeroReveal Client and ZeroReveal Server components run on Linux (CentOS 7.4 with SELinux) and the Java Runtime is the Oracle Java 8 JRE with the Unlimited Strength Jurisdiction Policy installed.

The ZeroReveal Client and ZeroReveal Server rely on the Java Cryptographic Architecture for cryptographic services.

2.2.1.2 Hardware Requirements

The ZeroReveal Client and ZeroReveal Server run on ordinary commodity hardware using general purpose x86_64 CPUs with at least 64 GB of available disk space.

An external LDAP directory is required to authenticate users.

2.2.2 Logical Boundaries

This section summarizes the security functions provided by the TOE:

- Cryptographic support
- User data protection
- Identification and authentication
- Security management

- Privacy
- Protection of the TSF
- Trusted path/channels

2.2.2.1 Cryptographic support

The ZeroReveal Client and ZeroReveal Server components use cryptographic services provided by the platform. Users communicate with the ZeroReveal Client Component through REST interfaces protected by HTTPS/TLS. The ZeroReveal Client Component and ZeroReveal Server Component communicate via REST over mutually authenticated TLS. The ZeroReveal Client Component communicates with a LDAP Server using TLS. The ZeroReveal Client supports mutual authentication of the user connections at its REST interfaces; and with connections to the LDAP directory.

Credentials are stored in platform provided GNOME keyrings.

2.2.2.2 User data protection

The ZeroReveal Client network communication is restricted to user-initiated communication for authentication via LDAP directory, respond to API requests, and initiate communications with the ZeroReveal Server. The ZeroReveal Server network communication is limited to responding to the ZeroReveal Client requests.

All credentials and sensitive data are stored in GNOME keyrings.

2.2.2.3 Identification and authentication

The ZeroReveal Client and ZeroReveal Server components rely on certificate validation functions provided by the platform to authenticate the X.509 certificate as part of establishing a TLS connection.

2.2.2.4 Security management

An enterprise manages the ZeroReveal Client Component and ZeroReveal Server Component via configuration files on each platform. There is no management CLI, GUI, or interface to manage a component over a network.

The ZeroReveal Client and ZeroReveal Server components do not include any predefined or default credentials, and utilize the platform recommended storage process for configuration files.

2.2.2.5 Privacy

The ZeroReveal Client Component and ZeroReveal Server Component do not collect or transmit PII over a network.

2.2.2.6 Protection of the TSF

The ZeroReveal Client and ZeroReveal Server leverage platform provided package management for secure installation and updates. The ZeroReveal Client and ZeroReveal Server package includes only those third-party libraries necessary for its intended operation. The ZeroReveal Client and ZeroReveal Server are designed to utilize compiler provided anti-exploitation capabilities.

2.2.2.7 Trusted path/channels

The ZeroReveal Client and ZeroReveal Server components communicate via mutually authenticated TLS over REST. ZeroReveal Client Components communicate with an authentication server using Lightweight Directory Access Protocol (LDAP) secured with TLS. Users communicate with ZeroReveal Client Component through REST. ZeroReveal Client Component requires HTTPS/TLS for connections to the REST interface.

2.3 TOE Documentation

The TOE includes the following Enveil ZeroReveal™ Compute Fabric documentation.

- ZeroReveal Compute Fabric Configuration Guide for Common Criteria v3.1, Version 1.1.1, 2018
- ZeroReveal Compute Fabric Manual, Version 1.1.1, 2018

3. Security Problem Definition

This security target includes by reference the Security Problem Definition from the [PP APP SW]. The Security Problem Definition consists of threats that a conformant TOE is expected to address and assumptions about the operational environment of the TOE.

In general, the [PP APP SW] has presented a Security Problem Definition appropriate for application software that runs on mobile devices, as well as on desktop and server platforms. The Enveil ZeroReveal™ Compute Fabric TOE is a Linux application running within an enterprise deployment. As such [PP APP SW] Security Problem Definition applies to the TOE.

4. Security Objectives

Like the Security Problem Definition, this security target includes by reference the Security Objectives from the [PP APP SW]. The [PP APP SW] security objectives for the operational environment are reproduced below, since these objectives characterize technical and procedural measures each consumer must implement in their operational environment.

In general, the [PP APP SW] has presented a Security Objectives statement appropriate for application software that runs on mobile devices, as well as on desktop and server platforms. Consequently, the [PP APP SW] security objectives are suitable for the Enveil ZeroReveal™ Compute Fabric TOE.

4.1 Security Objectives for the Operational 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. IT Security Requirements

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

The security functional requirements have all been drawn from: *Protection Profile for Application Software*, Version 1.2, 22 April 2016 [PP APP SW]. As a result, refinements and operations already performed in that PP are not identified (e.g., highlighted) here, rather the requirements have been copied from that PP and any residual operations have been completed herein. Of particular note, [PP APP SW] made a number of refinements and completed some of the SFR operations defined in the CC. [PP APP SW] should be consulted to identify those changes if necessary.

The security assurance requirements are the set of SARs specified in [PP APP SW].

5.1 Extended Requirements

All of the extended requirements in this ST have been drawn from the [PP APP SW]. The [PP APP SW] defines the following extended SFRs. Since these SFRs are not redefined in this ST, readers should consult [PP APP SW] for more information in regard to these CC extensions.

- FCS_RBG_EXT.1 Random Bit Generation Services
- FCS_STO_EXT.1 Storage of Credentials
- FCS_HTTPS_EXT.1 HTTPS Protocol
- FCS_TLSC_EXT.1 TLS Client Protocol
- FCS_TLSC_EXT.2 TLS Client Protocol
- FCS_TLSC_EXT.4 TLS Client Protocol
- FCS_TLSS_EXT.1 TLS Server Protocol
- FDP_DAR_EXT.1 Encryption Of Sensitive Application Data
- FDP_NET_EXT.1(1) Network Communications (ZeroReveal Client Component)
- FDP_NET_EXT.1(2) Network Communications (ZeroReveal Server Component)
- FDP_DEC_EXT.1 Access to Platform Resources
- FIA_X509_EXT.1 X.509 Certificate Validation
- FIA_X509_EXT.2 X.509 Certificate Authentication
- FMT_MEC_EXT.1 Supported Configuration Mechanism
- FMT_CFG_EXT.1 Secure by Default Configuration
- 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 Use of Third Party Libraries
- FPT_TUD_EXT.1 Integrity for Installation and Update
- FTP_DIT_EXT.1 Protection of Data in Transit

5.2 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by the Enveil ZeroReveal™ Compute Fabric TOE.

Table 2 - TOE Security Functional Components

Requirement Class	Requirement Component
FCS: Cryptographic support	FCS_RBG_EXT.1 Random Bit Generation Services
	FCS_STO_EXT.1 Storage of Credentials
	FCS_HTTPS_EXT.1 HTTPS Protocol
	FCS_TLSC_EXT.1 TLS Client Protocol
	FCS_TLSC_EXT.2 TLS Client Protocol
	FCS_TLSC_EXT.4 TLS Client Protocol
	FCS_TLSS_EXT.1 TLS Server Protocol
FDP: User data protection	FDP_DAR_EXT.1 Encryption of Sensitive Application Data
	FDP_DEC_EXT.1 Access to Platform Resources
	FDP_NET_EXT.1(1) Network Communications (ZeroReveal Client Component)
	FDP_NET_EXT.1(2) Network Communications (ZeroReveal Server Component)
FIA: Identification and authentication	FIA_X509_EXT.1 X.509 Certificate
	FIA_X509_EXT.2 X.509 Certificate Authentication
FMT: Security management	FMT_CFG_EXT.1 Secure by Default Configuration
	FMT_MEC_EXT.1 Supported Configuration Mechanism
	FMT_SMF.1 Specification of Management Functions
FPR: Privacy	FPR_ANO_EXT.1 User Consent for Transmission of Personally Identifiable Information
FPT: Protection of the TSF	FPT_API_EXT.1 Use of Supported Services and APIs
	FPT_AEX_EXT.1 Anti-Exploitation Capabilities
	FPT_LIB_EXT.1 Use of Third Party Libraries
	FPT_TUD_EXT.1 Integrity for Installation and Update
FTP: Trusted path/channels	FTP_DIT_EXT.1 Protection of Data in Transit

5.2.1 Cryptographic Support (FCS)

5.2.1.1 Random Bit Generation Services (FCS_RBG_EXT.1)

FCS_RBG_EXT.1.1 The application shall [***use no DRBG functionality***] for its cryptographic operations.

5.2.1.2 Storage of Credentials (FCS_STO_EXT.1)

FCS_STO_EXT.1.1[2] The application shall [***invoke the functionality provided by the platform to securely store [LDAP password, LDAP X509 Certificate, TLS certificates and private keys, and credentials for access to TLS certificates and private keys]***] to non-volatile memory.

5.2.1.3 HTTPS Protocol (FCS_HTTPS_EXT.1)

FCS_HTTPS_EXT.1.1 The ~~application~~ **ZeroReveal Client Component** shall implement the HTTPS protocol that complies with RFC 2818.

[2] Updated per NIAP TD0119

FCS_HTTPS_EXT.1.2[3] The ~~application~~ **ZeroReveal Client Component** shall implement HTTPS using TLS in accordance with [***FCS_TLSC_EXT.1, FCS_TLSS_EXT.1***].

FCS_HTTPS_EXT.1.3[4] The ~~application~~ **ZeroReveal Client Component** shall **[*notify the user and not establish the connection*]** if the peer certificate is deemed invalid.

5.2.1.4 TLS Client Protocol (FCS_TLSC_EXT.1)

FCS_TLSC_EXT.1.1[5] The ~~application~~ **ZeroReveal Client Component** shall [***invoke platform-provided TLS 1.2***] supporting the following ciphersuites:
[
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 ciphersuite.

