# NETCAD-Enterprise Products Platform

# v1.0

# Security Target

# Lite

Version No: 2.3

## VERSION HISTORY

| Version No | Reason for Change | Author | Release Date |
|---|---|---|---|
| 0.1 | First Draft | Özgür Yılmaz Barış Kaya | 27.01.2017 |
| 1.0 | First Publication | Özgür Yılmaz Barış Kaya | 07.04.2017 |
| 1.1 | First Revision | Özgür Yılmaz Barış Kaya | 23.06.2017 |
| 1.2 | Second Revision | Özgür Yılmaz Barış Kaya | 21.07.2017 |

| | | | |
|---|---|---|---|
| 1.3 | Third Revision | Özgür Yılmaz<br>Barış Kaya | 11.08.2017 |
| 1.4 | Fourth Revision , FIA_USB removed, Logical scope is rewritten, difference between Applications’ aliases are defined | Özgür Yılmaz<br>Barış Kaya | 24.08.2017 |
| 1.5 | Fifth Revision SFR’s are updated, TOE use cases are added | Özgür Yılmaz<br>Barış Kaya | 08.09.2017 |
| 1.6 | Sixth revision, format updated and summary is improved, delivery information is added | Özgür Yılmaz<br>Barış Kaya | 2.10.2017 |
| 1.7 | Seventh revision, minor fixes, new MSA iterations added | Özgür Yılmaz<br>Barış Kaya | 12.10.2017 |
| 1.8 | FDP_ACC.1, ACF.1, FMT_MSA.1,MSA.3 Updated | Özgür Yılmaz<br>Barış Kaya | 23.10.2017 |
| 1.9 | ST Reference, OE&Assumptions and FDP_ACC requirements are updated | Özgür Yılmaz<br>Barış Kaya | 3.11.2017 |
| 2.0 | FMT_MSA.1/3 ,FPT_FLS and FDP_ITC/ETC are changed. Guest user removed | Özgür Yılmaz<br>Barış Kaya | 8.12.2017 |
| 2.1 | Organisational Policies added | Özgür Yılmaz<br>Barış Kaya | 15.05.2018 |
| 2.2 | Table 1 updated | Özgür Yılmaz<br>Barış Kaya | 16.05.2018 |
| 2.3 | Use cases of TOE are added | Özgür Yılmaz<br>Barış Kaya | 27.07.2018 |

## GLOSSARY

**NetAdmin:** NetAdmin is the administrator user that comes out of the box with TOE. It has no restrictions on its actions. It can also create other admins by giving all possible authorizations to a role group.

**NetSystem:** NetSystem is a system user that comes default. It only has static authorizations and it is only able to perform automed or timed tasks.

**TSF Mediated Data:** TSF Mediated Data is the user created data that is used in the TOE. Geographicial information like city traffic density information can be given as an example to TSF Mediated Data.

**Meta data:** Meta data is the user created data that customizes the TOE. It is used for defining new menu items ,pages and actions.

**Administrator:** Administrator in this document refers both to NetAdmin and users with administrative authorizations.

**Service user:** Service users are any kind of users that communicate TOE via SSL to import , export, query ,create, update, delete TSF Mediated Data over a web service. Service users are also referred as remote users in this document

## CONTENTS

## List of Tables

# LIST OF FIGURES

# 1. ST Introduction

## 1.1. Security Target & TOE Reference

**ST Title:** NETCAD – Enterprise Products Platform Security Target Lite

**ST Reference:** v2.3

**TOE Identification:** NETCAD – Enterprise Products Platform v1.0

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

**Assurance Level:** EAL4+ (ALC_FLR.2)

**Keywords:** GIS, Geographical Information Systems, MIS, Management Information Systems, GEOCODING, GEOSPATIAL

**Note:** This Security Target Lite has been derived from the full Security Target Version 2.3.

## 1.2. TOE Overview

TOE consists of web applications which are installed onto the operating system in a computing platform.

These web applications are assembled in three groups:

- **Netigma**
  Netigma is the web application which end users mainly use. Netigma web application provides view and modification operations on TSF data. Also dynamic reports and queries are presented on Netigma. End users also reach map data and spatial data using Netigma. Reports, queries and objects which are related to TSF data can be managed by administrator users on development pages.

- **NetGIS Server**
  NetGIS Server is the web application consisting of web services which provides map data and map drawings as partial according to related coordinates. NetGIS Server also has a configuration interface.  .

