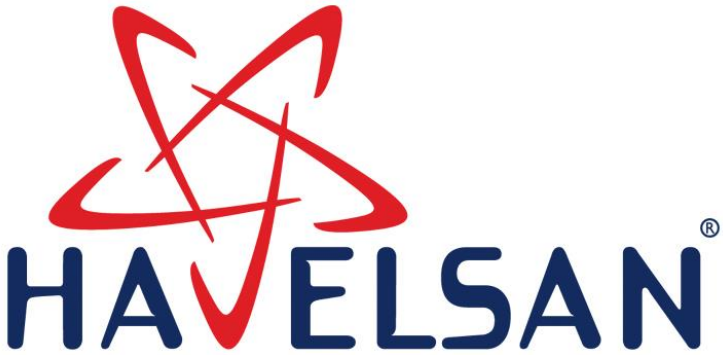

HAVELSAN GÖZCÜ v1.0.3

Security Target

Version No: 1.16

VERSION HISTORY

Version No	Reason for Change	Author	Release Date
1.0	First Draft	Cansu Tetik Barış Kaya	18.10.2017
1.1	First Review	Cansu Tetik Barış Kaya	20.11.2017
1.2	Second Review	Cansu Tetik Barış Kaya	23.11.2017
1.3	Third Review	Barış Kaya	05.12.2017
1.4	Fourth Review	Barış Kaya	25.12.2017
1.5	TOE type is clarified, tables are fixed	Barış Kaya	18.01.2018
1.6	Security requirements are re-fined	Barış Kaya	09.08.2018
1.7	Cryptographic operations are refined	Barış Kaya	02.11.2018
1.8	7.1.6 is refined	Barış Kaya	05.11.2018
1.9	SFRs are updated	Barış Kaya	16.11.2018
1.10	SFRs are updated	Barış Kaya	10.01.2019
1.11	Branding changes	Barış Kaya	23.01.2019
1.12	Version updated	Barış Kaya	20.11.2019
1.13	Branding changes	Barış Kaya	19.12.2019
1.14	Version updated	Barış Kaya	20.10.2020
1.15	Fifth Review	Sedat Akleylek	02.02.2021
1.16	ST Introduction refined	Fatih Furkan Bayar	08.07.2021

GLOSSARY

GÖZCÜ : Security information and event management

CONTENTS

LIST OF TABLES

1. ST INTRODUCTION

1.1. SECURITY TARGET & TOE REFERENCE

ST Title: HAVELSAN GÖZCÜ Security Target

ST Reference: HAVELSAN GÖZCÜ Security Target, version 1.15, Havelsan, 02.02.2021

TOE Reference: HAVELSAN GÖZCÜ v1.0.3

CC Conformance: Common Criteria for Information Technology Security Evaluation, Version 3.1 (Revision 5)

Assurance Level: EAL4+ (ALC_FLR.1)

Keywords: GÖZCÜ, Logging, SIEM

1.2. TOE OVERVIEW

TOE is a web application that manages a system that collects security and event logs from applications, products appliances of organizations. The collected logs then get correlated, achieved and timestamped. Working on these logs, the system creates near real time alerts and flexible reports. TOE visualizes collected logs, alerts and reports. Authorized users can define custom alerts and reports. TOE is capable of managing authorization. Unauthorized actions will be denied and logged as well as user interactions. Additionally, TOE provides configuration management interface, network topology visualization, archiving and supports high level REST API integration with third party applications.

Reporting

Reporting operations such as filtering, grouping, aggregation (sum, average, count etc..) could be conducted to the previously collected logs by the system. System provides report outputs in PDF, HTML and CSV formats. Reports could be retrieved in scheduled manner. Daily, weekly or monthly reports could be generated from the system. Apart from predefined reports user can create custom reports for their needs.

Generating Network Topologies

Within the scope of the system, the topology for given organization could be generated and observed by the users. Network hierarchies, domain and user groups provide information for the topology.

Case Management

Alarms that require human interaction can be assigned to users and the status of the cases could be tracked.

Authorization

With the capability of configuration, the actions which the users can conduct are controlled by the system. Unauthorized actions are prevented with this feature.

Auditing

Critical actions on the system are recorded for internal auditing.

Archiving

System supports timestamping which ensures that logs have not been changed since then.

Managing Configuration

Configuration can be saved, exported and imported easily.

Metric Collection

The system establishes a health check mechanism via the connected agents.

Integration

REST API support enables integration with third party applications.

1.2.1. USAGE & MAJOR SECURITY FEATURES OF A TOE

Usage: HAVELSAN GÖZCÜ provides centrally collection, correlation, inquiry and alarm generation of records created by IT infrastructure components of organizations.

The following features are the major security functionality of the TOE;

- **Audit:** TOE will generate audit logs in order to provide accountability for the administrators

and users.