FCS_TLSC_EXT.1.2 The ~~application~~ **ZeroReveal Client Component** shall verify that the presented identifier matches the reference identifier according to RFC 6125.

FCS_TLSC_EXT.1.3 The ~~application~~ **ZeroReveal Client Component** shall establish a trusted channel only if the peer certificate is valid.

5.2.1.5 TLS Client Protocol (FCS_TLSC_EXT.2)

FCS_TLSC_EXT.2.1 The ~~application~~ **ZeroReveal Client Component** shall support mutual authentication using X.509v3 certificates.

5.2.1.6 TLS Client Protocol (FCS_TLSC_EXT.4)

FCS_TLSC_EXT.4.1[6] The ~~application~~ **ZeroReveal Client Component** shall present the Supported Elliptic Curves Extension in the Client Hello with the following NIST curves: [***secp256r1, secp384r1***]~~ and no other curves~~.

5.2.1.7 TLS Server Protocol (FCS_TLSS_EXT.1)

FCS_TLSS_EXT.1.1[7] The application shall [***invoke platform-provided TLS 1.2***] supporting 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
]
and no other ciphersuite.

FCS_TLSS_EXT.1.2 The application shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0, TLS 1.1, and [***none***].

[3] Updated per NIAP TD0215

[4] Updated per NIAP TD0296

[5] Updated per NIAP TD0283

[6] Updated per NIAP TD0244

[7] Updated per NIAP TD0283

FCS_TLSS_EXT.1.3[8] The application shall generate key establishment parameters using **[*RSA with key size* [*2048 bits, 3072 bits, 4096 bits*], *ECDHE over NIST curves*: [*secp256r1, secp384r1*] *and no other curves*]**.

FCS_TLSS_EXT.1.4 The application shall support mutual authentication of TLS clients using X.509v3 certificates.

FCS_TLSS_EXT.1.5 The application shall not establish a trusted channel if the peer certificate is invalid.

FCS_TLSS_EXT.1.6 The application shall not establish a trusted channel if the distinguished name (DN) or Subject Alternative Name (SAN) contained in a certificate does not match the expected identifier for the peer.

5.2.2 User Data Protection (FDP)

5.2.2.1 Encryption of Sensitive Application Data (FDP_DAR_EXT.1)

FDP_DAR_EXT.1.1[9] The application shall [***protect sensitive data in accordance with FCS_STO_EXT.1***] in nonvolatile memory.

5.2.2.2 Access to Platform Resources (FDP_DEC_EXT.1)

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***].

5.2.2.3 Network Communications (ZeroReveal Client Component) (FDP_NET_EXT.1(1))

FDP_NET_EXT.1.1(1) The ~~application~~ **ZeroReveal Client Component** shall restrict network communication to [

- ***user-initiated communication for [authentication via LDAP directory, initiate communication with server],***
- ***respond to [API requests],***

] .

5.2.2.4 Network Communications (ZeroReveal Server Component) (FDP_NET_EXT.1(2))

FDP_NET_EXT.1.1(2) The ~~application~~ **ZeroReveal Server Component** shall restrict network communication to [

- ***respond to [communication request from client]***

] .

[8] Updated per NIAP TD0326

[9] Updated per NIAP TD0300

5.2.3 Identification and authentication (FIA)

5.2.3.1 X.509 Certificate Validation (FIA_X509_EXT.1)

FIA_X509_EXT.1.1[10] The application [***invoke platform-provided functionality***] to validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation.
- The certificate path must terminate with a trusted CA certificate.
- The application shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates.
- The application shall validate the revocation status of the certificate using [

  the Online Certificate Status Protocol (OCSP) as specified in RFC 2560,
  a Certificate Revocation List (CRL) as specified in RFC 5280 Section 6.3,
  a Certificate Revocation List (CRL) as specified in RFC 5759,

  ] .
- 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 (idkp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
  - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
  - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
  - S/MIME certificates presented for email encryption and signature shall have the Email Protection purpose (id-kp 4 with OID 1.3.6.1.5.5.7.3.4) in the extendedKeyUsage field.
  - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.
  - Server certificates presented for EST shall have the CMC Registration Authority (RA) purpose (id-kp-cmcRA 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.

5.2.3.2 X.509 Certificate Authentication (FIA_X509_EXT.2)

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***].

[10] Updated per NIAP TD0217

5.2.4 Security Management (FMT)

5.2.4.1 Secure by Default Configuration (FMT_CFG_EXT.1)

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

FMT_CFG_EXT.1.2[11] 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.

5.2.4.2 Supported Configuration Mechanism (FMT_MEC_EXT.1)

FMT_MEC_EXT.1.1[12] The TSF shall [***invoke the mechanisms recommended by the platform vendor for storing and setting configuration options***].

5.2.4.3 Specification of Management Functions (FMT_SMF.1)

FMT_SMF.1.1[13] The TSF shall be capable of performing the following management functions [***no other function***].

5.2.5 Privacy

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

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

5.2.6 Protection of the TSF (FPT)

5.2.6.1 Anti-Exploitation Capabilities (FPT_AEX_EXT.1)

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 [Oracle 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 compiled with stack-based buffer overflow protection enabled.

5.2.6.2 Use of Supported Services and APIs (FPT_API_EXT.1)

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

[11] Updated per NIAP TD0327

[12] Updated per NIAP TD0121

[13] Updated per NIAP TD0221

5.2.6.3 Use of Third Party Libraries (FPT_LIB_EXT.1)

FPT_LIB_EXT.1.1 The application shall be packaged with only [**GNU Multiple Precision Arithmetic Library (GMP)**].

5.2.6.4 Integrity for Installation and Update (FPT_TUD_EXT.1)

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 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 [***leverage the platform***] 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.

5.2.7 Trusted path/channels (FTP)

5.2.7.1 Protection of Data in Transit (FTP_DIT_EXT.1)

FTP_DIT_EXT.1.1 The application shall [***encrypt all transmitted data with [HTTPS, TLS]***] between itself and another trusted IT product.

5.3 TOE Security Assurance Requirements

The security assurance requirements in Table 3 are included in this ST by reference from the [PP APP SW].

Table 3 - Assurance Components

Requirement Class	Requirement Component
ADV: Development	ADV_FSP.1 Basic functional specification
AGD: Guidance documents	AGD_OPE.1: Operational user guidance
	AGD_PRE.1: Preparative procedures
ALC: Life-cycle support	ALC_CMC.1 Labelling of the TOE
	ALC_CMS.1 TOE CM coverage
	ALC_TSU_EXT.1 Timely Security Updates
ATE: Tests	ATE_IND.1 Independent testing - conformance
AVA: Vulnerability assessment	AVA_VAN.1 Vulnerability survey

These assurance requirements imply the following requirements from CC class ASE: Security Target Evaluation.

- ASE_CCL.1 Conformance claims
- ASE_ECD.1 Extended components definition
- ASE_INT.1 ST introduction

- ASE_OBJ.1 Security objectives for the operational environment
- ASE_REQ.1 Stated security requirements
- ASE_TSS.1 TOE summary specification

Consequently, the assurance activities specified in [PP APP SW] apply to the TOE evaluation.

6. TOE Summary Specification

This chapter describes the security functions:

- Cryptographic support
- User data protection
- Identification and authentication
- Security management
- Privacy
- Protection of the TSF
- Trusted path/channels

6.1 Cryptographic support

6.1.1 FCS_RBG_EXT.1

The ZeroReveal Client and ZeroReveal Server Components rely on the platform for cryptographic services. Consequently, the ZeroReveal Client and ZeroReveal Server Components themselves use no DRBG functions.

6.1.2 FCS_STO_EXT.1

Users and applications must authenticate to a ZeroReveal Client Component in order to request ZeroReveal Compute Fabric operations. Enveil uses an external LDAP directory to authenticate users. The ZeroReveal Client Component relays information presented by a user to the LDAP directory and the directory provides the client with decision to either accept or deny, which the client enforces.

The ZeroReveal Client component protects its LDAP access credentials using the platform, as follows:

- ZeroReveal Client may authenticate itself to LDAP with a username and password or with a certificate.
    - If authenticating with a username and password, the password is required to be stored in the GNOME keyring in non-volatile memory.
    - If authenticating with a certificate, the certificate and private key must be stored in its TLS Key Store. The password to the TLS Key Store is required to be stored in the GNOME keyring in non-volatile memory.
- ZeroReveal Client may optionally use TLS when communicating with LDAP. In this case, ZeroReveal Client requires a CA certificate to be stored in its TLS Trust Store. The password to the TLS Trust Store is required to be stored in the GNOME keyring in non-volatile memory.

The ZeroReveal Client and ZeroReveal Server Components store certificates and associated PKI private keys for communication over TLS in their TLS Key Store and TLS Trust Store, which are encrypted files protected with a password. The ZeroReveal Client and ZeroReveal Server Components require the passwords to their TLS Key Store and TLS Trust Store to be stored in the GNOME keyring in non-volatile memory.

Both components must be launched manually by an administrator, who must enter the password to unlock the keyring for the application.

For Common Criteria compliance, passwords may not be stored in any of the plain-text configuration files used by ZeroReveal Client or ZeroReveal Server. Instead, they must be stored in the Enveil user's default GNOME keyring (also called the "login" keyring), with attributes that allow ZeroReveal Client and ZeroReveal Server to look them up. The GNOME keyring stores the sensitive data in encrypted format in a keyring file in the user's home directory.

To load passwords into the keyring, use the secret-tool command (which comes with the libsecret library, usually installed with GNOME). For the password to be loaded by ZeroReveal Client or ZeroReveal Server, it must have two attributes: enveil_type, which must be set to password, and enveil_handle, which must be set to a unique handle that will be stored in ZeroReveal Platform configuration files. Stored passwords must also have a label, which will be displayed in the keyring UI. Key stores require a password in order to decrypt and read the contents.