- **NetCad Base**
  NetCad Base consists of three types of web applications. Those are:

- **Parameter Server:** Provides management of configuration parameters by administrator users for all applications
- **Log Server:** Provides log review and filtering by administrator users for all applications.
- **Authentication Server:** Manages access control for all applications using single sign-on mechanism, authorization of users and verification of user-rights for all applications. User, group, role and authorization management is also done by administrator users on authentication server.

Web Server(s) which TOE components are hosted on, Database Server(s) which TOE should use to store data, BIOS and other firmware, the operating system kernel, and other systems software (and drivers) provided as part of the platform are outside the scope of this document.

### 1.2.1. TOE Components' Use Cases

Components of TOE have different use cases which are same in core security functionalities and design with the versions included in TOE but different in naming, meta data and licensing.

Different use cases of web applications in TOE can be listed as below:

- **Netigma**

  NETIGMA Developer: Netigma version that has the licence that is needed to develop project/product.

  NETIGMA Runtime: Netigma version that has the licence that is needed for the server that Netigma will run on.

  BELNET.WEB : Municipal administration application

  KEOS.ABS : Address Information System Application

  KEOS.TBS : Immovable Information System Application

  KBS : Expropriation Information System

  KEOS.PARK BAHCELER : Parks and recreation works application

  KEOS.FEN ISLERI : Civil works application

KEOS.YAPI.RUHSATI : MAKS Integrated Building Forms application

KEOS.ISTEK SIKAYET: Request Complaint Tracking Application

KEOS.KPS Servisi : Identity Sharing System Integration Application

KEOS.MAKS Servisi : Population and Citizenship Affairs MAST System Integration Application

KRM.TAPUSERVIS : Land Registry System Online Service Integration Web Application

KEOS.SBS : Infrastructure Network Information System Application

KEOS.PYS : Plan Management System Application

KEOS.MEBIS : Cemetery Information System Application

KEOS.EIMAR : E-Reconstruction Application via Internet application

KEOS.KENT REHBERI : Interactive City Guide Application

KEOS.KENT REHBERI 3D : 3D Interactive City Guide Application

KEOS.VERIAKTARIM : Data transfer application

KEOS.MISGIS/TEMEL : Basic MIS GIS Integration Application

KEOS.MISGIS/ILERI : Advanced MIS GIS Integration Application

KEOS.ABIS : Disaster Information System Application

KEOS.AYKOME : Infrastructure and Coordination Center Application

KRM.TAPU : Land Registry System Online Service Integration Web Application

KRM.MOBILTAPU : Land Registry System Online Service Integration Mobile Application

KEOS.KENT REHBERI.NE– Nöbetçi Eczane : Pharmacy on duty locator application

KEOS.KENT REHBERI.NW – Kısayol : City Guide and shortcuts application

Coğrafi Plan Arşiv Uygulaması : Geographical plans archiving application

KEOS.EIMAR.ASKI : Municipality zoning tracking and publishing application

Tadilat Süreç Takip Sistemi : Renovation tracking system

- **NetGIS Server**

  NETGIS Server Standard : Standard NETGIS Server Application

  NETGIS Server Enterprise : OGC compliant Web Based Geo Data Server Application

  NETGIS Server 360 : Panoramic Imaging Server Application

  NETGIS Server Network : Standard NETGIS Server Application Configured for Network Applications

  NETGIS Server WebEdit : NETGIS Server Web Edit Server Application

- **NetCad Base**

### 1.2.2. Major Security Features of a TOE

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 system users. The assigned roles have the capability to review the audit logs.
- **Cryptographic Support:** TOE should provide mechanisms for encryption and decryption of session data and encryption of user passwords
- **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 software solution which provides GIS and security services to third party IT solutions and/or users.