- **Cryptographic Support:** TOE should provide hashing mechanisms for securely storing user passwords. No keys are required for this action.
- **Identification, Authentication and Authorization:** TOE will successfully identify, authenticate and authorize its users.
- **Data Protection:** TOE provides confidentiality and integrity of user and TSF data during import/export of data to/from third parties.
- **Security Management:** TOE will manage the security attributes and user roles.

1.2.2. TOE TYPE

TOE is a management web application that manages Security Information and Event Management (GÖZCÜ) software solution which is used in the process of identifying, monitoring, recording and analysing security events or incidents within a real- time IT environment using agents in targeted computers, event forwarding (syslog) and correlation engine in the management side.

Management Server component consists of a GUI (Graphical User Interface) and the back-end functionalities to manage operations between components of the system.

Data Engine stores logs collected by the agents together with logs retrieved through event forwarding (syslog) and correlates these logs to identify security events. TOE can access stored logs and visualizes events with infographics by using filters defined by authorized users.

TOE can be delivered to customers either in a dedicated hardware or as virtual appliances. The delivery and deployment of TOE components are achieved through Docker containers depending on the TOE configuration. In the dedicated hardware configuration, all docker containers are located on the same machine whereas in virtual appliance configuration, docker containers are distributed among machines.

1.2.3. NON TOE HARDWARE/ SOFTWARE/ FIRMWARE

The TOE operates in a web server environment. The following elements are not evaluated but required to achieve its main goal of TOE.

- minimum Ubuntu version 14.04
- Minimum 16 GB RAM per server.
- Minimum 4 core processors for each server.
- At least half of the servers should have at least 1000 GB of HDD.
- Servers should have minimum 300 GB of HDD.
- MySQL DB 5.7.18
- JRE 8 should be installed on management server.
- Hadoop 2.7.3.
- Flink 1.3.1
- Drill 1.10.0
- Redis v3.2.9
- Kafka v2.11-0.9.0.1
- Zookeeper v3.4.9
- Elasticsearch v5.6.2

1.3. TOE DESCRIPTION

1.3.1. TOE PHYSICAL SCOPE

TOE consists of GUI (Graphical User Interface) and the back-end functionalities in a single application. TOE is supplied as a software product installed on Linux-based platforms. TOE will be evaluated PC platform under the following Linux operating systems: Ubuntu 14.04, Ubuntu 16.04, Ubuntu 18.04.

Additionally, TOE will be installed on client workstation and client will be required to connect log sources to TOE.

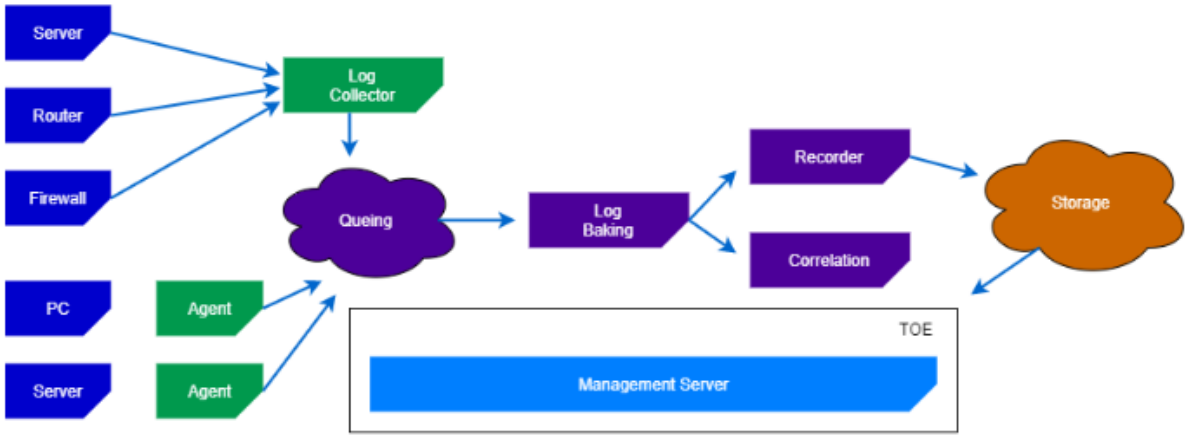

Figure 2. TOE Physical Scope

1.3.2. TOE LOGICAL SCOPE

Log Generation:

The TSF generates audit logs that consist of various auditable events. Date and time of events, usernames, and events taken by the authorized users are recorded.

Cryptographic Key Management and Operations:

Hashing action is taken by the TSF for storing and authenticating users' passwords. Hashing action does not require any additional keys.

User Login and Authentication:

When a user issues a request to the TOE to access a protected resource, the TOE requires that the user (being a User, Administrator) identify and authenticate themselves before performing any action on

behalf of the user. Identification and Authentication is required to ensure that users are associated with the proper security attributes (e.g. identity, group, roles, and security or integrity levels). Once the user attempts administrator defined unsuccessful authentication, his/her status is disabled and (s)he will wait the status is enabled again by administrator. Users' passwords are controlled according to the organizational password quality requirements.