Both ZeroReveal Client and ZeroReveal Server Components must be launched manually by an administrator, who must enter the password to unlock the keyring. The Client and Server components then look up passwords in the keyring by their "enveil handle" attribute, which is stored in the component's configuration file.

6.1.3 FCS_HTTPS_EXT.1

The ZeroReveal Client Component's HTTPS protocol complies with RFC 2818 and is implemented using TLS 1.2 (RFC 5246). The ZeroReveal Client Component REST interface does not accept a connection when a peer's certificate is invalid and records the failure in the components log.

6.1.4 FCS_TLSC_EXT.1, FCS_TLSC_EXT.2, FCS_TLSC_EXT.4

The ZeroReveal Client acts as a TLS client when establishing connection to LDAP directory for authentication and when establishing connection to ZeroReveal Server Component for operation requests and responses. When acting as a TLS client, the ZeroReveal Client supports mutual authentication using X.509v3 certificates. The ZeroReveal Client certificate must contain the hostname or the IP address of the ZeroReveal Client host machine as a Subject Alternative Name (SAN). The ZeroReveal Client and ZeroReveal Server components verify that the Common Name and DN presented identifiers match the reference identifier according to RFC 6125. Both ZeroReveal Client and ZeroReveal Server Components support wildcards. The ZeroReveal Client does not support certificate pinning. The ZeroReveal Server has an explicit list of client certificates that are allowed to connect to it, and what their permissions are, and will reject any connections not using those certificates.

When acting as a TLS client, the ZeroReveal Client component invokes the platform-provided TLS 1.2 from the SunJCE cryptographic provider and supports the following ciphersuites:

- 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 ZeroReveal Client Component supports Elliptic Curves Extension in the Client Hello with the secp256r1, and secp384r1 NIST curves. The supported curves are hardcoded and there are no configuration options.

The ZeroReveal Server Component does not act as a TLS client.

6.1.5 FCS_TLSS_EXT.1

The ZeroReveal Client Component when accepting user connections (acting as a TLS server) supports mutual authentication of the user connections at its REST interfaces. For user connections to ZeroReveal Client Component, the client relies on the LDAP directory for the expected identifier. The ZeroReveal Client verifies that the presented identifier matches the reference identifier according to RFC 6125 and only establish a trusted channel if the peer certificate is valid. Both ZeroReveal Server and ZeroReveal Client Components support wildcards. The ZeroReveal Client Component and ZeroReveal Server Component communicate via REST over mutually authenticated TLS. The ZeroReveal Server Component acts as a TLS server. For ZeroReveal Client Component connections to the ZeroReveal Server Component, the server validates the client's X.509v3 certificate, extracts the client public key, maps client public key to user names in a configuration file, and compares the identified user name to DN or SAN. The ZeroReveal Server will not establish a trusted channel if the distinguished name (DN) or Subject Alternative Name (SAN) contained in a certificate does not match the expected identifiers for the peer for Client Component connections with the REST interfaces. The user's permissions will be looked up in LDAP using the DN and SANs on their certificate in place of their username. The DN and SANs are extracted from the user's client certificate and used to retrieve the user's entry from the configured LDAP service.

The ZeroReveal Server will not accept an invalid certificate.

When acting as a TLS server, the ZeroReveal Client Component and the ZeroReveal Server Component invoke the platform-provided TLS 1.2 from the SunJCE cryptographic provider and supports the following ciphersuites:

- 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 ZeroReveal Client and ZeroReveal Server generate key establishment parameters using RSA with key size 2048 bits, 3072 bits, or 4096 bits and ECDHE over NIST curves: secp256r1 or secp384r1.

The ZeroReveal Client and ZeroReveal Server Components deny SSL 3.0, TLS 1.0, TLS 1.1 connections.

After initial installation, no additional configuration is required for correct TLS operation of the ZeroReveal Client and ZeroReveal Server components.

6.2 User data protection

6.2.1 FDP_DAR_EXT.1

Sensitive data consists of the LDAP password, LDAP X509 Certificate, TLS certificates and private keys, and credentials for access to TLS certificates and private keys. The sensitive data is protected in accordance with FCS_STO_EXT.1 as described above in Section 6.1.2.

The ZeroReveal Client and ZeroReveal Server Components do not contain any other sensitive data on the system that is not protected by FCS_STO_EXT.1.

6.2.2 FDP_DEC_EXT.1

The ZeroReveal Client and ZeroReveal Server Components access the physical resources for network connectivity. The guidance documentation identifies when ZeroReveal Client Component and ZeroReveal Server Component require network connectivity.

6.2.3 FDP_NET_EXT.1(1) (ZeroReveal Client Component), FDP_NET_EXT.1(2) (ZeroReveal Server Component)

The ZeroReveal Client network communication is restricted to user-initiated communication for authentication via LDAP directory, respond to API requests, and initiate communications with the ZeroReveal Server. The ZeroReveal Server network communication is limited to responding to the ZeroReveal Client requests.

6.3 Identification and authentication

6.3.1 FIA_X509_EXT.1 Certificate Validation

The ZeroReveal Client and ZeroReveal Server components use X.509v3 certificates to authenticate network endpoints for TLS and HTTPS communication. The ZeroReveal Client and ZeroReveal Server components comply with RFC5280 and the certificate path is terminated with a trusted CA certificate. The ZeroReveal Client and ZeroReveal Server components rely on the certificate validation functions provided by the platform.

Certificates used for trusted updates and executable code integrity verification have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field. The JARS are signed by a commercial certificate authority. The ZeroReveal Client and ZeroReveal Server components do not support email, therefore SMIME certificates are not used.

The ZeroReveal Client and ZeroReveal Server components only support OCSP and CRL checking if the peer certificate includes endpoint (for example, CRL Distribution Points field). The ZeroReveal Client and ZeroReveal Server components do not permit OCSP stapling. The ZeroReveal Client and ZeroReveal Server components treat a certificate as a CA certificate only if the basicConstraints extension is present and the CA flag is set to TRUE.

Both the ZeroReveal Client and ZeroReveal Server components use the Java PKIX certificate 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) provided during configuration by the class sun.security.provider.certpath.PKIXCertPathValidator and X509TrustManager.

The ZeroReveal Server checks can be found in sun.security.ssl.X509KeyManagerImpl.CheckType#check http://hg.openjdk.java.net/jdk8/jdk8/jdk/file/687fd7c7986d/src/share/classes/sun/security/ssl/X509KeyManagerImpl.

java#l568. It ensures certificates presented for server authentication have the digitalSignature keyUsage and TLS Server extendedKeyUsage.

The ZeroReveal Client checks can be found in sun.security.ssl.X509KeyManagerImpl.CheckType#check http://hg.openjdk.java.net/jdk8/jdk8/jdk/file/687fd7c7986d/src/share/classes/sun/security/ssl/X509KeyManagerImpl.java#l568. It ensures certificates presented for client authentication have the digitalSignature keyUsage and TLS Client extendedKeyUsage.

Certificates used to sign OCSP responses are checked for the correct OCSP extendedKeyUsage. The OCSP checking is described at: http://hg.openjdk.java.net/jdk8/jdk8/jdk/file/57c26829deb6/src/share/classes/sun/security/provider/certpath/OCSPResponse.java#l502.

OCSP as specified in RFC 2560, CRL checking as specified in RFC 5759, and RFC 5280 Section 6.3 revocation checking will be attempted on certificates that have listed endpoints. 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 list of X509 TrustManagers associated with TLS contexts used to form connections.

6.3.2 FIA_X509_EXT.2

The ZeroReveal Client and ZeroReveal Server use X.509v3 client-side certificates as defined by RFC 5280 to support mutual authentication for HTTPS and TLS. The ZeroReveal Client and ZeroReveal Server rely upon the platform for network communication; it also relies upon the platform for validation of X.509v3 certificates as well as for checking the revocation status of the certificate. The determination whether or not to establish a connection if the revocation status of a certificate cannot be established is configured by an administrator in the configuration file.

An administrator sets certificate selection decision in configuration file.

ZeroReveal Client Communications

- `enveil.security.tls.truststore.path` – identifies the path to the TLS trust store. ZeroReveal. The Server will only accept connections that contain one of these certificates in their certificate chain.

User Communications

- By default, ZeroReveal Client does not require users of its REST API to present a TLS certificate when connecting. To require a TLS client certificate for users, `enveil.client.require.mtls.for.users` must be set to "true" in client.properties. A key store will be created and a copy of the CA certificate used when issuing client certificates will be placed in the directory where ZeroReveal Client can read it. This key store will be used as the ZeroReveal Client's TLS trust store.

ZeroReveal Server Communications

- enveil.security.tls.truststore.path – identifies the path to the TLS trust store for incoming connections to ZeroReveal Client.

ZeroReveal Client LDAP communication

- `enveil.client.auth.ldap.ssl.keystore.path (path)` - identifies the path to the TLS key store for the certificate used for the LDAP connection.

6.4 Security management

6.4.1 FMT_CFG_EXT.1

The ZeroReveal Client and Server Components are not installed with default credentials.

Before running the installer packages, an Enveil user with sudo privileges must be created on the ZeroReveal Client and Server Components. Following installation using the package manager, an administrator must log in as the Enveil user, then create a Password Keyring named "login" for the Enveil user. Its password must be set to be the Enveil user's login password.

Additionally, the ZeroReveal Client and ZeroReveal Server installer packages makes sure all configuration and data directories are configured with appropriate permissions to restrict against modification by unprivileged users.

Passwords may not be stored in any of the plain-text configuration files used by ZeroReveal Client or ZeroReveal Server. Instead, they must be stored in the Enveil user's default GNOME keyring (also called the "login" keyring), with attributes that allow ZeroReveal Client and ZeroReveal Server to look them up. All private credentials (such as key store and LDAP passwords) must be stored in the GNOME keyring for the Enveil user.