### 1.2.3. Non TOE Hardware/ Software/ Firmware

TOE consists of 3 different parts and they are called Netigma, NetGIS Server and NetCad Base. Software and hardware requirements for each group are given below:

**NETCAD Base Hardware Requirements**

| Hardware | Requirement |
|---|---|
| Processor | 1 GHz + |
| RAM | 512 MB + |
| Disk space | 850 MB (32 bit), 2GB (64 bit) |

**NETCAD Base Software Requirements:**

- Internet Information Service (IIS) 6.0+
- Microsoft .NET Framework 4.6

**Netigma Hardware Requirements**

| Hardware | Requirement |
|---|---|
| Processor | 1 GHz + |
| RAM | 512 MB + |
| Disk space | 850 MB (32 bit), 2GB (64 bit) |

**Netigma Software Requirements:**

- Internet Information Service (IIS) 6.0+
- Microsoft .NET Framework 4.6
- Netcad Base 1.0.24+
- Relational Database:
- Access
- Oracle
- PostGIS
- SQL Server
- DB2

**NetGIS Server Software Requirements**

- Internet Information Service (IIS)
- Microsoft .NET Framework 4
- Netcad 6.0/ GP10 or higher
- Netcad Base 1.0.24
- Relational Database:
- Access
- Oracle
- PostGIS
- SQL Server

- DB2

**NETGIS Server Minimum Hardware Requirements**

| Hardware | Requirement |
|---|---|
| CPU | 1 GHz |
| RAM | 512 MB |
| Disk Space | 850 MB (32 bit) - 2GB (64 bit) |
| | |

#### 1.2.3.1.Typical Software/ Firmware Environment of TOE

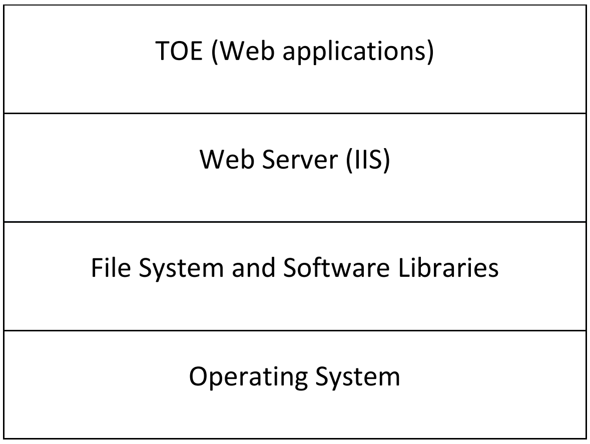

**Figure 1. Typical Software/Firmware Environment of TOE**

Web applications in TOE scope are web applications which run on web server. All web applications of TOE can be configured to run on a single server or server farm consists of multiple servers.

TOE operates using a database to store data. Database also can be hosted on the same server which TOE applications run or on a separated database server.

#### 1.2.3.2.Hardware Environment of TOE

Servers in the operating environment can be configured to operate on the same local network or different remote networks.

## 1.3. TOE Description

### 1.3.1. TOE Physical Scope

TOE scope consists of Netigma, NetGIS Server and Netcad Base. Since TOE is a software that works with 3rd party database applications database is not included in the TOE Scope. TOE is a group of web applications running on a webserver. Webserver and other hardware are not included in the scope.

**Delivery and Integration of TOE:**

TOE is delivered to customers by a sales representative. Deliverables are listed below:

-Netigma.exe : Netigma installation file

Netcad Base.exe : Netcad Base installation file

Webgis SDK.exe : Netgis SDK installation file

Netgis Server.exe : Netgis Server installation file

Install manual.pdf : Installation manual for TOE

Usage manual.pdf : User guide for TOE

Security Helper.exe : Application used for applying CC configurations

License key files

3rd party extentions needed by TOE are included in installation files. Other dependencies based on

Windows version of the system are provided by the OS.

**Configuration for the evalutated version of TOE:**

These configurations are all done using SecurityHelper.exe :

-Unsecure SSL/TSL channels are closed using OS settings

-Base application addresses are updated as https

-Machinekeys that are use for cryptographic operations are updated

-Exception handling settings are done to enable showing detailed errors to user

-HTTP Header configurations are done

-Anti clickjacking is activated

-Input validation is activated

-Parameter server is configured as below:

-Two-factor authentication with email is activated

-Captcha is enabled

-Cookies are set to require SSL by default

-Password complexity is set as 4

-Number of allowed unsuccessful login attempts before ban is set to 5

### 1.3.2. TOE Logical Scope

**Audit:**

TOE generates audit logs that consist of various auditable events. Those logs include information about actions like user login/log out events, meta data changes, rule changes and errors. User IP, application name, log description, database table name, old and new values for modified data attributes  and date and time of events are recorded. TOE allows authorized administrators to filter, search and review all the recorded logs stated above.

