Security Target

Zeta Server v1.1.1

Version	v1.3
Date	2021.06.18.

Version history

Version	Date	Author	Description
v1.0	2021.03.04.	Prolan Power	Initial version of the Security Target.
V1.1	2021.05.10.	Prolan Power	TOE scope and relation to Zeta Solution clarified. Extended Non-TOE Hardware /Software /Firmware. Updated credential storing parts.
V1.2	2021.05.25.	Prolan Power	Added Timely Security Updates Updated TOE version
V1.3	2021.06.18.	Prolan Power	Updated Platform API used by the TOE.

Table of Contents

1 Introduction

1.1 Security Target and TOE References

ST Title	Security Target - Zeta Server v1.1.1
ST Version	v1.3
ST Creation Date	2021.06.18.
TOE Reference	Zeta Server v1.1.1

1.2 TOE Overview

Zeta is an enterprise monitoring and supervisory control (SCADA) solution used to collect analogue and binary measurements of electrical equipment from a substation gateway and provides the possibility of issuing commands to that equipment if needed through a Web GUI that facilitates the visualization of the substation.

Zeta Solution consists of Zeta Server (TOE) and Substation Automation Gateway.

Substation Automation Gateway (SAGateway) interfaces with the Zeta Server application and relays collected data from the substation via various standard communication protocols and also provides the possibility to issue commands to technology devices. Without this connection being established Zeta Server's functionality is very limited. SAGateway is not subject to the TOE.

Zeta Server (TOE) consists of the following subsystems:

- **Zeta Substation Controller (ZSC)**. It heavily relies on data collected from Substation Automation Gateway and responsible for additional application calculations what supply necessary information for the Web GUI visualization relayed through the Zeta Client Manager.
- **Zeta Client Manager (ZCM)**. It is a web server application that supplies the necessary data to clients for substation visualization – according to user role - via secure SSL connection. It also interfaces Zeta Substation Controller to relay substation-related messages to clients, also authorizes messages from clients and forwards them to Zeta Substation Controller. This software component is responsible for the majority of security functions on the server-side.
    - **Web GUI** is the most important module of ZCM. In their web browser authorized users can see the current state of the substation in a visualized form and leverage the features provided by the software such as getting detailed information about technology devices, being informed about important changes and immediately alarmed on unexpected behavior. This interface usually consists of many views depending on the Substation Model, with various contents helping the operators to focus on different subsets of technology devices or the whole substation. Many of the security related settings can be changed through the security administration page. Users can also manage Substation Models on the corresponding management page.

Later on, the Zeta Client Manager and Zeta Substation Controller together are referred as Zeta Server. The TOE is the Zeta Server v1.1.1 which is a software application.

User management, user role management and all security-related settings are handled by the Zeta Client Manager (invoked through Web GUI) and included in the TOE.

1.2.1 TOE Usage and Major Security Features

The software-only TOE is the Zeta Server v1.1.1. It is a server application that runs on a Linux operation system and provides monitoring and supervisory control functionality for the users of the application.
The following Zeta Server application capabilities are considered to be within the scope of the evaluation:

- Trusted communication of user credential data and configuration data between the TOE and the operational environment.
- The extent to which the TSF relies on platform-provided and third-party library capabilities to perform its functionality.
- The extent to which data used to determine the behavior of the TSF is secured while at rest and in transit.
- The ability of the TOE to interface with the low-level components of its host platform in such a manner that the TOE cannot be used as an attack vector to exploit the host platform.

The basic workflows of the TOE usage are listed below:

User role selection

1. In operational environment, User logs on to the Web GUI and chooses an eligible role preconfigured by the Security Administrator.
2. Chosen role is transmitted to the Zeta Client Manager over HTTPS connection and being validated by the TOE.
3. If it is found to be a valid role for the particular user it is bound to that client instance therefore all visualization and user action is authorized according to that role.
4. Currently active user role cannot be changed on the Web GUI. The user must log out, log in again and select a different role.

Security Administration

1. In the operational environment, User logs on to Web GUI and after selecting the Security Adminisrator role, the User is redirected to the Security Administration page.
2. User makes user-related changes such as adding new user or editing existing user's settings.
3. The changes are transmitted to the Zeta Client Manager over HTTPS connection.

When received by the ZCM the changes are authorized and propagated to the PostgreSQL Database if applicable.

Operator action

1. In operational environment, User logs on to the Web GUI and after selecting an appropriate user role the User is redirected to the Operator page.
2. User issues an action such as operating a technology device, blocking a datapoint from being updated or appending an operator-note to the application logs.
3. Command describing the user action is transmitted to the Zeta Server application and being validated by the TOE.
4. If it is found to be a valid command then an application log is created indicating the user's intent. The log is stored in the PostgreSQL Database and forwarded to other

client instances if needed. One of the following actions can be performed by the Server application based on the command.

5. Operation commands are relayed to the substation gateway and a notification message is sent by the server to all client instances indicating an operation is in progress and further operation commands are temporarily disabled until the one in progress is succeeded or failed.
6. Further user actions not related to the substation gateway are evaluated by the Server application itself and changes are propagated to the PostgreSQL Database or the underlying file system if needed.