Protection of Data

The access control function permits a user to access a protected resource only if a user ID or role of the user is given permission to perform the requested action on the resource by Administrator. On the other hand, Administrators of the TOE can perform assigning the privileges, modify his/her own authentication data, users' password and other information.

User Roles and Security Rules

Only administrators are allowed to manage and configure security functions. Administrators can assign access privileges to users by user levels based on the functions or resources that they are allowed to perform. Additional functionalities such as modifying access privileges and unlocking password for users are also accessible by authorized administrator. Predefined roles are maintained and can be assigned to users.

Connection via Trusted Path

Users' sessions are forced to be established through trusted path.

TOE Access

The session inactivity of users exceeds 30 minutes, the authorized users are returned to the login page. The users are also able to terminate their own sessions.

1.3.3. ROLE GROUPS

The TOE supports the following roles, which are subjects of the "User Access Control Policy":

a) User (Default)

b) Admin Created Role Groups (Authorized User)

c) Administrator

The TOE implements an access control SFP named "User Access Control SFP". Associated roles with this SFP are described below:

a) User(Default): Role has limited access to TOE. It is actually the most basic role in the TOE and it does not have any privileges. Initially created users have no responsibility unless they are associated

with an admin created role group.

b) Admin Created Role Groups: Role groups in this document are created by administrator and users are binded to role groups to have authorizations so users in role groups with appropriate authorizations are referred as Authorized User in this document. These roles allow users to have different organisational responsibilities.

c) Administrator Role: An administrator in the TOE has every authorization. It can create users, role groups, modify role groups' authorizations and associate users with role groups. Administrators can execute management functions via User Interface. In this document administrator is also referred as Administrator User, Authorised Administrator and Admin.

2. CONFORMANCE CLAIMS

2.1. CC CONFORMANCE CLAIM

This Security Target claims conformance to

- Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; Version 3.1, Revision 5, CCMB-2017-04-001, [1]
- Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components; Version 3.1, Revision 5, CCMB-2017-04-002, [2]
- Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Components; Version 3.1, Revision 5, CCMB-2017-04-003, [3]

referenced hereafter as [CC].

This Security Target claims the following CC conformance:

- part 2 conformant
- part 3 conformant
- evaluation assurance level (EAL) 4+

2.2. PP CLAIM

- This ST does not claim conformance to any protection profile.

2.3. PACKAGE CLAIM

This ST is conforming to assurance package EAL4 augmented with ALC_FLR.1 defined in CC Part 3 (CC Part 3).

3. SECURITY PROBLEM DEFINITION

3.1. THREATS

T.UNAUTHORIZED_ACCESS: A malicious user may gain unauthorized access to the TOE and change the TOE configuration.

Threat agent: A malicious user

Assets: the TOE configuration

Adverse action: change the TOE configuration in such a way to result security flaws

T.EAVES_DROPPING: Malicious Users could gain the valuable information (passwords and enterprise data) of authorized administrator by sniffing the traffic between waf and web application.

Threat agent: a malicious user

Assets: passwords and enterprise data

Adverse action: gain the valuable information by sniffing

T.NO_ROUTE : A malicious user may cause the TOE to lose connection on the network layer to the source of its enforcement policies, adversely affecting data collection.

Threat agent: A malicious user

Assets: access control behaviors

Adverse action: lose connection to the source of its enforcement policies

3.2. ORGANIZATIONAL SECURITY POLICIES

There are no Organizational Security Policies for the application.

3.3. ASSUMPTIONS

The assumptions are described in below;

A.ADMIN It is assumed that authorized administrator who is responsible to install, configure and operate the TOE and the IT entities in the operational environment of the TOE are experienced,

trained and meet the security conditions.

A.PROTECT It is assumed that all hardware within the environment, including network and peripheral devices, has been approved for the transmitting of secure data. Each of these appliance configurations is securely managed by administrators to provide protection of secured data in terms of its confidentiality and integrity.

A.NO_GENERAL_PURPOSE It is assumed that there are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.

A.PHYSICAL Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the environment.

A.TIME_SERVER It is assumed that trusted time server provides reliable time information.

4. SECURITY OBJECTIVES

In this section part-wise solutions are given against the security problem defined in Part 3.

4.1. SECURITY OBJECTIVES FOR THE TOE

The security objectives for the TOE are described in below;

O.AUTH : TOE will successfully identify and authenticate its users before allowing any actions.

O.PROTECTED_COMMUNICATION : The TOE will provide protected communication channels for administrators, other parts of a distributed TOE, and authorized IT entities.

O.AUDIT : The TOE will provide the capability to generate audit data and send those data to an external IT entity.

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.