**Cryptographic Support:**

TOE uses AES-256 algorithm and cryptographic key sizes 256 bits that meet the following: FIPS 197 (for AES) for encryption and decryption of session data and encryption of user passwords.

Cryptographic keys are generated and destructed using mechanisms provided by IIS, they can be generated by using manual or automatic triggers that can be configured by Administrator.

**Identification, Authentication and Authorization:**

TOE provides an identification and authentication layer with a login page as a part of GUI. Since TOE consists of 3 different applications, each application require different logins altough they can be used with same credentials.

TOE includes administrator defined role groups and provides authentication before any action. This security feature acts to protect and prevent access by unauthorized users to
the system.

TOE also provides configurable authentication failure handling by locking users after an administrator defined number of unsuccesful login attemps .

Authorized administrators are granted the ability to set the idle timeout threshold after which an authorized user would be automatically logged out of his active session.

**Data Protection:**

TOE provides access control to TOE functions and Information flow control for TSF Data. The access control function permits a user to access a protected resource only if the role group of the user is given permission to perform the requested action on the resource by Administrator.

**Security Management:**

Although TOE includes static authorizations ,authorized administrator can also create dynamic authorizations.

Administrator can create role groups and bind static and dynamic authorizations to those role groups.After creating role groups administrator can assign role groups to users to define 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.

# 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.2 defined in CC part 3 (CC Part 3).

## 3. Security Problem Definition

This part of the ST defines the security problem that is to be addressed by the TOE. It consists of Assets, Subjects and External Entities, Organizational Security Policies, Threats and Assumptions.

### 3.1. Threats

**T. MASQUERADE:** An unauthorized user, process or external IT entity may masquerade as an authorized entity to gain access to data or TOE resources.

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

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

### 3.2. Organizational Security Policies

**P.CONF_KEY:** Keys that are used to encrypt and export confidential data are under administrator's responsibility. Administrator should keep that keys safe and never share with anybody.

**P.FULL_LOG_ACTION:** It is administrator's responsibility to take action when the informative mail about the reached log limit is sent by the TOE.

### 3.3. Assumptions

**A. PLATFORM:** The TOE relies upon a trustworthy computing platform for its execution. This includes isolation of the TOE Application from other applications on the platform.

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

**A. PROPER_ADMIN:** The administrator of the application software is not careless, willfully negligent or hostile, and administers the software within compliance of the applied enterprise security policy.

**A. PROPER_DB_ADMIN:** The administrator of the database(s) which application software uses, is not careless, willfully negligent or hostile, and administers the database within compliance of the applied enterprise security policy.

**A.SECURE_NETWORK:** The network connection between TOE parts are secure and uninterrupted.

# 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

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

**O. AUDIT**: TOE will provide the capability to create audit records of security relevant events associated with users and allow capability to review audit information.

**O. MANAGE**: TOE will allow administrators to effectively manage the TOE and its security functions and must ensure that only authorized administrators are able to access such functionality.

**O. DATA_PROTECTION**: TOE will protect the confidentiality and integrity of TSF data during transmission to other trusted IT entities.

**O. SECURE_COMM:** To address both passive (eavesdropping) and active (packet modification) network attack threats, conformant TOEs will use a trusted channel for sensitive data.

## 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 tracks with the assumptions about the 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.

**OE. PROPER_DB_ADMIN:** The administrator of the database(s) which application software uses, is not careless, willfully negligent or hostile, and administers the database within compliance of the applied enterprise security policy.

**OE.SECURE_NETWORK:** The TOE parts are installed in a secure network and the network connection between TOE parts are secure and uninterrupted.

## 4.3. Security Objectives Rationale

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

| Threat, Assumption, or OSP | Security Objectives | Rationale |
|---|---|---|
| **T. MASQUERADE** | **O. AUTHORIZATION**<br>**O. AUDIT**<br>**O. MANAGE** | The threat T. MASQUERADE is countered by O. AUTHORIZATION, O. AUDIT and O. MANAGE 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. NETWORK_ATTACK** | **O. AUDIT**<br>**O. DATA_PROTECTION** | The threat T. NETWORK_ATTACK will be countered by access control and information flow control based O. DATA_PROTECTION and also log management via O. AUDIT. |