Substation Model management

1. In operational environment, User logs on to the Web GUI and after selecting the Engineer role the User is redirected to the Substation Model page.
2. User can upload new substation model, delete an already existing one or activate any previously uploaded model. Applied changes will be sent to Zeta server and will be validated on the server side.

1.2.2 Non-TOE Hardware/Software/Firmware

All parts of the Zeta Server v1.1.1 are part of the TOE and the scope of the evaluation.

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

- 4x 2GHz cores
- 8 GB RAM
- Minimum of 1 GB disk storage

The TOE can be installed on 1 GB of free space, but we recommend at least 30 GB disk storage for storing logs and data history.

Zeta Server requires some software components to be installed on the host platform prior to its installation. These software are included in the following list with versions of them used in the evaluated TOE version. It is possible to use different versions but the TOE is only tested with the listed versions, therefore it is not guaranteed to properly function with other versions.

- postgresql-server (9.2)
- python3 (3.6.8)
- nginx (1.16.1)

  (many webservers can be used but we recommend Nginx and the guide only includes instructions accordingly)
- python-virtualenv (15.1.0)
- openssl (1.0.2k)
- sagateway (3.7.6)

1.3 TOE Description

Figure 1 illustrates the physical boundary of TOE and ties together other components of the product and TOE environment.

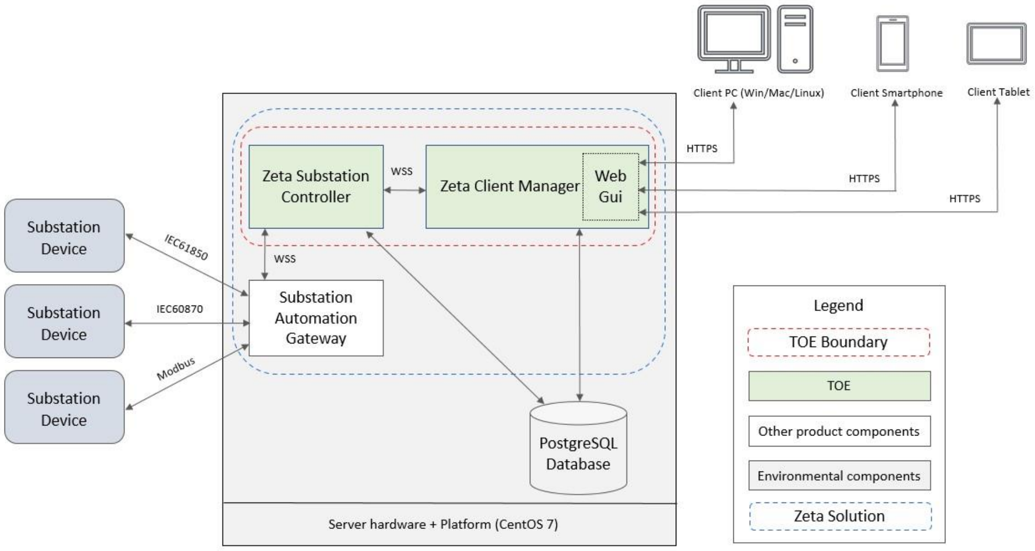

Figure 1TOE architecture

The TOE consists of Zeta Client Manager and Zeta Substation Controller.

Zeta Client Manager is a server application dealing with all requests coming from clients via the Web GUI. This module is responsible for server-side user administration functionality, also user authentication and authorization. It distributes authorized application messages between clients and the Zeta Substation Controller. It stores user session information and login attempt logs in a PostgreSQL database.

Zeta Substation Controller is the communication agent between Substation Automation Gateway and Zeta Client Manager. It is processing analogue or binary measurements from Substation Automation Gateway whilst doing further calculations on those. Those data are necessary for the visualization of the substation done by the Zeta Client Manager through the Web GUI. It stores measurement states and their history, also commands issued by users in the PostgreSQL database.

Substation Automation Gateway (SAGateway) relays analogue or binary measurements to Zeta Substation Controller from Substation Devices according to the Substation Model. It collects data from various Substation Devices through a variety of standard communication protocols and also relays user commands to those coming from Zeta Server users. SAGateway is not subject to the TOE.

The TOE's operational environment includes the following:

- Platform on which the TOE is hosted. The TOE is capable of running on a general-purpose Linux operating system on a consumer-grade hardware.
- Other product components. Substation Automation Gateway is not mandatory but expected to be present. Without it the functionality of the TOE is very limited.
- A PostgreSQL Database containing user and application data, logs and history of measurement values of the substation devices.

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

- 4x 2GHz cores
- 8 GB RAM
- Minimum of 1 GB disk storage

The TOE can be installed on 1 GB of free space but we recommend at least 30 GB disk storage for storing logs and data history.

1.3.1 Physical Scope of the TOE

The physical scope of the TOE is the Zeta Server v1.1.1. It has two subsystems:

- Zeta Client Manager
- Zeta Substation Controller

The physical parts of the TOE are as follows:

- Zeta_Server_1.1.1.zip, which includes the install package of the Zeta Server and a setup directory containing additional script files and environmental configuration files supporting the installation process.
- Guidance Documentation – Zeta Server v1.1.1 [AGD]

Following the purchase of the product Customers are registered on the Developer's site and after authentication they can download all parts of the TOE.