OE.NETWORK Those responsible for the TOE must ensure that appropriate network layer protection, that there is a firewall in place that only permits access through required ports for external users to access the web- server.

OE.SEC_ENV Operational environment of the TOE shall ensure physical and environmental security of the TOE. Unauthorized access shall be restricted and all components in the operational environment shall be secured. Only specifically authorized people shall be allowed to access critical components.

OE.CREDEN Those responsible for the TOE must ensure that all access credentials, such as passwords or other authentication information, are protected by the users (by complying with organizational policies and procedures disallowing disclosure of user credential information) in a manner which maintains organizational IT security objectives.

OE.ADMIN Administrator is non-hostile, well-trained, and follow all user guidance, installation

guidance and configuration guidance

OE.PHYSICAL : Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.

OE.NO_GENERAL_PURPOSE : There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.

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. TIME Operational environment shall allow access to a reliable NTP server.

4.3. SECURITY OBJECTIVES RATIONALE

Rationale tables of Threats, Assumptions and Security Objectives are given in Table 1:

Threats-Objective	O.AUTH	O.PROTECTED_COMMUNICATION	O.AUDIT	OE.NETWORK	OE.SEC_ENV	OE.CREDEN	OE.ADMIN	OE.PHYSICAL	OE.NO_GENERAL_PURPOSE	OE. PLATFORM	OE.TIME
T.UNAUTHORIZED_ACCESS	X		X		X	X					
T.EAVES_DROPPING		X			X			X		X	
T.NO_ROUTE			X	X							
A.ADMIN							X				
A.PROTECT					X			X		X	
A.NO_GENERAL_PURPOSE									X		
A.PHYSICAL								X			
A.TIME_SERVER											X

Table 1 Threats vs. Objectives table

Threat, Assumption, or OSP	Security Objectives	Rationale
T. UNAUTHORIZED_ACCESS	**O. AUTH** **O.AUDIT** **OE.CREDEN** **OE.SEC_ENV**	The threat T. UNAUTHORIZED_ACCESS is countered by O. AUTH, OE.CREDEN, O.AUDIT and OE.SEC_ENV where each users of the TOE will be successfully authenticated before any actions and TOE will generate audit logs to review user actions. TOE will provide security management functionality for user management.
T. EAVES_DROPPING	**O. PROTECTED_COMMUNICATION** **OE.SEC_ENV** **OE.PHYSICAL** **OE.PLATFORM**	The threat T. EAVES_DROPPING will be countered by OE.PHYSICAL , OE.SEC_ENV, OE.PLATFORM and O.PROTECTED_COMMUNICATION where O.PROTECTED_COMMUNICATION ensures that TOE will communicate via secure channels and information flow to third parties will be under security control , OE.PHYSICAL ensures that there is enough physical security provided to protect TOE from physical interactions with malicious users , OE.SEC_ENV which ensures that all hardware has been approved for the transmitting of secure data and managed securely and OE.PLATFORM ensures that TOE relies upon a trustwor-

		thy computing platform that has its own security precautions.
T. NO_ROUTE	**O. AUDIT** **OE.NETWORK**	The threat T. NO_ROUTE will be countered by O. AUDIT and OE.NETWROK where TOE will log every action and there will be a firewall to protect TOE from malicious activities coming from network layer.
A. ADMIN	**OE. ADMIN**	This assumption is addressed by OE. ADMIN , which ensures that Administrators are experienced, trained and meet the security conditions.
A. PROTECT	**OE.SEC_ENV** **OE.PHYSICAL** **OE.PLATFORM**	This assumption is addressed by OE.SEC_ENV, which ensures that all hardware has been approved for the transmitting of secure data and managed securely., and OE.PHYSICAL , which ensures that TOE's physical environment has a trustworthy physical security and OE.PLATFORM ,which ensures that TOE relies upon a trustworthy computing platform that has its own security precautions.

A. PHYSICAL	**OE. PHYSICAL**	This assumption is addressed by OE.PHYSICAL which ensures that TOE's physical environment has a trustworthy physical security.
A.NO_GENEREAL_PURPOSE	**OE.NO_GENERAL_PURPOSE**	This assumption is addressed by OE.NO_GENERAL_PURPOSE which ensures that there are no general-purpose computing capabilities on TOE's environment.
A.TIME_SERVER	**OE.TIME**	This assumption is addressed by OE.TIME , which ensures that a connection to a trusted time server is provided.

Table 2 Security Objectives Rationales

5. Extended Components Definition

There is not any extended components definition within this Security Target.

6. Security Requirements

6.1. Security Functional Requirements

This part of the ST defines the detailed security requirements that shall be satisfied by the TOE. The statement of TOE security requirements shall define the functional and assurance security requirements that the TOE needs to satisfy in order to meet the security objectives for the TOE. The CC allows several operations to be performed on functional requirements; refinement, selection, assignment, and iteration are defined in Section 8.1 of Common Criteria Part1 [17]. The following operations are used in the ST.