| | | |
|---|---|---|
| **T. NETWORK_EAVESDROP** | **O. DATA_PROTECTION**<br>**O. SECURE_COMM** | The threat T. NETWORK_EAVESDROP will be countered by O. SECURE_COMM and O. DATA_PROTECTION where TOE will communicate via secure channels and information flow to third parties will be under security control. |
| **A. PLATFORM** | **OE. PLATFORM** | The assumption A. PLATFORM is addressed by OE. PLATFORM. This objective ensures that the TOE relies upon a trustworthy computing platform for its execution. |
| **A. PROPER_USER** | **OE. PROPER_USER** | The assumption A. PROPER_USER is addressed by OE. PROPER_USER. This objective ensures that the user of the application is educated about the software and uses the software in a secure manner. |
| **A. PROPER_ADMIN** | **OE. PROPER_ADMIN** | The assumption A. PROPER_ADMIN is addressed by OE. PROPER_ADMIN. This objective ensures that the administrator of the application software administers the software in a secure manner. |
| **A. PROPER_DB_ADMIN** | **OE. PROPER_ DB_ADMIN** | The assumption A.PROPER_DB_ADMIN is addressed by OE. PROPER_DB_ADMIN. This objective ensures that database administrator administers the database within compliance of the applied enterprise security policy. |

| | | |
|---|---|---|
| **A.SECURE_NETWORK** | **OE.SECURE_NETWORK** | The assumption A.SECURE_NETWORK is addressed by OE. SECURE_NETWORK. This objective ensures that the network connection between TOE parts are secure and uninterrupted. |
| **P.CONF_KEY** | **OE. PROPER_ADMIN** | The assumption P.CONF_KEY is addressed by OE. PROPER_ADMIN. This objective ensures that the administrator of the application software secures confidential keys in a secure manner. |
| **P.FULL_LOG_ACTION** | **OE. PROPER_ADMIN** | The assumption P.FULL_LOG_ACTION is addressed by OE. PROPER_ADMIN. This objective ensures that the administrator of the application software takes action when the log limit is reached. |

**Table 1. Security Objectives Rationale**

# 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 underlined text.

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.

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.

### 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 [minimum] level of audit; and

c) [

- *User login / logout*
- *Configuration Meta Data Save*
- *Application & Parameter Registration*
- *Application Data Modification*
- *Modification on Authorization Rules*

].

**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 PP/ST, [*user IP, application name, log type, error message (if applicable), stack trace (if applicable), log description, database table name (if applicable), old and new values for modified data attributes (if applicable), data input variables (for web forms - if applicable), priority (Debug, info, Error, Trace, Warn, Fatal)*].

| **SFRs** | Minimum Audit |
|---|---|
| FAU_GEN.1 | - |
| FAU_GEN.2 | - |
| FAU_SAR.1 | - |
| FAU_SAR.2 | - |

| | |
|---|---|
| FAU_SAR.3 | - |
| FAU_SEL.1 | Minimal: All modifications to the audit configuration that occur while the audit collection functions are operating. |
| FAU_STG.1 | - |
| FAU_STG.3 | - |
| FCS_CKM.1/AES | Minimal: Success and failure of the activity. |
| FCS_CKM.4/AES | Minimal: Success and failure of the activity. |
| FCS_COP.1 | (see application note below)* |
| FDP_ACC.1 | - |
| FDP_ACF.1 | Minimal: Successful requests to perform an operation on an object covered by the SFP. |
| FDP_IFC.1 | - |
| FDP_IFF.1 | Minimal: Decisions to permit requested information flows. |

| | |
|---|---|
| FDP_ITC.2 | Minimal: Successful import of user data, including any security attributes. |
| FDP_ETC.2 | Minimal: Successful export of information. |
| FDP_SDI.2 | Minimal: Successful attempts to check the integrity of user data, including an indication of the results of the check. |
| FIA_AFL.1 | a) Minimal: the reaching of the threshold for the unsuccessful authentication attempts and the actions (e.g. disabling of a terminal) taken and the subsequent, if appropriate, restoration to the normal state (e.g. re-enabling of a terminal). |
| FIA_ATD.1 | - |
| FIA_SOS.1 | Minimal: Rejection by the TSF of any tested secret; |
| FIA_UAU.2 | Minimal: Unsuccessful use of the authentication mechanism; |
| FIA_UAU.5 | Minimal: The final decision on authentication; |