Once the ZeroReveal Client has been installed, the following configuration steps must be completed:

- Set up TLS for ZeroReveal Client.
- Configure ZeroReveal Client's LDAP interaction.
- Assign ZeroReveal Client permissions in LDAP.
- Configure at least one ZeroReveal Server connection

ZeroReveal Server is configured so that all communications from ZeroReveal Clients must use mutually-authenticated TLS. In order to support this, the following steps must be completed:

- Configure a Certificate Authority.
- Create a ZeroReveal Client TLS key store.
- Create a ZeroReveal Server certificate and a ZeroReveal Server TLS key store.

The ZeroReveal Client Component does not provide any functionality until an administrator provides configuration files.

The ZeroReveal Server Component does not provide any functionality until an administrator provides configuration files and identifies which certificate is valid (as determined by the criteria/procedure listed in the FIA_X509_EXT.1 section above).

6.4.2 FMT_MEC_EXT.1

Configuration files (modifiable by a text editor) are used to manage ZeroReveal Client Component and ZeroReveal Server Component configuration. Non-functional configuration file templates are put in place by the installer package. The components store configuration files in the /etc and home directories. The values and settings in the configuration files which are relevant to the Security Functional Requirements (SFRs) are identified below:

- Configuration changes to Enveil property/configuration files can only be made by editing the configuration files with a text editor.
- CC Guidance Document Appendix A describes how to store the relevant passwords in the GNOME keyring.
  - To store the TLS key store password for ZeroReveal Client, use the command:

    bash$ secret -tool store --label ="ZeroReveal Client TLS key store password" enveil_handle client_tls_key_store_password enveil_type password

    After entering this command, a secret prompt for the password that will be stored will be presented. The handle for this password is `client_tls_key_store_password`. The following value in client.properties:

    enveil.security.tls.keystore.password=client_tls_key_store_password
  - To store the TLS key store password for ZeroReveal Server, use the command:

    bash$ secret -tool store --label ="ZeroReveal Server TLS key store password" enveil_handle server_tls_key_store_password enveil_type password

After entering this command, a secret prompt for the password that will be stored will be presented. The handle for this password is server_tls_key_store_password. The following value in server.properties:

enveil.security.tls.keystore.password=server_tls_key_store_password

- CC Guidance Document Appendix A identifies password attributes stored in the configuration file
  - **enveil_type**, which must be set to password
  - **enveil_handle,** which must be set to a unique handle that will be stored in ZeroReveal Platform configuration files.
  - Stored passwords must also have a label, which will be displayed in the keyring UI.
- The ZeroReveal Server settings that are mandated are enumerated in CC Guidance Document section 2.4. Make sure that `server.properties` is configured with the following constraints:
  - **enveil.common.niap.enforce** is set to ”true”.
  - **enveil.security.passwords.are.keyring.handles** is set to ”true”.
  - **enveil.security.tls.strict** is set to ”true”.
  - **enveil.server.data.source.dir** is set to ”/etc/enveil/server/dataSources” if using local storage (if using cluster storage, any value is valid).
  - **enveil.server.authFile** is set to ”/etc/enveil/server/authorized_clients.json”.
  - **enveil.security.cert.revocation.check.mode** is set to check for certificate revocation using any provided CRL or OCSP. If NONE, no checks are performed. If SOFT_FAIL, checks do not fail if the CRL/OCSP cannot be reached. If HARD_FAIL, checks fail if the CRL/OCSP cannot be reached.
- ZeroReveal Server Client component settings that are mandated are enumerated in CC Guidance Document section 3.4. Make sure that client.properties is configured with the following constraints:
  - **enveil.common.niap.enforce** is set to ”true”.
  - **enveil.security.passwords.are.keyring.handles** is set to ”true”.
  - **enveil.security.tls.strict** is set to ”true”.
  - **enveil.client.keygen.jce** is set to ”true”.
  - **enveil.client.auth.mechanisms** is set to ”ldap”.
  - **enveil.client.gateway.specification.dir** is set to ”/etc/enveil/client/gateways”.
  - **enveil.client.key.registry.dir** is set to ”/etc/enveil/client/keydir”.
  - **enveil.security.cert.revocation.check.mode** is set to check for certificate revocation using any provided CRL or OCSP. If NONE, no checks are performed. If SOFT_FAIL, checks do not fail if the CRL/OCSP cannot be reached. If HARD_FAIL, checks fail if the CRL/OCSP cannot be reached.

6.4.3 FMT_SMF.1

An enterprise manages the ZeroReveal Client Component and ZeroReveal Server Component via configuration files on each platform. There is no management CLI, GUI, or interface to manage a component.

6.5 Privacy

6.5.1 FPR_ANO_EXT.1

The ZeroReveal Client Component and ZeroReveal Server Component do not collect or transmit PII over a network.

6.6 Protection of the TSF

6.6.1 FPT_AEX_EXT.1

The ZeroReveal Client and ZeroReveal Server components are written in Java which relies on the JRE for memory and stack protection, this means the JRE must be compiled correctly. The ZeroReveal Client and ZeroReveal Server components link to native code GMP which is compiled using GCC with the required compiler flags for ASLR (GCC CFLAG –fPIC, "Generate position-independent code") and stack protection (-fstackprotector-all). The ZeroReveal Client and ZeroReveal Server components rely on the Java Runtime Environment (JRE) for memory protection. The memory protections for the GMP native code portion were verified through static analysis. The ZeroReveal Client and ZeroReveal Server components allocate memory regions with write and execute permissions for Oracle Java runtime performing just-in-time compilation. The ZeroReveal Client and ZeroReveal Server components install 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.

The ZeroReveal Client Component and ZeroReveal Server Component run on a CentOS 7.4 system with SELinux enabled.

6.6.2 FPT_API_EXT.1

Enveil only uses public APIs in the ZeroReveal Client and ZeroReveal Server components. The ZeroReveal Client and ZeroReveal Server components use the Linux APIs listed in section 9 Appendix: Linux APIs.

6.6.3 FPT_LIB_EXT.1

The only dynamically loaded native library that is packaged with the ZeroReveal Client and ZeroReveal Server Components is the GMP library as modified by Enveil. The third-party java dependencies are listed Section 10 Appendix: Dependencies.

6.6.4 FPT_TUD_EXT.1

Enveil will publish Yum repositories for updates and patches to ZeroReveal Client Component and ZeroReveal Server Component. Each component relies on Yum to periodically poll the repositories for updates and notify the user. The Components do not check for or apply updates on their own. Each component 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 a component (for example, if ZeroReveal Client has been installed on a machine without internet access), the enterprise will need to mirror the repositories locally. The ZeroReveal Client and ZeroReveal Server components support 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 ZeroReveal Platform 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 ZeroReveal Client Component and ZeroReveal Server Component each records their version in their log files at startup. An administrator can determine the current version by checking head of the component's log.

The update/install packages include the required information so that the package manager will perform removal and deletion of all traces of the application when an uninstall command is issued through that package manager.

The ZeroReveal Client and ZeroReveal Server components are 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; ZeroReveal Client and ZeroReveal Server components never download, modify, replace or update their 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 ZeroReveal Client and ZeroReveal Server components . 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.

Enveil uses commercial software to automatically check for active CVEs in any third-party dependencies, as part of its software development and release process.

The window between public disclosure of a vulnerability and availability of a security update on the package manager will be 14 - 90 days.

6.7 Trusted path/channels

6.7.1 FTP_DIT_EXT.1

The ZeroReveal Client and ZeroReveal Server components communicate via REST over mutually authenticated TLS. ZeroReveal Client Components communicate with an authentication server using Lightweight Directory Access Protocol (LDAP) secured with TLS. Users communicate with ZeroReveal Client Component through REST interfaces via HTTPS/TLS.

7. Protection Profile Claims

This ST conforms to the *Protection Profile for Application Software,* Version 1.2, 22 April 2016 [PP APP SW].

As explained in Section 3, Security Problem Definition, the Security Problem Definition of the [PP APP SW] has been included by reference into this ST.

As explained in Section 4, Security Objectives, the Security Objectives of the [PP APP SW] have been included by reference into this ST.

The following table identifies all the security functional requirements in this ST. Each SFR is reproduced from the [PP APP SW] and operations completed as appropriate.

Table 4 - SFR Protection Profile Sources

Requirement Class	Requirement Component	Source
FCS: Cryptographic support	FCS_RBG_EXT.1 Random Bit Generation Services	[PP APP SW]
	FCS_STO_EXT.1 Storage of Credentials	[PP APP SW]
	FCS_HTTPS_EXT.1 HTTPS Protocol	[PP APP SW]
	FCS_TLSC_EXT.1 TLS Client Protocol	[PP APP SW]
	FCS_TLSC_EXT.2 TLS Client Protocol	[PP APP SW]
	FCS_TLSC_EXT.4 TLS Client Protocol	[PP APP SW]
	FCS_TLSS_EXT.1 TLS Client Protocol	[PP APP SW]
FDP: User data protection	FDP_DAR_EXT.1 Encryption of Sensitive Application Data	[PP APP SW]
	FDP_DEC_EXT.1 Access to Platform Resources	[PP APP SW]
	FDP_NET_EXT.1(1) Network Communications (ZeroReveal Client Component)	[PP APP SW]
	FDP_NET_EXT.1(2) Network Communications (ZeroReveal Server Component)	[PP APP SW]
FIA: Identification and authentication	FIA_X509_EXT.1 X.509 Certificate Validation	[PP APP SW]
	FIA_X509_EXT.2 X.509 Certificate Authentication	[PP APP SW]
FMT: Security management	FMT_CFG_EXT.1 Secure by Default Configuration	[PP APP SW]
	FMT_MEC_EXT.1 Supported Configuration Mechanism	[PP APP SW]
	FMT_SMF.1 Specification of Management Functions	[PP APP SW]
FPR: Privacy	FPR_ANO_EXT.1 User Consent for Transmission of Personally Identifiable Information	[PP APP SW]
FPT: Protection of the TSF	FPT_AEX_EXT.1 AntiExploitation Capabilities	[PP APP SW]
	FPT_API_EXT.1.1 Use of Supported Services and APIs	[PP APP SW]
	FPT_LIB_EXT.1 Use of Third Party Libraries	[PP APP SW]
	FPT_TUD_EXT.1 Integrity for Installation and Update	[PP APP SW]
FTP: Trusted path/channels	FTP_DIT_EXT.1 Protection of Data in Transit	[PP APP SW]

8. Rationale

This security target includes by reference the [PP APP SW] Security Problem Definition, Security Objectives, and Security Assurance Requirements. The security target makes no additions to the [PP APP SW] assumptions. [PP APP SW security functional requirements have been reproduced with the [PP APP SW] operations completed. Operations on the security requirements follow [PP APP SW] application notes and assurance activities. Consequently, [PP APP SW] rationale applies but is incomplete. The TOE Summary Specification rationale below serves to complete the rationale required for the security target.