The **refinement** operation is used to add detail to a requirement, and thus further restricts a requirement. Refinements of security requirements are denoted in such a way that added words are in **bold text** and removed are ~~crossed out~~.

The **selection** operation is used to select one or more options provided by the CC instating a requirement. Selections, having been made, are denoted as <u>underlined text</u>.

The **assignment** operation is used to assign a specific value to an unspecified parameter, such as the

length of a password. Assignments are denoted by *italicized* text. If an assignment is done under a selection it will be denoted by ***italicized and bold*** text.

The **iteration** operation is used when a component is repeated with varying operations. Iteration is denoted by showing a slash "/", and the iteration indicator after the component identifier.

Requirement Class	Requirement Component
FAU: SECURITY AUDIT	FAU_GEN.1 : Audit Data Generation
	FAU_GEN.2 : User Identity Association
FCS: CRYPTOGRAPHIC SUPPORT	FCS_COP.1 : Cryptographic operation
FDP: USER DATA PROTECTION	FDP_ACC.2 : Complete Access Control
	FDP_ACF.1 : Security attribute based access control
FIA: IDENTIFICATION AND AUTHENTICATION	FIA_AFL.1 : Authentication Failure Handling
	FIA_SOS.1 : Verification of Secrets
	FIA_UAU.2 : User authentication before any action
	FIA_UAU.7 : Protected authentication feedback
	FIA_UID.2 : User identification before any action
	FIA_ATD.1 : User attribute definition
FMT: SECURITY MANAGEMENT	FMT_MSA.1: Management of security attributes
	FMT_MSA.3: Static attribute initialization
	FMT_SMF.1 : Specification of management functions
	FMT_SMR.1 Security Roles
FTA: TOE ACCESS	FTA_SSL.1: TSF-initiated session locking
	FTA_SSL.4: User-initiated termination
FTP: TRUSTED PATHS	FTP_TRP.1 : Trusted Paths

6.1.1. CLASS FAU: SECURITY AUDIT

6.1.1.1.FAU_GEN.1 Audit Data Generation

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shutdown of the audit functions;

b) All auditable events for the [not specified] level of audit; and

c) [none**].**

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the ST, **[***related java class, related java class method, line number, username, api name***].**

6.1.1.2.FAU_GEN.2 User Identity Association

FAU_GEN.2.1 For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

6.1.2. CLASS FCS: CRYPTOGRAPHIC SUPPORT

6.1.2.1.FCS_COP.1 Cryptographic operation

FCS_COP.1.1 The TSF shall perform **[***hashing of passwords before storing them in database***]** in accordance with a specified cryptographic algorithm **[***bcrypt***]** and cryptographic key sizes [*none**] that meet the following: **[***none***].**

Application note:

*: Hashing algorithms do not require any additional keys.

CLASS FDP: DATA PROTECTION

6.1.3.1.FDP_ACC.2 Complete Access Control

FDP_ACC.2.1 The TSF shall enforce the [*User Access Control SFP*] on [

Subjects:

- *User*
- *User role group (Admin defined groups)*
- *Administrator*

Objects:

- *User interface web pages' access privileges (that provide access to objects including list below:*

  GÖZCÜ Logs (Not related to FAU_GEN.1 or FAU_GEN.2)

  Parameters

  Configuration Meta Data

  User Role Settings

  Menu Items

  Report Data

  Statistics Data

  Data Parsing Metadata

  Data Rule Metadata

  Data Classification metadata

  Data processing request data)

] and all operations among subjects and objects covered by the SFP.

FDP_ACC.2.2 The TSF shall ensure that all operations between any subject controlled by the TSF and any object controlled by the TSF are covered by an access control SFP.

6.1.3.2.FDP_ACF.1 Security Attribute Based Access Control

FDP_ACF.1.1 The TSF shall enforce the [*User Access Control SFP*] to objects based on the following: [

-User role settings assigned to administrator defined role groups

-Users that are assigned to an administrator defined user role

-Administrators have all privileges.

].

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [

- Administrator can take every action over every object

- Administrator can define User roles

- Administrator can modify user role settings

- Administrator can associate/deassociate User role groups with Users over application.

- User can take actions over objects based on their user role group's settings

].

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [*none*].

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [*none*].

6.1.4. CLASS FIA: IDENTIFICATION AND AUTHENTICATION

6.1.4.1.FIA_AFL.1 Authentication Failure Handling

FIA_AFL.1.1 The TSF shall detect when **[***an administrator configurable positive integer within [greater or equal to **3**]***]** unsuccessful authentication attempts occur related to **[***login***].**

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been **[**met**],** the TSF shall **[***block user for an admin defined time interval that is greater than or equal to 5 minutes***].**

6.1.4.2.FIA_ATD.1 User attribute definition

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users: **[***username, password, email, role group, phone number***].**

6.1.4.3.FIA_SOS.1 Verification of Secrets

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet [

password must contain;

-at least 8 characters

-at least 1 lowercase character

-at least 1 uppercase character

-at least 1 numeric character

-at least 1 special character (!@#$%^&()-_=+,.?\/:;{}[]~)*

].