| | |
|---|---|
| FIA_UID.2 | a) Minimal: Unsuccessful use of the user identification mechanism, including the user identity provided; |
| FMT_MOF.1 | - |
| FMT_MSA.1 | - |
| FMT_MSA.3 | - |
| FMT_MTD.1 | - |
| FMT_SMF.1 | Minimal: Use of the management functions. |
| FMT_SMR.1 | Minimal: modifications to the group of users that are part of a role; |
| FPT_FLS.1 | - |
| FPT_ITT.1 | - |
| FPT_TDC.1 | Minimal: Successful use of TSF data consistency mechanisms. |
| FRU_FLT.1 | Minimal: Any failure detected by the TSF. |

| | |
|---|---|
| FTA_MCS.1 | Minimal: Rejection of a new session based on the limitation of multiple concurrent sessions. |
| FTA_SSL.3 | Minimal: Termination of an interactive session by the session locking mechanism. |
| FTA_SSL.4 | Minimal: Termination of an interactive session by the user. |
| FTA_TAH.1 | - |
| FTA_TSE.1 | Minimal: Denial of a session establishment due to the session establishment mechanism. |
| FTP_TRP.1 | Minimal: Failures of the trusted path functions. |

Table2: Audit logs table

**Application note:**

While salted SHA-2 operations take place at one time password events, operations using AES algorithm are done in every request so logging them would result performance issues. As a result all SHA-2 operations are logged but operations using AES algorithm are not.

### 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.

### 6.1.1.3.FAU_SAR.1 Audit Review

**FAU_SAR.1.1** The TSF shall provide [*authorized administrators*] with the capability to read [*all information of audit record*] from the audit records.

**FAU_SAR.1.2** The TSF shall provide the audit records in a manner suitable for the user to interpret the information.

### 6.1.1.4.FAU_SAR.2 Restricted Audit Review

**FAU_SAR.2.1** The TSF shall prohibit all users read access to the audit records, except those users that have been granted explicit read-access.

### 6.1.1.5.FAU_SAR.3 Selectable audit review

FAU_SAR.3.1 The TSF shall provide the ability to apply [*filtering and ordering*] of audit data based on [*log type, application name, user ID, database table name, user IP, username ,URL ,Posted Data, Notes, Current Data or Error, log date, priority*].

### 6.1.1.6.FAU_SEL.1 Selective audit

**FAU_SEL.1.1** The TSF shall be able to select the set of events to be audited from the set of all auditable events based on the following attributes:

a) [user identity]

b) [*log type (Security, Datachange etc.), priority (Trace, Debug,Info, Warn, Error, Fatal)*]

#### 6.1.1.7.FAU_STG.1 Protected audit trail storage

**FAU_STG.1.1** The TSF shall protect the stored audit records in the audit trail from unauthorized deletion.

**FAU_STG.1.2** The TSF shall be able to [prevent] unauthorized modifications to the stored audit records in the audit trail.

#### 6.1.1.8.FAU_STG.3 Prevention of audit data loss

**FAU_STG.3.1** The TSF shall [*ignore audited events andnotify administrator via e-mail*] if the audit trail exceeds [ *administrator configurable predefined limit*].

### 6.1.2. CLASS FCS: CRYPTOGRAPHIC SUPPORT

#### 6.1.2.1.FCS_CKM.1 /AES Cryptographic key generation - AES

**FCS_CKM.1.1** The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [*AES-256*] and specified cryptographic key sizes [*256 bit*] that meet the following: [*FIPS 197*].

#### 6.1.2.2.FCS_CKM.4 /AES Cryptographic key destruction - AES

**FCS_CKM.4.1** The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [*overwrite*] that meets the following: [*FIPS 197*].

#### 6.1.2.3.FCS_COP.1 / AES Cryptographic operation - AES

**FCS_COP.1.1** The TSF shall perform [*encryption and decryption*] in accordance with a specified cryptographic algorithm [*AES-256*] and cryptographic key sizes [*256 bits*] that meet the following: [*FIPS 197 (for AES), NIST Recommendation for Block Cipher Modes of Operations (for CBC mode)*].

**Application Note:** FCS_COP.1 / AES is used to encrypt session variables while writing to user’s cookie, and decrypt while reading from user’s cookie.

### 6.1.2.4.FCS_COP.1 / Salted SHA-2 Cryptographic operation – Salted SHA-2

**FCS_COP.1.1** The TSF shall perform [*one-way encryption*] in accordance with a specified cryptographic algorithm [*Salted SHA-2*] and cryptographic key sizes [*256 bits*] that meet the following: [*FIPS 180-4*].