8.1 TOE Summary Specification Rationale

Each subsection in Section 6, the TOE Summary Specification, describes a security function of the TOE. Each description is followed with rationale that indicates which requirements are satisfied by aspects of the corresponding security function. The security functions work together to satisfy all of the security functional requirements. Furthermore, all of the security functions are necessary in order for the TSF to provide the required security functionality.

This section in conjunction with Section 6 TOE Summary Specification provides evidence that the security functions are suitable to meet the TOE security requirements. The collection of security functions works together to provide all of the security requirements. The security functions described in the TOE summary specification are all necessary for the required security functionality in the TSF. Table 5 demonstrates the relationship between security requirements and security functions.

Table 5 – Security Functions vs. Requirements Mapping

	Cryptographic support	User data protection	Identification and authentication	Security management	Privacy	Protection of the TSF	Trusted path/channels
FCS_RBG_EXT.1	X						
FCS_STO_EXT.1	X						
FCS_HTTPS_EXT.1	X						
FCS_TLSC_EXT.1	X						
FCS_TLSC_EXT.2	X						
FCS_TLSC_EXT.4	X						
FCS_TLSS_EXT.1	X						
FDP_DAR_EXT.1		X					
FDP_NET_EXT.1(1)		X					
FDP_NET_EXT.1(2)		X					
FDP_DEC_EXT.1		X					
FIA_X509_EXT.1			X				
FIA_X509_EXT.2			X				
FMT_CFG_EXT.1				X			
FMT_MEC_EXT.1				X			
FMT_SMF.1				X			

	Cryptographic support	User data protection	Identification and authentication	Security management	Privacy	Protection of the TSF	Trusted path/channels
FPR_ANO_EXT.1					X		
FPT_AEX_EXT.1						X	
FPT_API_EXT.1						X	
FPT_LIB_EXT.1						X	
FPT_TUD_EXT.1						X	
FTP_DIT_EXT.1							X

9. Appendix: Linux APIs

The Enveil ZeroReveal™ Compute Fabric components use the following APIs from Java:

1. java.io.BufferedInputStream
2. java.io.BufferedReader
3. java.io.BufferedWriter
4. java.io.ByteArrayInputStream
5. java.io.ByteArrayOutputStream
6. java.io.Closeable
7. java.io.DataInput
8. java.io.DataInputStream
9. java.io.DataOutput
10. java.io.DataOutputStream
11. java.io.EOFException
12. java.io.File
13. java.io.FileInputStream
14. java.io.FileNotFoundException
15. java.io.FileOutputStream
16. java.io.FileReader
17. java.io.FileWriter
18. java.io.IOException
19. java.io.InputStream
20. java.io.InputStreamReader
21. java.io.ObjectInputStream
22. java.io.ObjectOutputStream
23. java.io.OutputStream
24. java.io.OutputStreamWriter
25. java.io.PipedInputStream
26. java.io.PipedOutputStream
27. java.io.PrintWriter
28. java.io.Reader
29. java.io.Serializable
30. java.io.StringReader
31. java.io.StringWriter
32. java.io.UnsupportedEncodingException
33. java.io.Writer
34. java.lang.annotation.Annotation
35. java.lang.instrument.UnmodifiableClassException

36. java.lang.management.ManagementFactory
37. java.lang.management.ThreadMXBean
38. java.lang.reflect.Type
39. java.math.BigDecimal
40. java.math.BigInteger
41. java.math.MathContext
42. java.net.InetAddress
43. java.net.JarURLConnection
44. java.net.MalformedURLException
45. java.net.Socket
46. java.net.SocketException
47. java.net.URI
48. java.net.URISyntaxException
49. java.net.URL
50. java.net.URLConnection
51. java.net.URLEncoder
52. java.nio.ByteBuffer
53. java.nio.channels.FileChannel
54. java.nio.charset.Charset
55. java.nio.charset.StandardCharsets
56. java.nio.file.FileSystem
57. java.nio.file.FileSystems
58. java.nio.file.Files
59. java.nio.file.NoSuchFileException
60. java.nio.file.Path
61. java.nio.file.Paths
62. java.nio.file.StandardOpenOption
63. java.nio.file.attribute.PosixFilePermission
64. java.security.DigestInputStream
65. java.security.DigestOutputStream
66. java.security.GeneralSecurityException
67. java.security.InvalidAlgorithmParameterException
68. java.security.InvalidKeyException
69. java.security.Key
70. java.security.KeyManagementException
71. java.security.KeyPair
72. java.security.KeyPairGenerator

73. java.security.KeyStore
74. java.security.KeyStoreException
75. java.security.MessageDigest
76. java.security.NoSuchAlgorithmException
77. java.security.NoSuchProviderException
78. java.security.Principal
79. java.security.PublicKey
80. java.security.SecureRandom
81. java.security.Security
82. java.security.SignatureException
83. java.security.UnrecoverableKeyException
84. java.security.cert.CertPathBuilder
85. java.security.cert.Certificate
86. java.security.cert.CertificateEncodingException
87. java.security.cert.CertificateException
88. java.security.cert.CertificateExpiredException
89. java.security.cert.CertificateFactory
90. java.security.cert.CertificateNotYetValidException
91. java.security.cert.CertificateParsingException
92. java.security.cert.PKIXBuilderParameters
93. java.security.cert.PKIXRevocationChecker
94. java.security.cert.X509CertSelector
95. java.security.cert.X509Certificate
96. java.security.interfaces.RSAPrivateCrtKey
97. java.security.spec.InvalidKeySpecException
98. java.sql.Array
99. java.sql.Blob
100. java.sql.CallableStatement
101. java.sql.ClientInfoStatus
102. java.sql.Clob
103. java.sql.Connection
104. java.sql.DatabaseMetaData
105. java.sql.Date
106. java.sql.DriverManager
107. java.sql.DriverPropertyInfo
108. java.sql.JDBCType
109. java.sql.NClob

110. java.sql.ParameterMetaData
111. java.sql.PreparedStatement
112. java.sql.Ref
113. java.sql.ResultSet
114. java.sql.ResultSetMetaData
115. java.sql.RowId
116. java.sql.RowIdLifetime
117. java.sql.SQLClientInfoException
118. java.sql.SQLException
119. java.sql.SQLFeatureNotSupportedException
120. java.sql.SQLWarning
121. java.sql.SQLXML
122. java.sql.Savepoint
123. java.sql.Statement
124. java.sql.Struct
125. java.sql.Time
126. java.sql.Timestamp
127. java.sql.Types
128. java.text.DateFormat
129. java.text.DecimalFormat
130. java.text.NumberFormat
131. java.text.ParseException
132. java.text.SimpleDateFormat
133. java.time.Instant
134. java.util.ArrayList
135. java.util.Arrays
136. java.util.Base64
137. java.util.Calendar
138. java.util.Collection
139. java.util.Collections
140. java.util.Comparator
141. java.util.Date
142. java.util.EnumSet
143. java.util.Enumeration
144. java.util.HashMap
145. java.util.HashSet
146. java.util.Hashtable

147. java.util.IdentityHashMap
148. java.util.Iterator
149. java.util.LinkedHashMap
150. java.util.LinkedHashSet
151. java.util.LinkedList
152. java.util.List
153. java.util.Locale
154. java.util.Map
155. java.util.Map.Entry
156. java.util.NoSuchElementException
157. java.util.Objects
158. java.util.Optional
159. java.util.PriorityQueue
160. java.util.Properties
161. java.util.Queue
162. java.util.Random
163. java.util.Set
164. java.util.SortedMap
165. java.util.SortedSet
166. java.util.Spliterator
167. java.util.Spliterators
168. java.util.TimeZone
169. java.util.Timer
170. java.util.TimerTask
171. java.util.TreeMap
172. java.util.TreeSet
173. java.util.UUID
174. java.util.concurrent.ArrayBlockingQueue
175. java.util.concurrent.BlockingQueue
176. java.util.concurrent.BrokenBarrierException
177. java.util.concurrent.Callable
178. java.util.concurrent.ConcurrentHashMap
179. java.util.concurrent.ConcurrentLinkedQueue
180. java.util.concurrent.ConcurrentMap
181. java.util.concurrent.ConcurrentSkipListMap
182. java.util.concurrent.ConcurrentSkipListSet
183. java.util.concurrent.CountDownLatch