1.3.2 Logical Scope of the TOE

This section summarizes the security functions provided by the TOE. The SFRs implemented by the TOE are usefully grouped under the following Security Function Classes:

Cryptographic support

Zeta Server protects its data using platforms built in random generator. Generates session identifiers, password encryption keys and other random values to defend against potential attacks.

User data protection

The TOE protects sensitive data in non-volatile memory according to the requirements in FCS_STO_EXT.1. The TOE restricts its access to network connectivity provided by the platform's hardware resources and does not access any of the platform's sensitive information repositories.

Security Management

The Zeta Server comes with default credentials and provides a Web GUI for user administration. The Security Administrator user can log in to the Web GUI using default credentials and has to change its password immediately. The Security Administrator can add or modify users and set user roles using the Web GUI.

Users and their passwords are stored in the PostgreSQL database in a hashed form.

Privacy

The TOE does not handle personally identifiable information (PII).

TSF Protection

The TOE is compatible with its host OS platform when that is configured in a secured manner, using SELinux.
The TOE uses a well-defined set of platform APIs and third-party libraries.
The TOE provides the ability for the Installer to check its version and if an update is available. Internet connection needed for the latter.
Updates are delivered in formats appropriate for the platform on which the TOE is hosted. It is digitally signed, and the signature is validated prior to installation. Installing an update of the application removes all previously installed files except configuration and audit/log files.

Trusted Channel/Path

The TOE encrypts data in transit between itself and clients as well as the Substation Automation Gateway using HTTPS. The TOE relies on the platform to implement HTTPS encryption.

2 Conformance Claims Protection Profile Conformance Rationale

This section provides the identification for any CC, PP, Technical Decisions (TD), and EAL package conformance claims. Rationale is provided for any extensions or augmentations to the conformance claims. Rationale for CC and PP conformance claims can be found in Section 2.1.

Common Criteria (CC) Identification and Conformance

- Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, April 2017;
- CC Part 2 extended
- CC Part 3 extended
- PP claim to the *Protection Profile for Application Software v1.3; March 01, 2019* [PP] conformant

This ST claims exact conformance to [PP].

2.1 TD Conformance

The table below contains the technical decisions related to the [PP].

0540 – Expanded AES Modes in FCS_COP	Applied
0521 – Updates to Certificate Revocation (FIA_X509_EXT.1)	Not applicable as FIA_X509_EXT is not included.

0519 – Linux symbolic links and FMT_CFG_EXT.1	Applied
0498 – Application Software PP Security Objectives and Requirements Rationale	Applied
0495 – FIA_X509_EXT.1.2 Test Clarification	Not Applied as FIA_X509_EXT.1.2 is not included.
0486 – Removal of PP-Module for VPN Clients from allowed with list	Applied
0473 – Support for Client or Server TOEs in FCS_HTTPS_EXT	Not Applicable as FCS_HTTPS_EXT is not included.
0445 – User Modifiable File Definition	Applied
0444 – IPsec selections	Applied
0437 – Supported Configuration Mechanism	Applied
0435 – Alternative to SELinux for FPT_AEX_EXT.1.3	Not applicable. SELinux is used.
0427 – Reliable Time Source	Applied
0416 – Correction to FCS_RBG_EXT.1 Test Activity	Not applicable because the TOE doesn’t implement DRBG functionality.

3 Security Problem Description

The security problem is described in terms of the threats that the TOE is expected to address, assumptions about the operational environment, and any organizational security policies that the TOE is expected to enforce.

3.1 Threats

3.1.1 T.NETWORK_ATTACK

An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may engage in communications with the application software or alter communications between the application software and other endpoints in order to compromise it.

3.1.2 T.NETWORK_EAVESDROP

An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may monitor and gain access to data exchanged between the application and other endpoints.

3.1.3 T.LOCAL_ATTACK

An attacker can act through unprivileged software on the same computing platform on which the application executes.

Attackers may provide maliciously formatted input to the application in the form of files or other local communications.

3.1.4 T.PHYSICAL_ACCESS

An attacker may try to access sensitive data at rest.

3.2 Assumptions

3.2.1 A.PLATFORM

The TOE relies upon a trustworthy computing platform with a reliable clock for its execution. This includes the underlying platform and whatever runtime environment it provides to the TOE.

3.2.2 A.PROPER_USER

The user of the application software is not willfully negligent or hostile, and uses the software in compliance with the applied enterprise security policy.

3.2.3 A.PROPER_ADMIN

The administrator[1] of the application software is not careless, willfully negligent or hostile, and administers the software in compliance with the applied enterprise security policy.

3.3 Organizational Security Policies

There are no Organizational Security Policies for the application.

[1] The ST uses roles defined by [IEC 62351-8] but because of strict conformance to [PP] we didn't update roles in parts adopted from [PP]. The corresponding role for the administration of the application software is the Installer role.