**Application Note:** FCS_COP.1 / Salted SHA-2 is used to salt and encrypt user password before it is stored in database, and before login to compare with the values stored in the database.

## 6.1.3. Class FDP: Data Protection

### 6.1.3.1.FDP_ACC.1 Subset Access Control /A

**FDP_ACC.1.1** The TSF shall enforce the [*Auth SFP*] on [

*Subjects:*

- *User*
- *Administrator (predefined administrator called "NetAdmin")*

*Objects:*

- *User role groups*
- *Dynamic authorizations that are manually created by the administrator*
- *Predefined authorizations that are hard coded privileges*

*Operations:*

- *Create*
- *Delete*
- *Update*
- *Assocaiate*
- *Deassociate*

*covered by the SFP*].

### 6.1.3.2.FDP_ACF.1 Security Attribute Based Access Control /A

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

*-Dynamic and Static authorizations assigned to role groups and NetAdmin*

*-The users assigned to a role group*

].

**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 create/delete/update user role groups over application*

*- Administrator can create/delete/update Dynamic Authorizations over application*

*- Administrator can associate/deassociate Dynamic Authorizations and Static Authorizations with User role groups over application.*

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

*-User can create/delete/update Dynamic Authorizations if their user role group's authorizations allow it over application.*

*-User can create/delete/update user role groups if their user role group's authorizations allow it over application.*

*-User can associate/deassociate Dynamic Authorizations and Static Authorizations with User role groups if their user role group's authorizations allow it over application.*

*-User can associate/deassociate User role groups with Users if their user role group's authorizations allow it over application.*

].

**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*].

### 6.1.3.3.FDP_ACC.1 Subset Access Control / B

**FDP_ACC.1.1** The TSF shall enforce the [*Data Access Control SFP*] on [

*Subjects:*

- *User*
- *NetSystem*
- *Administrator (predefined administrator called "NetAdmin")*

*Objects:*

- *TSF Mediated Data*
- *Meta Data*
- *User Data*

*Operations:*

- *Query*
- *Insert*
- *Update*
- *Delete*
- *Import*
- *Export*

*covered by the SFP*].

### 6.1.3.4.FDP_ACF.1 Security Attribute Based Access Control /B

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

[

*-Static and dynamic authorizations assigned to a user's role group, NetAdmin and Netsystem*

].

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

*-NetAdmin can apply query/insert/update/delete/import/export actions over TSF Mediated Data, Meta Data and User Data over application.*

*-NetSystem can query/insert/update/delete/import/export actions over TSF Mediated Data and Meta Data if there is no extra dynamic authorization needed to take action over that data over application.*

*-User can query/insert/update/delete/import/export actions over TSF Mediated Data and Meta Data if both their user role group's authorizations allow it over application and there is no limitation (location limitation, ip limitation etc.) defined in the data over application.*

].

**FDP_ACF.1.3** The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [*if there is no dynamic or static authorization related to an action over a certain type of data, it can be used by any user* ].

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

### 6.1.3.5.FDP_IFC.1 Subset information flow control

**FDP_IFC.1.1** The TSF shall enforce the [*Information Flow Control SFP*] on [

*Subjects:*

*•Service Users*

*Subject attributes:*

*-SessionId*

*Information:*

*-TSF Mediated Data*

*Operations:*

*-Allow or Deny Query/Create/Update/Delete/Import/Export actions over a web service*

].

### 6.1.3.6.FDP_IFF.1 Simple security attributes

**FDP_IFF.1.1** The TSF shall enforce the [*information flow control SFP*] based on the following types of subject and information security attributes: [*Session Id*].

**FDP_IFF.1.2** The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: [

*-TOE shall allow/deny Service Users to apply Query/Create/Update/Delete/Import/Export actions on TSF Mediated Data over web services based on their Session Ids .*

].

**FDP_IFF.1.3** The TSF shall enforce the [*communication via secure SSL channel between client application and the TSF*].

**FDP_IFF.1.4** The TSF shall explicitly authorize an information flow based on the following rules: [*none*].

**FDP_IFF.1.5** The TSF shall explicitly deny an information flow based on the following rules: [*none*].

**ApplicationNote:**

Service users are any kind of clients that communicate TOE via SSL using Session Ids(that is unique to each user) to import , export, query ,create, update, delete TSF Mediated Data over a web service.