184.java.util.concurrent.CyclicBarrier

185.java.util.concurrent.ExecutionException

186.java.util.concurrent.Executor

187.java.util.concurrent.ExecutorService

188.java.util.concurrent.Executors

189.java.util.concurrent.Future

190.java.util.concurrent.LinkedBlockingQueue

191.java.util.concurrent.RejectedExecutionException

192.java.util.concurrent.ScheduledExecutorService

193.java.util.concurrent.ScheduledFuture

194.java.util.concurrent.ThreadLocalRandom

195.java.util.concurrent.TimeUnit

196.java.util.concurrent.TimeoutException

197.java.util.concurrent.atomic.AtomicBoolean

198.java.util.concurrent.atomic.AtomicInteger

199.java.util.concurrent.atomic.AtomicLong

200.java.util.concurrent.locks.Lock

201.java.util.concurrent.locks.ReadWriteLock

202.java.util.concurrent.locks.ReentrantLock

203.java.util.concurrent.locks.ReentrantReadWriteLock

204.java.util.function.Consumer

205.java.util.function.Function

206.java.util.function.Predicate

207.java.util.function.Supplier

208.java.util.jar.JarFile

209.java.util.logging.Handler

210.java.util.logging.Level

211.java.util.logging.LogManager

212.java.util.logging.Logger

213.java.util.regex.Matcher

214.java.util.regex.Pattern

215.java.util.regex.PatternSyntaxException

216.java.util.stream.Collectors

217.java.util.stream.LongStream

218.java.util.stream.Stream

219.java.util.stream.StreamSupport

220.java.util.zip.GZIPInputStream

221.java.util.zip.GZIPOutputStream

222.java.util.zip.ZipEntry

223.javax.annotation.Nonnull

224.javax.annotation.Nullable

225.javax.annotation.concurrent.NotThreadSafe

226.javax.crypto.BadPaddingException

227.javax.crypto.Cipher

228.javax.crypto.CipherInputStream

229.javax.crypto.CipherOutputStream

230.javax.crypto.IllegalBlockSizeException

231.javax.crypto.KeyGenerator

232.javax.crypto.NoSuchPaddingException

233.javax.crypto.SecretKey

234.javax.crypto.SecretKeyFactory

235.javax.crypto.spec.IvParameterSpec

236.javax.crypto.spec.PBEKeySpec

237.javax.crypto.spec.SecretKeySpec

238.javax.inject.Inject

239.javax.inject.Named

240.javax.inject.Singleton

241.javax.jms.Connection

242.javax.jms.Destination

243.javax.jms.JMSException

244.javax.jms.Message

245.javax.jms.MessageConsumer

246.javax.jms.MessageProducer

247.javax.jms.Session

248.javax.jms.TextMessage

249.javax.management.ServiceNotFoundException

250.javax.naming.NamingEnumeration

251.javax.naming.NamingException

252.javax.naming.OperationNotSupportedException

253.javax.naming.directory.Attribute

254.javax.naming.directory.SearchControls

255.javax.naming.directory.SearchResult

256.javax.naming.ldap.LdapContext

257.javax.net.SocketFactory

258. javax.net.ssl.CertPathTrustManagerParameters
259. javax.net.ssl.HostnameVerifier
260. javax.net.ssl.KeyManager
261. javax.net.ssl.KeyManagerFactory
262. javax.net.ssl.SSLContext
263. javax.net.ssl.SSLHandshakeException
264. javax.net.ssl.SSLServerSocket
265. javax.net.ssl.SSLSession
266. javax.net.ssl.SSLSocket
267. javax.net.ssl.SSLSocketFactory
268. javax.net.ssl.TrustManager
269. javax.net.ssl.TrustManagerFactory
270. javax.net.ssl.X509TrustManager
271. javax.persistence.Access
272. javax.persistence.AccessType
273. javax.persistence.AttributeOverride
274. javax.persistence.AttributeOverrides
275. javax.persistence.CascadeType
276. javax.persistence.Column
277. javax.persistence.ElementCollection
278. javax.persistence.Embeddable
279. javax.persistence.Embedded
280. javax.persistence.Entity
281. javax.persistence.EnumType
282. javax.persistence.Enumerated
283. javax.persistence.GeneratedValue
284. javax.persistence.Id
285. javax.persistence.JoinColumn
286. javax.persistence.JoinTable
287. javax.persistence.Lob
288. javax.persistence.ManyToMany
289. javax.persistence.ManyToOne
290. javax.persistence.OneToMany
291. javax.persistence.OneToOne
292. javax.persistence.Temporal
293. javax.persistence.TemporalType
294. javax.persistence.Transient

295. javax.persistence.criteria.CriteriaBuilder
296. javax.persistence.criteria.CriteriaQuery
297. javax.persistence.criteria.Join
298. javax.persistence.criteria.Predicate
299. javax.persistence.criteria.Root
300. javax.security.auth.login.Configuration
301. javax.servlet.*
302. javax.servlet.FilterChain
303. javax.servlet.FilterConfig
304. javax.servlet.ServletConfig
305. javax.servlet.ServletContext
306. javax.servlet.ServletException
307. javax.servlet.ServletRequest
308. javax.servlet.ServletResponse
309. javax.servlet.http.HttpServlet
310. javax.servlet.http.HttpServletResponse
311. javax.sql.DataSource
312. javax.validation.constraints.NotNull
313. javax.validation.constraints.Size
314. javax.ws.rs.*
315. javax.ws.rs.Consumes
316. javax.ws.rs.DELETE
317. javax.ws.rs.DefaultValue
318. javax.ws.rs.GET
319. javax.ws.rs.InternalServerErrorException
320. javax.ws.rs.NotFoundException
321. javax.ws.rs.POST
322. javax.ws.rs.PUT
323. javax.ws.rs.Path
324. javax.ws.rs.PathParam
325. javax.ws.rs.ProcessingException
326. javax.ws.rs.Produces
327. javax.ws.rs.QueryParam
328. javax.ws.rs.WebApplicationException
329. javax.ws.rs.client.Client
330. javax.ws.rs.client.Entity
331. javax.ws.rs.client.WebTarget

332.javax.ws.rs.container.ContainerRequestContext

333.javax.ws.rs.container.ContainerRequestFilter

334.javax.ws.rs.container.ContainerResponseContext

335.javax.ws.rs.container.ContainerResponseFilter

336.javax.ws.rs.container.PreMatching

337.javax.ws.rs.core.Context

338.javax.ws.rs.core.HttpHeaders

339.javax.ws.rs.core.MediaType

340.javax.ws.rs.core.MultivaluedMap

341.javax.ws.rs.core.Request

342.javax.ws.rs.core.Response

343.javax.ws.rs.core.SecurityContext

344.javax.ws.rs.core.StreamingOutput

345.javax.ws.rs.core.UriBuilder

346.javax.ws.rs.core.UriInfo

347.javax.ws.rs.ext.ExceptionMapper

348.javax.ws.rs.ext.MessageBodyReader

349.javax.ws.rs.ext.MessageBodyWriter

350.javax.ws.rs.ext.Provider

351.javax.xml.XMLConstants

352.javax.xml.bind.DatatypeConverter

353.javax.xml.bind.annotation.XmlTransient

354.javax.xml.parsers.DocumentBuilder

355.javax.xml.parsers.DocumentBuilderFactory

356.javax.xml.parsers.ParserConfigurationException

357.javax.xml.transform.Source

358.javax.xml.transform.dom.DOMSource

359.javax.xml.transform.stream.StreamSource

360.javax.xml.validation.Schema

361.javax.xml.validation.SchemaFactory

362.javax.xml.validation.Validator

The included GMP library imports these C/C++ headers:

1. algorithm
2. assert.h
3. cfloat
4. cstring
5. ctype.h
6. errno.h
7. fcntl.h
8. float.h
9. gmp.h
10. ia64intrin.h
11. intrinsics.h
12. inttypes.h
13. invent.h
14. iosfwd
15. langinfo.h
16. limits.h
17. limits
18. locale.h
19. machine/builtins.h
20. machine/hal_sysinfo.h
21. math.h
22. nl_types.h
23. obstack.h
24. readline/history.h
25. readline/readline.h
26. setjmp.h
27. signal.h
28. sstream
29. stdarg.h
30. stddef.h
31. stdexcept
32. stdint.h
33. stdio.h
34. stdlib.h
35. string.h
36. string
37. strstream
38. sys/attributes.h
39. sys/ioctl.h
40. sys/iograph.h
41. sys/mman.h
42. sys/param.h
43. sys/processor.h
44. sys/pstat.h
45. sys/resource.h
46. sys/sysctl.h
47. sys/sysinfo.h
48. sys/syssgi.h
49. sys/systemcfg.h
50. sys/time.h
51. sys/times.h
52. sys/types.h
53. time.h
54. type_traits
55. unistd.h
56. utility

10. Appendix: Dependencies

The Java dependencies are included in the project:

The dependencies are provided in the Maven identifier format of groupId : package name : version.