4 Security Objectives

4.1 Security Objectives for the TOE

4.1.1 O.INTEGRITY

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

Addressed by: FDP_DEC_EXT.1, FMT_CFG_EXT.1, FPT_AEX_EXT.1, FPT_TUD_EXT.1

4.1.2 O.QUALITY

To ensure quality of implementation, conformant TOE's leverage services and APIs provided by the runtime environment rather than implementing their own versions of these services and APIs. This is especially important for cryptographic services and other complex operations such as file and media parsing. Leveraging this platform behaviour relies upon using only documented and supported APIs.

Addressed by: FMT_MEC_EXT.1, FPT_API_EXT.1, FPT_LIB_EXT.1, FPT_TUD_EXT.2, FCS_CKM.1

4.1.3 O.MANAGEMENT

To facilitate management by users and the enterprise, conformant TOEs provide consistent and supported interfaces for their security-relevant configuration and maintenance. This includes the deployment of applications and application updates through the use of platform-supported deployment mechanisms and formats, as well as providing mechanisms for configuration. This also includes providing control to the user regarding disclosure of any PII.

Addressed by: FMT_SMF.1, FPT_IDV_EXT.1, FPT_TUD_EXT.1, FPR_ANO_EXT.1

4.1.4 O.PROTECTED_STORAGE

To address the issue of loss of confidentiality of user data in the event of loss of physical control of the storage medium, conformant TOEs will use data-at-rest protection. This involves encrypting data and keys stored by the TOE in order to prevent unauthorized access to this data. This also includes unnecessary network communications whose consequence may be the loss of data.

Addressed by: FDP_DAR_EXT.1, FCS_STO_EXT.1, FCS_RBG_EXT.1, FCS_COP.1

4.1.5 O.PROTECTED_COMMS

To address both passive (eavesdropping) and active (packet modification) network attack threats, conformant TOEs will use a trusted channel for sensitive data. Sensitive data includes cryptographic keys, passwords, and any other data specific to the application that should not be exposed outside of the application.

Addressed by: FTP_DIT_EXT.1, FCS_RBG_EXT.1, FCS_CKM_EXT.1, FDP_NET_EXT.1

4.2 Security Objectives for the Operational Environment

The following security objectives for the operational environment assist the TOE in correctly providing its security functionality.

These track with the assumptions about the environment.

4.2.1 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.

4.2.2 OE.PROPER_USER

The user of the application software is not wilfully negligent or hostile, and uses the software within compliance of the applied enterprise security policy.

4.2.3 OE.PROPER_ADMIN

The administrator of the application software is not careless, wilfully negligent or hostile, and administers the software within compliance of the applied enterprise security policy.

4.3 Security Objectives Rationale

This section describes how the assumptions, threats, and organizational security policies map to the security objectives.

The rationale can be found in TD0498.

5 Security Requirements

This chapter describes the security requirements which have to be fulfilled by the TOE. Those requirements comprise functional components from Part 2 and assurance components from Part 3 of [CC]. The following notations are used:

- **Refinement** operation (denoted by **bold text**): is used to add details to a requirement, and thus further restricts a requirement.
- **Selection** (denoted by *italicized text*): is used to select one or more options provided by the [CC] in stating a requirement.

- **Assignment** operation (denoted by underlined text): is used to assign a specific value to an unspecified parameter, such as the length of a password. Showing the value in square brackets indicates assignment.
- **Iteration** operation: are identified with a number inside parentheses (e.g."(1)")

5.1 Security Functional Requirements

5.1.1 Cryptographic Support (FCS)

FCS_RBG_EXT.1 – Random Bit Generation Services

FCS_RBG_EXT.1.1 The application shall [

- *invoke platform-provided DRBG functionality*

] [2] for its cryptographic operations.

FCS_CKM_EXT.1 – Cryptographic Key Generation Services

FCS_CKM_EXT.1.1 The application shall [

- *invoke platform-provided functionality for asymmetric key generation*

][3].

FCS_CKM.1 – Cryptographic Asymmetric Key Generation

FCS_CKM.1.1 The application shall [

- *invoke platform-provided functionality*

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

- ***[RSA schemes]*** *using cryptographic key sizes of* ***[2048-bit or greater]*** *that meet the following* ***FIPS PUB 186-4, "Digital Signature Standard (DSS), Appendix B.3"***

][5].

[2] [selection: use no DRBG functionality, invoke platform-provided DRBG functionality, implement DRBG functionality]

[3] [selection: generate no asymmetric cryptographic keys, invoke platform-provided functionality for asymmetric key generation, implement asymmetric key generation]

[4] [selection: invoke platform-provided functionality, implement functionality]

[5] [selection: [RSA schemes] using cryptographic key sizes of[2048-bit or greater] that meet the following FIPS PUB 186-4, "Digital Signature Standard (DSS), Appendix B.3" , [ECC schemes] using ["NIST curves" P-256, P-384 and [selection: P-521 , no other curves]]that meet the following: [FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4] , [FFC schemes] using cryptographic key sizes of[2048-bit or greater] that meet the following: [FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.1], [FFC Schemes] using Diffie-Hellman group 14that meet the following: RFC 3526, Section 3 , [FFC Schemes] using "safe-prime" groups