6.1.4.4.FIA_UAU.2 User authentication before any action

FIA_UAU.2.1 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

6.1.4.5.FIA_UAU.7 Protected authentication feedback

FIA_UAU.7.1 The TSF shall provide only *[showing of asterisk characters on password field]* to the user while the authentication is in progress.

6.1.4.6.FIA_UID.2 User identification before any action

FIA_UID.2.1 The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

6.1.5. CLASS FMT: SECURITY MANAGEMENT

6.1.5.1.FMT_MSA.1 Management of Security Attributes

FMT_MSA.1.1 The TSF shall enforce the **[***User access control SFP]* to restrict the ability to *[*modify, delete, ***[enable user, disable user, change group]****]* the security attributes **[** *password, phone number, email, user group***]** to **[***administrator and authorized user***].**

6.1.5.2. FMT_MSA.3 Static Attribute Initialization

FMT_MSA.3.1 The TSF shall enforce the **[***User access control SFP***]** to provide **[**restrictive**]** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the **[***administrators***]** to specify alternative initial values to override the default values when an object or information is created.

6.1.5.3.FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions: [*create, delete, modify, and read security attributes defined in FIA_ATD.1*].

6.1.5.4.FMT_SMR.1 Security Roles

FMT_SMR.1.1 The TSF shall maintain the roles [*administrator and admin created role groups*].

FMT_SMR.1.2 The TSF shall be able to associate users with roles.

CLASS FTA: TOE ACCESS

6.1.6.1.FTA_SSL.1 TSF-initiated session locking

FTA_SSL.1.1 The TSF shall lock an interactive session after [*30 minutes*] by:

a) clearing or overwriting display devices, making the current contents unreadable;

b) disabling any activity of the user's data access/display devices other than unlocking the session.

FTA_SSL.1.2 The TSF shall require the following events to occur prior to unlocking the session: **[***relog-in***]**.

6.1.6.2.FTA_SSL.4 User-initiated termination

FTA_SSL.4.1 The TSF shall allow user-initiated termination of the user's own interactive session.

CLASS FTP: TRUSTED PATHS

6.1.7.1.FTP_TRP.1 Trusted Path

FTP_TRP.1.1 The TSF shall provide a communication path between itself and [remote and local] users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from [modification and ***deletion***].

FTP_TRP.1.2 The TSF shall permit [remote and local users] to initiate communication via the trusted path.

FTP_TRP.1.3 The TSF shall require the use of the trusted path for [initial user authentication].

6.2. SECURITY ASSURANCE REQUIREMENTS

Assurance Class	Assurance Component
ADV: Development	ADV_ARC.1 – Security architecture description
	ADV_FSP.4 – Complete Functional Specification
	ADV_IMP.1 – Implementation Representation of the TSF
	ADV_TDS.3 – Basic Modular Design
AGD: Guidance Documents	AGD_OPE.1 – Operational user guidance
	AGD_PRE.1 – Preparative procedures
ALC: Life-cycle Support	ALC_CMC.4 – Production support, acceptance procedures automation
	ALC_CMS.4– Problem tracking CM coverage
	ALC_DEL.1 – Delivery procedures
	ALC_DVS.1 – Identification of security measures
	ALC_LCD.1 – Developer defined life-cycle model
	ALC_TAT.1 – Well defined development tools
	ALC_FLR.1 – Basic Flaw Remediation

ASE: Security Target Evaluation	ASE_CCL.1 – Conformance claims
	ASE_ECD.1 - Extended components definition
	ASE_INT.1 – ST Introduction
	ASE_OBJ.2 – Security objectives
	ASE_REQ.2 – Derived security requirements
	ASE_SPD.1 – Security problem definition
	ASE_TSS.1 – TOE summary specification
ATE: Test	ATE_COV.2 – Analysis of coverage
	ATE_DPT.1 – Testing: basic design
	ATE_FUN.1 – Functional testing
	ATE_IND.2 – Independent testing – sample
AVA: Vulnerability Assessment	AVA_VAN.3 –Focused vulnerability analysis