1. aopalliance:aopalliance:1.0
2. com.google.code.findbugs:jsr305:3.0.2
3. com.google.errorprone:error_prone_annotations:2.1.3
4. com.google.guava:guava:24.0-jre
5. com.google.inject:guice:4.1.0
6. com.google.j2objc:j2objc-annotations:1.1
7. javax.inject:javax.inject:1
8. javax.validation:validation-api:2.0.1.Final
9. log4j:log4j:1.2.17
10. net.java.dev.jna:jna:4.5.1
11. net.revelc.code:gnome-keyring-java:1.0.0-SNAPSHOT
12. org.checkerframework:checker-compat-qual:2.0.0
13. org.codehaus.mojo:animal-sniffer-annotations:1.14
14. com.google.code.gson:gson:2.8.2
15. it.unimi.dsi:fastutil:8.1.1
16. com.fasterxml.jackson.core:jackson-annotations:2.6.7
17. io.swagger:swagger-annotations:1.5.18
18. antlr:antlr:2.7.7
19. asm:asm:3.1
20. com.fasterxml:classmate:1.3.0
21. com.fasterxml.jackson.core:jackson-core:2.6.7
22. com.github.cliftonlabs:json-simple:2.3.0
23. com.github.jnr:jffi:1.2.9
24. com.github.jnr:jnr-ffi:2.0.5
25. com.github.jnr:jnr-x86asm:1.0.2
26. com.google.inject.extensions:guice-multibindings:4.1.0
27. com.google.protobuf:protobuf-java:2.5.0
28. com.h2database:h2:1.4.196
29. com.jamesmurty.utils:java-xmlbuilder:0.4
30. com.jcraft:jsch:0.1.54
31. com.lambdaworks:scrypt:1.4.0
32. com.mchange:c3p0:0.9.5.2
33. com.mchange:mchange-commons-java:0.2.11
34. com.microsoft.azure:azure-keyvault-core:1.0.0
35. com.microsoft.azure:azure-storage:7.0.0
36. com.nimbusds:nimbus-jose-jwt:3.9
37. com.squareup.okhttp:okhttp:2.4.0
38. com.squareup.okio:okio:1.4.0
39. com.sun.jersey:jersey-core:1.9
40. com.sun.jersey:jersey-server:1.9
41. com.thoughtworks.paranamer:paranamer:2.3
42. commons-beanutils:commons-beanutils:1.9.3
43. commons-cli:commons-cli:1.2
44. commons-codec:commons-codec:1.11
45. commons-collections:commons-collections:3.2.2
46. commons-configuration:commons-configuration:1.6
47. commons-daemon:commons-daemon:1.0.13
48. commons-digester:commons-digester:1.8
49. commons-io:commons-io:2.6
50. commons-lang:commons-lang:2.6
51. commons-logging:commons-logging:1.2

52. commons-net:commons-net:3.1
53. dom4j:dom4j:1.6.1
54. io.netty:netty:3.6.2.Final
55. io.netty:netty-all:4.0.56.Final
56. net.java.dev.jets3t:jets3t:0.9.0
57. net.jcip:jcip-annotations:1.0
58. net.jpountz.lz4:lz4:1.3.0
59. net.minidev:json-smart:1.1.1
60. org.abstractj.kalium:kalium:0.7.0
61. org.apache.avro:avro:1.7.4
62. org.apache.commons:commons-compress:1.4.1
63. org.apache.commons:commons-lang3:3.7
64. org.apache.commons:commons-math3:3.1.1
65. org.apache.curator:curator-framework:2.7.1
66. org.apache.curator:curator-recipes:2.7.1
67. org.apache.directory.api:api-asn1-api:1.0.0-M20
68. org.apache.directory.api:api-util:1.0.0-M20
69. org.apache.directory.server:apacheds-i18n:2.0.0-M15
70. org.apache.directory.server:apacheds-kerberos-codec:2.0.0-M15
71. org.apache.hadoop:hadoop-annotations:2.8.3
72. org.apache.hadoop:hadoop-auth:2.8.3
73. org.apache.hadoop:hadoop-azure:2.8.3
74. org.apache.hadoop:hadoop-client:2.8.3
75. org.apache.hadoop:hadoop-common:2.8.3
76. org.apache.hadoop:hadoop-hdfs:2.8.3
77. org.apache.hadoop:hadoop-hdfs-client:2.8.3
78. org.apache.hadoop:hadoop-mapreduce-client-app:2.8.3
79. org.apache.hadoop:hadoop-mapreduce-client-common:2.8.3
80. org.apache.hadoop:hadoop-mapreduce-client-jobclient:2.8.3
81. org.apache.hadoop:hadoop-mapreduce-client-shuffle:2.8.3
82. org.apache.hadoop:hadoop-yarn-api:2.8.3
83. org.apache.hadoop:hadoop-yarn-client:2.8.3
84. org.apache.hadoop:hadoop-yarn-server-common:2.8.3
85. org.apache.htrace:htrace-core4:4.0.1-incubating
86. org.apache.httpcomponents:httpclient:4.5.2
87. org.apache.httpcomponents:httpcore:4.4.4
88. org.apache.lucene:lucene-analyzers-common:7.2.1
89. org.apache.lucene:lucene-core:7.2.1
90. org.apache.shiro:shiro-cache:1.4.0
91. org.apache.shiro:shiro-config-core:1.4.0
92. org.apache.shiro:shiro-config-ogdl:1.4.0
93. org.apache.shiro:shiro-core:1.4.0
94. org.apache.shiro:shiro-crypto-cipher:1.4.0
95. org.apache.shiro:shiro-crypto-core:1.4.0
96. org.apache.shiro:shiro-crypto-hash:1.4.0
97. org.apache.shiro:shiro-event:1.4.0
98. org.apache.shiro:shiro-lang:1.4.0
99. org.apache.zookeeper:zookeeper:3.4.6
100. org.codehaus.jackson:jackson-core-asl:1.9.13
101. org.codehaus.jackson:jackson-mapper-asl:1.9.13
102. org.fusesource.leveldbjni:leveldbjni-all:1.8
103. org.glassfish.grizzly:grizzly-framework:2.4.3
104. org.hibernate:hibernate-c3p0:5.2.12.Final
105. org.hibernate:hibernate-core:5.2.12.Final
106. org.hibernate.common:hibernate-commons-annotations:5.0.1.Final
107. org.hibernate.javax.persistence:hibernate-jpa-2.1-api:1.0.2.Final

108. org.javassist:javassist:3.20.0-GA
109. org.jboss:jandex:2.0.3.Final
110. org.jboss.logging:jboss-logging:3.3.0.Final
111. org.jboss.spec.javax.transaction:jboss-transaction-api_1.2_spec:1.0.1.Final
112. org.mortbay.jetty:jetty:6.1.26
113. org.mortbay.jetty:jetty-sslengine:6.1.26
114. org.mortbay.jetty:jetty-util:6.1.26
115. org.ow2.asm:asm:6.0
116. org.ow2.asm:asm-analysis:6.0
117. org.ow2.asm:asm-commons:6.0
118. org.ow2.asm:asm-tree:5.0.3
119. org.ow2.asm:asm-util:6.0
120. org.scala-lang:scala-library:2.11.12
121. org.slf4j:slf4j-api:1.7.25
122. org.slf4j:slf4j-log4j12:1.7.25
123. org.tukaani:xz:1.0
124. org.xerial.snappy:snappy-java:1.0.4.1
125. xerces:xercesImpl:2.9.1
126. xmlenc:xmlenc:0.52
127. com.101tec:zkclient:0.9
128. com.amazonaws:aws-java-sdk-core:1.11.276
129. com.amazonaws:aws-java-sdk-dynamodb:1.11.276
130. com.amazonaws:aws-java-sdk-kms:1.11.276
131. com.amazonaws:aws-java-sdk-s3:1.11.276
132. com.amazonaws:jmespath-java:1.11.276
133. com.clearspring.analytics:stream:2.7.0
134. com.esotericsoftware:kryo-shaded:4.0.1
135. com.esotericsoftware:minlog:1.3.0
136. com.fasterxml.jackson.core:jackson-databind:2.6.7.1
137. com.fasterxml.jackson.dataformat:jackson-dataformat-cbor:2.6.7
138. com.fasterxml.jackson.dataformat:jackson-dataformat-yaml:2.9.3
139. com.fasterxml.jackson.module:jackson-module-paranamer:2.6.5
140. com.fasterxml.jackson.module:jackson-module-scala_2.11:2.6.5
141. com.github.traviscrawford:spark-dynamodb:0.0.11
142. com.google.code.findbugs:annotations:3.0.1u2
143. com.intellij:annotations:12.0
144. com.ning:compress-lzf:1.0.3
145. com.sun.jersey:jersey-client:1.9
146. com.twitter:chill-java:0.8.0
147. com.twitter:chill_2.11:0.8.0
148. com.twitter:util-app_2.11:6.43.0
149. com.twitter:util-core_2.11:6.43.0
150. com.twitter:util-function_2.11:6.43.0
151. com.twitter:util-registry_2.11:6.43.0
152. com.univocity:univocity-parsers:2.6.2
153. com.yammer.metrics:metrics-core:2.2.0
154. commons-beanutils:commons-beanutils-core:1.8.0
155. commons-httpclient:commons-httpclient:3.1
156. io.dropwizard.metrics:metrics-core:3.1.2
157. io.dropwizard.metrics:metrics-graphite:3.1.2
158. io.dropwizard.metrics:metrics-json:3.1.2
159. io.dropwizard.metrics:metrics-jvm:3.1.2
160. io.swagger:swagger-core:1.5.18
161. io.swagger:swagger-jaxrs:1.5.18
162. io.swagger:swagger-models:1.5.18
163. javax.activation:activation:1.1