FCS_STO_EXT.1 – Storage of Credentials

FCS_STO_EXT.1.1 The application shall [

- *implement functionality to securely store* [database password, user login password][6] *according to* [*FCS_COP.1*][7]

][8] to non-volatile memory.

that meet the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and [selection: RFC 3526, RFC 7919]

[6] [assignment: list of credentials]

[7] [selection: FCS_COP.1(1), FCS_CKM.1(3)]

[8] [selection: not store any credentials, invoke the functionality provided by the platform to securely store [assignment: list of credentials], implement functionality to securely store [assignment: list of credentials] according to [selection: FCS_COP.1(1), FCS_CKM.1(3)]]

FCS_COP.1 – Cryptographic Operation - Encryption/Decryption

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

- *AES-CBC (as defined in NIST SP 800-38A) mode*

][9] and cryptographic key sizes [*128-bit*][10] .

5.1.2 User Data Protection (FDP)

FDP_DEC_EXT.1 – Access to Platform Resources

FDP_DEC_EXT.1.1 The application shall restrict its access to [

- *network connectivity*

][11].

FDP_DEC_EXT.1.2 The application shall restrict its access to [

- *no sensitive information repositories*

][12].

FDP_NET_EXT.1 – Network Communications

FDP_NET_EXT.1.1 The application shall restrict network communication to [

- *user-initiated communication for* [
  - User actions: login, logout, issue commands, administrative functions][13]
- *respond to* [device data value update messages sent by SAG][14]

[9] [selection:AES-CBC (as defined in NIST SP 800-38A) mode, AES-GCM (as defined in NIST SP 800-38D) mode, AES-XTS (as defined in NIST SP 800-38E) mode]

[10] [selection: 128-bit, 256-bit]

[11] [selection: no hardware resources, network connectivity, camera, microphone, location services, NFC, USB, Bluetooth,[assignment: list of additional hardware resources]]

[12] [selection: no sensitive information repositories, address book, calendar, call lists, system logs, [assignment: list of additional sensitive information repositories]]

[13] [**assignment**: list of functions for which the user can initiate network communication]

[14] [**assignment**: list of remotely initiated communication]

][15].

FDP_DAR_EXT.1 – Encryption Of Sensitive Application Data

FDP_DAR_EXT.1.1 The application shall [

- *protect sensitive data in accordance with FCS_STO_EXT.1*

][16] in non-volatile memory.

5.1.3 Security Management (FMT)

FMT_MEC_EXT.1 – Supported Configuration Mechanism

Note, this SFR has been modified in accordance with TD0437.

FMT_MEC_EXT.1.1 The application shall [

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

][17].

FMT_CFG_EXT.1 – Secure by Default Configuration

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

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

FMT_SMF.1 – Specification of Management Functions

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

- <u>*User management*</u>][18]

][19].

[15] [selection: no network communication, user-initiated communication for [assignment: list of functions for which the user can initiate network communication], respond to [assignment: list of remotely initiated communication], [assignment: list of application-initiated network communication]]

[16] [selection: leverage platform-provided functionality to encrypt sensitive data, implement functionality to encrypt sensitive data as defined in the EP for File Encryption, protect sensitive data in accordance withFCS_STO_EXT.1, not store any sensitive data]

[17] [selection: invoke the mechanisms recommended by the platform vendor for storing and setting configuration options, implement functionality to encrypt and store configuration options as defined by FDP_PRT_EXT.1 in the PP-Module for File Encryption]

[18] [assignment: list of other management functions to be provided by the TSF]

[19] [selection: no management functions, enable/disable the transmission of any information describing the system's hardware, software, or configuration, enable/disable the transmission of any PII, enable/disable transmission of

5.1.4 Privacy (FPR)

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

FPR_ANO_EXT.1.1 The application shall [

- *not transmit PII over a network*

][20].

5.1.5 Protection of the TSF (FPT)

FPT_API_EXT.1 – Use of Supported Services and APIs

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

FPT_AEX_EXT.1 – Anti-Exploitation Capabilities

FPT_AEX_EXT.1.1 The application shall not request to map memory at an explicit address except for [no exceptions][21].

FPT_AEX_EXT.1.2 The application shall [

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

][22].

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

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

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

any application state (e.g. crashdump) information, enable/disable network backup functionality to [assignment: list of enterprise orFPR_ANO_EXT.1.1 FPT_API_EXT.1.1 commercial cloud backup systems], [assignment: list of other management functions to be provided by the TSF]]

[20] [selection: not transmit PII over a network , require user approval before executing [assignment: list of functions that transmit PII over a network]]

[21] [assignment: list of explicit exceptions]

[22] [selection: not allocate any memory region with both write and execute permissions allocate memory regions with write and execute permissions for only [assignment: list of functions performing just-in-time compilation]]

FPT_TUD_EXT.1 – Integrity for Installation and Update

FPT_TUD_EXT.1.1 The application shall [*provide the ability*][23] to check for updates and patches to the application software.

FPT_TUD_EXT.1.2 The application shall [*provide the ability*][24] to query the current version of the application software.