Table 3 Security Assurance Requirements

6.3. SECURITY REQUIREMENTS RATIONALE

6.3.1. SECURITY FUNCTIONAL REQUIREMENTS DEPENDENCY RATIONALE

SFRs	Dependency	Fulfilled by security requirements in this ST
FAU_GEN.1	FPT_STM.1	Time stamps will be provided by operational environment.
FAU_GEN.2	FAU_GEN.1 FIA_UID.1	full-fit(FIA_UID.2 exists and it is hierarchical to FIA_UID.1)
FCS_COP.1	[FCS_CKM.1] FCS_CKM.4	Crypto keys are out sourced.
FDP_ACC.2	FDP_ACF.1	full-fit
FDP_ACF.1	FDP_ACC.1 FMT_MSA.3	full-fit
FIA_AFL.1	FIA_UAU.1	full-fit (FIA_UAU.2 exists and it is hierarchical to FIA_UAU.1)
FIA_ATD.1	-	-
FIA_SOS.1	-	-
FIA_UAU.2	FIA_UID.1	full-fit (FIA_UID.2 exists and it is hierarchical to FIA_UID.1)
FIA_UAU.7	FIA_UAU.1	full-fit (FIA_UAU.2 exists and it

		is hierarchical to FIA_UAU.1)
FIA_UID.2	-	-
FMT_MSA.1	[FDP_ACC.1] FMT_SMR.1 FMT_SMF.1	full-fit(FDP_ACC.2 exists and it is hierarchical to FDP_ACC.1)
FMT_MSA.3	FMT_MSA.1 FMT_SMR.1	full-fit
FMT_SMF.1	-	-
FMT_SMR.1	FIA_UID.1	full-fit (FIA_UID.2 exists and it is hierarchical to FIA_UID.1)
FTA_SSL.1	FIA_UAU.1	Full-fit (FIA_UAU.2 exists and it is hierarchical to FIA_UAU.1)
FTA_SSL.4	-	-
FTP_TRP.1	-	-

Table 4 Security Functional Requirements Dependency Rationale

6.3.2. SECURITY FUNCTIONAL REQUIREMENTS RATIONALE

Objectives	SFRs	Rationale
O. AUTH	**FIA_UAU.2,** **FIA_UAU.7,** **FIA_UID.2,** **FIA_AFL.1,** **FIA_ATD.1,** **FIA_SOS.1,** **FCS_COP.1 ,** **FTA_SSL.1** **FTA_SSL.4,** **FMT_SMR.1,** **FMT_MSA.1 ,** **FMT_MSA.3 ,** **FMT_SMF.1,** **FDP_ACC.2,** **FDP_ACF.1**	Before performing any action, FIA_UAU.2 forces TOE users to authenticate as well as identify provided by FIA_UID.2. FIA_UAU.7 provides multiple authentication mechanism for users. FIA_AFL.1 protects the TOE against brute-force attacks by introducing a protection mechanism. FIA_ATD.1 provides maintaining of security attributes such as: user id, name, e-mail, password, user role. FIA_SOS.1 also contributes to this objective because by this component TSF defines the rules for secrets which contribute to the measures taken against unauthorized access. FCS_COP.1 provides hashing function for user password. FTA_SSL.1 provides session termination after a defined period of inactivity. FTA_SSL.4 allows users to terminate their own session. FMT_SMR.1 associates users with role groups that include static & dynamic authorizations. FMT_MSA.1 applies the specified policy to manage security attributes to authorized users. FMT_MSA.3 limits to be able to manage default values for attributes according to a specified policy. FMT_SMF.1 and FMT_SMR.1 determines the management functions and roles. FDP_ACC.2, FDP_ACF.1 specify access control policy

Objectives	SFRs	Rationale
		details, information and rules on the management functions.
O. AUDIT	**FAU_GEN.1, FAU_GEN.2**	Auditing requirements of TOE are defined by using FAU_GEN.1 and generated audit records are associated with users of TOE by FAU_GEN.2.

Objectives	SFRs	Rationale
O. PROTECTED_COMMUNICATION	**FTP_TRP.1**	FTP_TRP.1 helps to establish a secure channel from the user's browser to Havelsan GÖZCÜ application protecting the user data from disclosure and modification.

Table 5 Security Functional Requirements Rationale

6.3.3. SECURITY FUNCTIONAL REQUIREMENTS RATIONALE TABLES

	O.AUTH	O.PROTECTED_COMMUNICATION	O.AUDIT
FAU_GEN.1			X
FAU_GEN.2			X
FCS_COP.1	X		
FDP_ACC.2	x		
FDP_ACF.1	x		
FIA_AFL.1	X		
FIA_ATD.1	X		
FIA_SOS.1	X		
FIA_UAU.2	X		
FIA_UAU.7	X		
FIA_UID.2	X		
FMT_MSA.1	x		
FMT_MSA.3	x		
FMT_SMR.1	X		
FMT_SMF.1	x		
FTA_SSL.1	X		
FTA_SSL.4	X		
FTP_TRP.1		X	

Table 6 SFR Rationale Table for TOE

6.3.4. SECURITY ASSURANCE REQUIREMENTS RATIONALE

EAL4 is the highest level at which it is likely to be economically feasible to retrofit to an existing product line. EAL4 is applicable in those circumstances where developers or users require a moderate to high level of independently assured security in conventional commodity TOEs and are prepared to incur additional security-specific engineering costs. In addition, ALC_FLR.1 is chosen to pro-

vide additional quality assurance to the TOE.