164. javax.annotation:javax.annotation-api:1.3.1
165. javax.servlet:javax.servlet-api:3.1.0
166. javax.servlet:servlet-api:2.5
167. javax.ws.rs:javax.ws.rs-api:2.1
168. javax.xml.bind:jaxb-api:2.2.2
169. javax.xml.stream:stax-api:1.0-2
170. joda-time:joda-time:2.8.1
171. mx4j:mx4j:3.0.2
172. mysql:mysql-connector-java:6.0.6
173. net.java.dev.jna:jna-platform:4.4.0
174. net.openhft:affinity:3.1.7
175. net.razorvine:pyrolite:4.13
176. net.sf.jopt-simple:jopt-simple:4.9
177. net.sf.py4j:py4j:0.10.4
178. org.antlr:antlr4-runtime:4.5.3
179. org.apache.activemq:activemq-all:5.15.2
180. org.apache.avro:avro-ipc:1.7.7
181. org.apache.avro:avro-mapred:1.7.7
182. org.apache.commons:commons-crypto:1.0.0
183. org.apache.curator:curator-client:4.0.1
184. org.apache.curator:curator-framework:4.0.1
185. org.apache.curator:curator-recipes:4.0.1
186. org.apache.derby:derby:10.14.1.0
187. org.apache.derby:derbyclient:10.14.1.0
188. org.apache.hadoop:hadoop-aws:2.8.3
189. org.apache.hadoop:hadoop-mapreduce-client-core:2.8.3
190. org.apache.hadoop:hadoop-yarn-common:2.8.3
191. org.apache.ivy:ivy:2.4.0
192. org.apache.kafka:kafka-clients:0.10.1.0
193. org.apache.kafka:kafka_2.11:0.10.1.0
194. org.apache.parquet:parquet-column:1.8.2
195. org.apache.parquet:parquet-common:1.8.2
196. org.apache.parquet:parquet-encoding:1.8.2
197. org.apache.parquet:parquet-format:2.3.1
198. org.apache.parquet:parquet-hadoop:1.8.2
199. org.apache.parquet:parquet-jackson:1.8.2
200. org.apache.shiro:shiro-guice:1.4.0
201. org.apache.spark:spark-catalyst_2.11:2.2.1
202. org.apache.spark:spark-core_2.11:2.2.1
203. org.apache.spark:spark-launcher_2.11:2.2.1
204. org.apache.spark:spark-network-common_2.11:2.2.1
205. org.apache.spark:spark-network-shuffle_2.11:2.2.1
206. org.apache.spark:spark-sketch_2.11:2.2.1
207. org.apache.spark:spark-sql_2.11:2.2.1
208. org.apache.spark:spark-streaming_2.11:2.2.1
209. org.apache.spark:spark-tags_2.11:2.2.1
210. org.apache.spark:spark-unsafe_2.11:2.2.1
211. org.apache.xbean:xbean-asm5-shaded:4.4
212. org.apache.zookeeper:zookeeper:3.5.3-beta
213. org.codehaus.jackson:jackson-jaxrs:1.9.13
214. org.codehaus.jackson:jackson-xc:1.9.13
215. org.codehaus.janino:commons-compiler:3.0.0
216. org.codehaus.janino:janino:3.0.0
217. org.elasticsearch:elasticsearch-hadoop:6.2.1
218. org.glassfish.grizzly:grizzly-http:2.4.3
219. org.glassfish.grizzly:grizzly-http-server:2.4.3

220. org.glassfish.grizzly:grizzly-http-servlet:2.4.3
221. org.glassfish.grizzly:grizzly-websockets:2.4.3
222. org.glassfish.hk2:hk2-api:2.5.0-b61
223. org.glassfish.hk2:hk2-locator:2.5.0-b61
224. org.glassfish.hk2:hk2-utils:2.5.0-b61
225. org.glassfish.hk2:osgi-resource-locator:1.0.1
226. org.glassfish.hk2.external:aopalliance-repackaged:2.5.0-b61
227. org.glassfish.hk2.external:javax.inject:2.5.0-b61
228. org.glassfish.jersey.containers:jersey-container-grizzly2-http:2.26
229. org.glassfish.jersey.containers:jersey-container-servlet:2.22.2
230. org.glassfish.jersey.containers:jersey-container-servlet-core:2.22.2
231. org.glassfish.jersey.core:jersey-client:2.26
232. org.glassfish.jersey.core:jersey-common:2.26
233. org.glassfish.jersey.core:jersey-server:2.26
234. org.glassfish.jersey.inject:jersey-hk2:2.26
235. org.glassfish.jersey.media:jersey-media-jaxb:2.26
236. org.glassfish.jersey.media:jersey-media-multipart:2.26
237. org.javassist:javassist:3.22.0-CR2
238. org.json4s:json4s-ast_2.11:3.2.11
239. org.json4s:json4s-core_2.11:3.2.11
240. org.json4s:json4s-jackson_2.11:3.2.11
241. org.jvnet.mimepull:mimepull:1.9.6
242. org.mongodb:mongo-java-driver:3.4.2
243. org.mongodb.spark:mongo-spark-connector_2.11:2.2.1
244. org.objenesis:objenesis:2.5.1
245. org.ow2.asm:asm-analysis:5.0.3
246. org.ow2.asm:asm-commons:5.0.3
247. org.ow2.asm:asm-util:5.0.3
248. org.postgresql:postgresql:42.2.1
249. org.reflections:reflections:0.9.11
250. org.roaringbitmap:RoaringBitmap:0.5.11
251. org.scala-lang:scala-reflect:2.11.12
252. org.scala-lang.modules:scala-parser-combinators_2.11:1.0.4
253. org.scala-lang.modules:scala-xml_2.11:1.0.6
254. org.slf4j:jcl-over-slf4j:1.7.16
255. org.slf4j:jul-to-slf4j:1.7.16
256. org.spark-project.spark:unused:1.0.0
257. org.xerial.snappy:snappy-java:1.1.2.6
258. org.yaml:snakeyaml:1.18
259. oro:oro:2.0.8
260. software.amazon.ion:ion-java:1.0.2
261. xml-apis:xml-apis:1.4.01
262. com.bettercloud:vault-java-driver:2.0.0
263. com.facebook.presto:presto-parser:0.189
264. com.fasterxml.jackson.jaxrs:jackson-jaxrs-base:2.6.7
265. com.fasterxml.jackson.jaxrs:jackson-jaxrs-json-provider:2.6.7
266. com.fasterxml.jackson.module:jackson-module-jaxb-annotations:2.6.7
267. com.google.inject.extensions:guice-assistedinject:4.1.0
268. com.neovisionaries:nv-websocket-client:2.3
269. com.squareup.okhttp:okhttp:2.7.5
270. com.squareup.okio:okio:1.6.0
271. io.airlift:slice:0.32
272. io.jsonwebtoken:jjwt:0.9.0
273. javax.annotation:javax.annotation-api:1.2
274. javax.servlet:javax.servlet-api:4.0.0
275. javax.xml.bind:jaxb-api:2.3.0

276. org.antlr:antlr4-runtime:4.6
277. org.glassfish.hk2:hk2-api:2.5.0-b42
278. org.glassfish.hk2:hk2-locator:2.5.0-b42
279. org.glassfish.hk2:hk2-utils:2.5.0-b42
280. org.glassfish.hk2.external:aopalliance-repackaged:2.5.0-b42
281. org.waxeye:waxeye:0.8.1
282. ch.qos.cal10n:cal10n-api:0.8.1
283. com.atlassian.commonmark:commonmark:0.9.0
284. com.fasterxml.jackson.core:jackson-annotations:2.9.3
285. com.fasterxml.jackson.core:jackson-core:2.9.3
286. com.fasterxml.jackson.core:jackson-databind:2.9.3
287. com.github.fge:btf:1.2
288. com.github.fge:jackson-coreutils:1.6
289. com.github.fge:json-patch:1.6
290. com.github.fge:msg-simple:1.1
291. com.github.fge:uri-template:0.9
292. com.github.java-json-tools:json-schema-core:1.2.8
293. com.github.java-json-tools:json-schema-validator:2.2.8
294. com.google.code.findbugs:jsr305:3.0.1
295. com.google.guava:guava:20.0
296. com.googlecode.libphonenumber:libphonenumber:8.0.0
297. com.samskivert:jmustache:1.12
298. com.squareup.okhttp:logging-interceptor:2.7.5
299. com.squareup.okio:okio:1.14.0
300. commons-codec:commons-codec:1.9
301. commons-io:commons-io:2.4
302. io.swagger:swagger-codegen:2.3.1
303. io.swagger:swagger-compat-spec-parser:1.0.33
304. io.swagger:swagger-parser:1.0.33
305. javax.mail:mailapi:1.4.3
306. javax.validation:validation-api:1.1.0.Final
307. joda-time:joda-time:2.9.9
308. junit:junit:4.12
309. net.sf.jopt-simple:jopt-simple:5.0.3
310. org.apache.commons:commons-lang3:3.4
311. org.hamcrest:hamcrest-core:1.3
312. org.mozilla:rhino:1.7R4
313. org.slf4j:slf4j-api:1.7.12
314. org.slf4j:slf4j-ext:1.7.12
315. org.slf4j:slf4j-simple:1.7.12
316. ch.qos.logback:logback-classic:1.2.3
317. ch.qos.logback:logback-core:1.2.3
318. com.fasterxml.jackson.core:jackson-annotations:2.6.0
319. io.swagger:swagger-jersey2-jaxrs:1.5.18
320. org.apache.commons:commons-lang3:3.2.1
321. org.glassfish.grizzly:grizzly-http-servlet:2.4.0
322. org.glassfish.hk2.external:javax.inject:2.5.0-b42
323. org.glassfish.jersey.containers:jersey-container-grizzly2-servlet:2.26
324. org.glassfish.jersey.containers:jersey-container-servlet:2.26
325. org.glassfish.jersey.containers:jersey-container-servlet-core:2.26
326. org.javassist:javassist:3.21.0-GA
327. javax.ws.rs:jsr311-api:1.1.1
328. org.codehaus.plexus:plexus-utils:3.1.0
329. cglib:cglib-nodep:3.2.5
330. com.facebook.presto:presto-base-jdbc:0.189
331. com.fasterxml.jackson.core:jackson-annotations:2.8.1

332.com.google.inject.extensions:guice-multibindings:4.0
333.com.sun.xml.bind:jaxb-impl:2.2.6
334.io.airlift:bootstrap:0.163
335.io.airlift:concurrent:0.156
336.io.airlift:configuration:0.163
337.io.airlift:log:0.156
338.io.airlift:log-manager:0.163
339.io.airlift:units:1.0
340.javax.xml.bind:jaxb-api:2.2.6
341.org.apache.bval:bval-core:1.1.1
342.org.apache.bval:bval-jsr:1.1.1
343.org.weakref:jmxutils:1.19
344.com.facebook.presto:presto-jdbc:0.189
345.com.google.code.findbugs:jsr305:1.3.9