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

FPT_TUD_EXT.1.4 The application installation package and its updates shall be digitally signed such that its platform can cryptographically verify them prior to installation.

FPT_TUD_EXT.1.5 The application is distributed [*as an additional software package to the platform OS*][25]

FPT_TUD_EXT.2 – Integrity for Installation and Update

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

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

FPT_LIB_EXT.1 – Use of Third Party Libraries

FPT_LIB_EXT.1.1 The application shall be packaged with only [<u>third-party libraries listed in Appendix A</u>][26].

FPT_IDV_EXT.1 – Software Identification and Versions

FPT_IDV_EXT.1.1 The application shall be versioned with [<u>*semantic versioning*</u>][27]

5.1.6 Trusted Path/Channels (FTP)

FTP_DIT_EXT.1 – Protection of Data in Transit

FTP_DIT_EXT.1.1 The application shall [

[23] [selection: provide the ability, leverage the platform]

[24] [selection: provide the ability, leverage the platform]

[25] [selection: with the platform OS, as an additional software package to the platform OS]

[26] [assignment: list of third-party libraries]

[27] [**selection**: SWID tags that comply with minimum requirements from ISO/IEC 19770-2:2015 , [**assignment**: other version information]]

- *invoke platform-provided functionality to encrypt all transmitted data with [HTTPS]*[28]

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

5.2 Security Assurance Requirements

The security assurance requirements for the TOE are reproduced verbatim from the [PPAS].

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] apply to the TOE evaluation.

[28] [selection: HTTPS, TLS, DTLS, SSH]

[29] selection:not transmit any [selection data,sensitive data], encrypt all transmitted [selection: sensitive data, data] with [selection: HTTPS in accordance with FCS_HTTPS_EXT.1, TLS as defined in theTLS Package, DTLS as defined in the TLS Package, SSH as conforming to the Extended Package forSecure Shell] , invoke platform-provided functionality to encrypt all transmitted sensitive data with [selection: HTTPS, TLS, DTLS, SSH] , invoke platform-provided functionality to encrypt all transmitted data with [selection: HTTPS, TLS, DTLS, SSH]]

6 TOE Summary Specification

6.1 Cryptographic support

FCS_RBG_EXT.1

The TOE uses python's built in `os.random` method to access platforms `/dev/urandom` for generating random numbers on the purpose of creating:

- Application token
- Session id
- Websocket connection id
- User password encryption key
- Database credential encryption key

used against CSRF and other potential attacks.

FCS_CKM_EXT.1, FCS_CKM.1

The TOE generates asymmetric cryptographic keys to

- encrypt user passwords hash and the database credentials.
- secure communication channels.

FCS_STO_EXT.1, FMT_MEC_EXT.1., FCS_COP.1
The TOE doesn't store any credential in plain text. TOE hashes user passwords and encrypts the password hash. The encrypted hash is stored in PostgreSQL database. It stores configuration under platforms `/etc` defined by [PP]. The database credential configuration is also encrypted the same way as user passwords.

6.2 User data protection

The TOE protects sensitive data in non-volatile memory according to the requirements in FCS_STO_EXT.1. The TOE restricts its access to network connectivity provided by the platform's hardware resources and does not access any of the platform's sensitive information repositories.

FDP_DEC_EXT.1

The TOE has access only to the platforms network connectivity and has no access to any sensitive information repository.

FDP_NET_EXT.1

The TOE relies on network connectivity in the following cases:

- User access the TOE from client browsers. The end users access the TOE from their browsers. They send different type of requests:
  - User action: Different actions requested by the user for example:
    - Login/Logout
    - Issue commands (for example_view data of different electronic devices, operate devices, view logs)

  - Administrative functions (for example managing users or roles)
- TOE access SA Gateway: Whenever a user operates a device, the TOE sends requests to the SAGateway which then forwards the proper command on the proper protocol to the selected device (outgoing)
- SAG sends updated device data to the TOE (incoming)
- TOE Accessing database either initiated by User action or SAG data update

FDP_DAR_EXT.1: The TOE doesn't store or handle sensitive data other than the ones described in FCS_STO_EXT.1

6.3 Security Management

The Zeta Server comes with default credentials and provides a Web GUI for user administration. The Security Administrator user can log in to the Web GUI using default credentials and has to change its password immediately. The Security Administrator can add or modify users and select user roles for the particular user using the Web GUI.

Users' passwords are hashed and encrypted then stored in the PostgreSQL database.

FMT_MEC_EXT.1: TOE stores its security related configuration under the Linux platforms `/etc` folder. The settings stored here are listed and explained in the Operational User Guidance.

FMT_CFG_EXT.1: After installation of the TOE there is only one user created who can login from its client's browser using the default credentials (described in the Operational User Guidance) and the only operation available for the user is to change its password. After changing the password all operation will become available and the default password cannot be used anymore.

Modification of program files (which are basically script files) is not possible because all files are not world-writeable. The server containing the TOE has a restricted access anyway and in most cases, it operates on a private network without any publicly available interface.

FMT_SMF.1: According to management functions the user is able to manage users (create, disable, modify, assign user roles).