7. TOE SUMMARY SPECIFICATION

7.1. TOE SECURITY FUNCTIONS

7.1.1. LOG GENERATION

The TSF generates audit logs for the following events:

- All user actions
- All admin actions
- System actions (info, debug, error, warning)

Date and time of events, type of events, usernames, success and failure of operations are recorded. When audit storage area is exceeded, audit function stops and the administrators are warned of the situation with an e-mail.

Implemented SFR's: FAU_GEN.1, FAU_GEN.2

7.1.2. CRYPTOGRAPHIC KEY MANAGEMENT & OPERATIONS

Bcrypt is used by the TSF for storing and authenticating users' passwords.

Zamane stamp provided by TÜBİTAK Certification Centre is used by the TSF for storing archived files to ensure that no further change can be done over them.

Hashing is provided by the TSF for storing and authenticating users' passwords. No key management is required for this operation since hashing does not require any additional keys. Since key creation is not required there is no key destruction involved in TOE.

Implemented SFR's: FCS_COP.1

7.1.3. USER LOGIN AND AUTHENTICATION

When a user issues a request to the TOE to access a protected resource, the TOE requires that the users identify and authenticate themselves before performing any action on behalf of the users. Users' passwords are controlled according to the following password quality requirements:

every password must have

- at least 8 characters,

- at least 1 number
- at least 1 uppercase letter,
- at least 1 lowercase letter,
- at least 1 special character.

Once the user attempts admin defined times of unsuccessful authentication, his/her status is disabled and her/his IP will be blocked for an admin defined interval.

The TSF shall maintain the following security attributes belonging to individual users:

username, password, email, user group, phone number.

Implemented SFR's: FIA_AFL.1, FIA_ATD.1, FIA_SOS.1, FIA_UAU.2, FIA_UAU.7, FIA_UID.2

7.1.4. PROTECTION OF DATA

The access control function permits a user to access a protected resource only if a user ID or role of the user is given permission to perform the requested action on the resource by Administrator. On the other hand, Authorized administrators of the TOE can perform assigning the privileges, modify his/her own authentication data, users' password and other information. Subjects:

- User
- Administrator
- Objects:
- GÖZCÜ Logs (Not related to FAU_GEN.1 or FAU_GEN.2)
- Parameters
- Configuration Meta Data
- User Roles
- Menu Items
- Report Data
- Statistics Data
- Data Parsing Metadata
- Data Rule Metadata

- Data Classification metadata
- Data processing request data

Implemented SFR's FDP_ACC.2, FDP_ACF.1, FMT_MSA.1, FMT_MSA.3

7.1.5. User Roles and Security Rules

Only administrators are allowed to manage and configure security functions. Authorized administrators are capable of performing the following management functions:

change active/passive state of a user

changing access control settings of users.

User roles are maintained by the TSF and users can be associated with these roles by authorized administrators. These roles are default user (which is the default user group with no authorizations) and admin by default. Also, admins are able to create custom roles.

The TOE supports the following roles:

a) User(Default)
b) Admin Created Role Groups (Authorized User)
c) Administrator

The TOE implements an access control SFP named "User Access Control SFP".
User Access Control SFP states that:
User (role group) is the default role group and has no privileges. Administrator on the other hand has all the privileges and has ability to create role groups. After creating role groups and setting their authorizations, administrator can bind users with role groups. Authorizations are not action based, they are page based and only users with related authorizations can see and use the pages.

Implemented SFR's: FMT_SMF.1, FMT_SMR.1

7.1.6. Connection via Trusted Path

SSL functions are not implemented within TOE and they are maintained by 3rd party libraries. Although SSL functions are outsourced, TOE also has secure connection control functions and it refuses any non-secure requests so that users' sessions are always established through trusted path.

Implemented SFR's: FTP_TRP.1

7.1.7. TOE ACCESS

When a session is inactive for 30 minutes, the user session is locked, making the display contents unreadable and users access is disabled. The user has to re-login to gain access and display functions.

The session inactivity of users exceeds 30 minutes, users are returned to the login page. The users are also able to terminate their own sessions.

Implemented SFR's: FTA_SSL.1, FTA_SSL.4

8. REFERENCES

[1]: Common Criteria Information Technology Security Evaluation Version 3.1 Rev 5 Part 1:

https://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R5.pdf

[2]: Common Criteria Information Technology Security Evaluation Version 3.1 Rev 5 Part 2:

https://www.commoncriteriaportal.org/files/ccfiles/CCPART2V3.1R5.pdf

[3]: Common Criteria Information Technology Security Evaluation Version 3.1 Rev 5 Part 3:

https://www.commoncriteriaportal.org/files/ccfiles/CCPART3V3.1R5.pdf