6.4 Privacy

FPR_ANO_EXT.1: The TOE does not handle personally identifiable information (PII) nor locally or on network.

6.5 TSF Protection

The TOE is compatible with its host OS platform when that is configured in a secured manner, using SELinux.

The TOE uses a well-defined set of platform APIs and third-party libraries.

The TOE provides the ability for the Installer to check its version and if an update is available. Internet connection needed for the latter.

Updates are delivered in formats appropriate for the platform on which the TOE is hosted. It is digitally signed, and the signature is validated prior to installation. Installing an update of the application removes all previously installed files except configuration and audit/log files.

FPT_API_EXT.1: The list of platform API used by the TOE can be found in Appendix B.

TOE is built of python scripts and libraries which does not need to be compiled by design. The scripts are read and processed line by line so there is no need for ASLR flags during compilation. On the fly processing also guarantees that there are no memory mappings at an explicit and consistent address.

FPT_TUD_EXT.1:

1.1 The application provides the ability to check for update. It can be done by the Installer running a script. Internet connection needed for this operation and detailed description will be included in the User Guidance document.

1.2 The application leverages the platform to query the current version of the application. The installed packages version matches the version of the application by definition.

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

The application can be downloaded from the developer's site and the update can be done manually after stopping the application. Detailed description will be included in the User Guidance document.

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

1.5 The application is distributed as an additional software to the platform OS.

FPT_TUD_EXT.2

The application is 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.

The configuration files and logs will be preserved. Their location will be described in the User Guidance document.

FPT_LIB_EXT.1

The TOE is packaged only with third parties listed in Appendix A.

FPT_IDV_EXT.1

The TOE using semantic versioning which works as follows:

Given a version number MAJOR.MINOR.PATCH, increment the:

1. MAJOR version when you make incompatible API changes,
2. MINOR version when you add functionality in a backwards compatible manner, and
3. PATCH version when you make backwards compatible bug fixes.

Additional labels for pre-release and build metadata are available as extensions to the MAJOR.MINOR.PATCH format.

More about semantic versioning can be found at https://semver.org/.

6.6 Trusted Channel/Path

The TOE encrypts data in transit between itself and clients as well as the Substation Automation Gateway using HTTPS and Secure Websocket connections. The TOE relies on the platform to implement HTTPS encryption.

FTP_DIT_EXT.1

A built-in python functionality is used to provide SSL context for “websockets” third-party library. This library builds up the connection between SAG and the TOE.

Between the TOE and the User “uvicorn” third-party library is used to secure the connection.

Both solutions use the Linux platform’s OpenSSL to implement the channel encryption.

6.7 Timely Security Updates

The developers of the TOE are keeping an eye on technologies, tools, platform they used during the development or the operational use of the TOE. If there are any vulnerabilities, bugfixes or a new version of any used module is found, they analyze it and decide whether the TOE should be updated or not. In case of a vulnerability is found developer notifies all customers immediately and tries to fix the problem within 14 working days.

Customers (end users) of the TOE can also report bugs they find in the TOE. They can send bug report to support@prolan-power.com. If the bug contains sensitive description of a possible vulnerability or logs or anything else, customer must use the Developers public key to encrypt the sensitive information. Support will ask for environment details, configuration files, version of the TOE and application logs. If it found out that it is related to a bug, developers fix it.

Either way a new release of the TOE is prepared. New releases (no matter if it’s a bugfix or new features developed) are tested by the testing team running all the tests of the TOE.

If all tests passed a new release is made available on Prolan Power's website. The new release is only available for the customers after authenticating themselves.

Customer can always check for updates using a script described in [AGD] but even without checking for updates, Developer always notifies customer whenever a new release is available so they can download the new version and update TOE based on [AGD] update description.

7 Glossary of Terms

The following definitions are used in this document:

Term	Meaning
Application (app)	Software that runs on a platform and performs tasks on behalf of the user or owner of the platform, as well as its supporting documentation. The terms TOE and application are interchangeable in this document.
Application Programming Interface (API)	A specification of routines, data structures, object classes, and variables that allows an application to make use of services provided by another software component, such as a library. APIs are often provided for a set of libraries included with the platform. Credential Data that establishes the identity of a user, e.g. a cryptographic key or password.
Data Execution Prevention (DEP)	An anti-exploitation feature of modern operating systems executing on modern computer hardware, which enforces a non-execute permission on pages of memory. DEP prevents pages of memory from containing both data and instructions, which makes it more difficult for an attacker to introduce and execute code. Developer An entity that writes application software. For the purposes of this document, vendors and developers are the same. Mobile Code Software transmitted from a remote system for execution within a limited execution environment on the local system. Typically, there is no persistent installation and execution begins without the user's consent or even notification. Examples of mobile code technologies include JavaScript, Java applets, Adobe Flash, and Microsoft Silverlight.
Operating System (OS)	Software that manages hardware resources and provides services for applications. In our case the OS is CentOS 7 Linux.
Personally Identifiable Information (PII)	Any information about an individual maintained by an agency, including, but not limited to, education, financial transactions, medical history, and criminal or employment history and information which can be used to distinguish or trace an individual's identity, such as their name, social security number, date and place of birth, mother's maiden name, biometric records, etc., including any other personal information which is linked or linkable to an individual. The TOE doesn't store or use any PII.
Platform	The environment in which application software runs. The platform can be an operating system, hardware environment, a software based execution environment, or some combination of these. These types of platforms may also run atop other platforms. In our case the platform is CentOS 7 Linux.
Sensitive Data	Sensitive data may include all user or enterprise data or may be specific application data such as emails, messaging, documents, calendar items, and contacts. Sensitive data must minimally include PII, credentials, and keys.
Vendor	An entity that sells application software. For purposes of this document, vendors and developers are the same. Vendors are responsible for maintaining and updating application software.
Security Administrator	Role. Security Administrator is the Admin according to the PP terminology. This role has full access to the application and the platform as well.

Installer	Role. Users assigned to this role are responsible for the installation of the TOE.
Engineer	Role. Users assigned to this role are able to update the Substation Model
Operator	Role. The end users of the product accessing it via their browser having the less privileges.
Substation Model	This is different in all installations of the TOE. This describes and defines the current substations, devices with all their parameters.

8 Acronyms

Acronym	Meaning
TOE	Target of evaluation which is Zeta Server
GUI	Graphical User Interface
SAGateway	Substation Automation Gateway
TSF	TOE Security Functional Interface
SELinux	Security-Enhanced Linux
CC	Common Criteria
PP	Protection Profile
EAL	Evaluation Assurance Level
ASLR	Address Space Layout Randomization
CSRF	Cross Site Request Forgery

9 Appendix A - Third Party Libraries used by the TOE

Server libraries:
python 3.6.8
python3-devel
python36-pycryptodomex 3.9.7-1

Pyhton libraries:
aiofile-3.3.3
aiofiles-0.6.0
attrs-20.3.0
caio-0.6.3-py3.6
click-7.1.2
dataclasses-0.8
dill-0.3.3
fastapi-0.62.0
fastapi_utils-0.2.1
greenlet-1.0.0
h11-0.12.0
importlib_metadata-3.10.0
Jinja2-2.11.3
jsonschema-3.2.0
MarkupSafe-1.1.1
pip-9.0.3
psycopg2-2.8.6-py3.6
pycrypto-2.6.1-py3.6
pydantic-1.7.3
pyfunctional-1.4.3
pyrsistent-0.17.3-py3.6
python_multipart-0.0.5-py3.6
PyYAML-3.13-py3.6
setuptools-39.2.0
six-1.15.0
SQLAlchemy-1.4.4
starlette-0.13.6
tabulate-0.8.9
typing_extensions-3.7.4.3
uvicorn-0.12.3
websockets-8.1 zipp-3.4.1

Javascript libraries:
material-ui/core 4.5.0
material-ui/icons 4.4.3
material-ui/lab 4.0.0-alpha.46
rehooks/component-size 1.0.3
thebiltheory/usebreakpoints 1.1.2

classnames 2.2.6
flux 3.1.3
ramda 0.27.0
react 16.9.0
react-device-detect 1.12.1
react-dom 16.9.0
react-icons 3.10.0
react-json-viewer 2.1.0
react-new-window 0.1.2
react-resize-detector 4.2.1
react-scripts 3.1.1
react-scrollbar-size 2.1.0
react-virtualized 9.21.1
styled-components 5.0.1
universal-cookie 4.0.3
use-breakpoint 1.1.2
axios 0.21.1

10 Appendix B – Platform API used by the TOE

accept4, access,arch_prctl, bind, brk, clock_getres, clone, close, connect, dup, dup2, epoll_create1, epoll_ctl, epoll_wait, eventfd2, execve, exit_group, fcntl, fstat, futex, getcwd, getdents, getdents64, getegid, geteuid, getgid, getpeername, getrandom, getrlimit, getsockname, getsockopt, getuid, ioctl, io_getevents, io_submit, listen, lseek, lstat, mmap, mprotect, mremap, munmap, open, openat, pipe2, poll, read, readlink, recvfrom, recvmsg, rt_sigaction, rt_sigprocmask, rt_sigreturn, sendto, set_robust_list, setsockopt, set_tid_address, sigaltstack, socket, socketpair, stat, statfs, uname, wait4, write

11 Bibliography

[CC_P1] Common Criteria, Part 1: Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model, Version 3.1, Revision 5, April 2017, CCMB-2017-04-001

[CC_P2] Common Criteria, Part 2: Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Requirements, Version 3.1, Revision 5, April 2017, CCMB-2017-04-002

[CC_P3] Common Criteria, Part 3: Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements, Version 3.1, Revision 5, April 2017, CCMB-2017-04-003

[CEM] Common Methodology for Information Technology Security Evaluation, Evaluation Methodology, Version 3.1, Revision 5, April 2017, CCMB-2017-04-004

[PP] Protection Profile for Application Software, Version 1.3, 1 March 2019

[IEC 62351 -8] Power systems management and associated information exchange - Data and communications security - Part 8: Role-based access control for power system management

[AGD] Guidance Documentation Zeta Server v1.1.1, v0.4, 2021.06.18.