### 6.1.3.7.FDP_ITC.2 Import of User Data with Security Attributes

**FDP_ITC.2.1** The TSF shall enforce the [*Auth Access Control SFP ,Data Acces Control SFP and Information Flow Control SFP*] when importing **user data** , controlled under the SFPs, from outside of the TOE.

**FDP_ITC.2.2** The TSF shall use the security attributes associated with the imported **user data**.

**FDP_ITC.2.3** The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the **user** data received.

**FDP_ITC.2.4** The TSF shall ensure that interpretation of the security attributes of the imported user**, TSF Mediated and meta data** is as intended by the source of the **user data**.

**FDP_ITC.2.5** The TSF shall enforce the following rules when importing **user** data controlled under the SFP from outside the TOE: [*none*].

### 6.1.3.8.FDP_ETC.2 Export of User Data with Security Attributes

**FDP_ETC.2.1** The TSF shall enforce the [*Auth Access Control SFP, Data Access Control SFP and Information Flow Control SFP*] when exporting user data controlled under the SFPs, outside of the TOE.

**FDP_ETC.2.2** The TSF shall export the user data with the user data's associated security attributes.

**FDP_ETC.2.3** The TSF shall ensure that the security attributes, when exported outside the TOE, are unambiguously associated with the exported **user** data.

**FDP_ETC.2.4** The TSF shall enforce the following rules when **user** is exported from the TOE: [*none*].

### 6.1.3.9.FDP_SDI.2 Stored data integrity monitoring and action

**FDP_SDI.2.1** The TSF shall monitor user data stored in containers controlled by the TSF for [*integrity errors*] on all objects, based on the following attributes: [*hash calculation*].

**FDP_SDI.2.2** Upon detection of a data integrity error, the TSF shall [*show records with integrity error to the administrator on administrator level accessible integrity error check page*].

## 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 [*user login*].

**FIA_AFL.1.2** When the defined number of unsuccessful authentication attempts has been [met], the TSF shall [*lock the user account until it is unlocked by authorized administrator*].

### 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: [*user id, name, e-mail, password, user role*].

### 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 [*at least 4 of the following predefined metrics:*

- *password length should be at least 6 characters*
- *password length should be at least 12 characters*
- *password should contain numeric characters*
- *password should contain both capital and lower case characters*
- *password should contain non-alphanumeric characters*

].

### 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.5 Multiple authentication mechanisms

**FIA_UAU.5.1** The TSF shall provide [*login user interface and login with session id*] to support user authentication.

**FIA_UAU.5.2** The TSF shall authenticate any user's claimed identity according to the [

*User Interface Login: Users login with user name and password.*

*Login with session id: Users login using their session ids over a terminal interface*

].

### 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_MOF.1 Management of Security Functions Behaviour

**FMT_MOF.1.1** The TSF shall restrict the ability to [disable, enable, modify the behavior of] the functions [*user password complexity policy, audit log information detail level, modification on user data*] to [*administrator*].

### 6.1.5.2.FMT_MSA.1 Management of Security Attributes / A

**FMT_MSA.1.1** The TSF shall enforce the [*Auth SFP*] to restrict the ability to [modify, delete , [*create, grant ,revoke*]] the security attributes [*user role,static & dynamic authorizations*] to [*administrator & users in authorized role groups*].

### 6.1.5.3.FMT_MSA.1 Management of Security Attributes / B

**FMT_MSA.1.1** The TSF shall enforce the [*Data Access Control SFP*] to restrict the ability to [modify, delete , [*none*]] the security attributes [*user id, name, password*] to [*administrator & users*].

### 6.1.5.4.FMT_MSA.1 Management of Security Attributes / C

**FMT_MSA.1.1** The TSF shall enforce the [*information flow control SFP*] to restrict the ability to [modify, delete ,[*create*]] the security attributes [*Session id*] to [*service users*].

### 6.1.5.5.FMT_MSA.3 Static Attribute Initialization / B

**FMT_MSA.3.1** The TSF shall enforce the [*Data 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 [*administrator*] to specify alternative initial values to override the default values when an object or information is created.

### 6.1.5.6.FMT_MSA.3 Static Attribute Initialization / A

**FMT_MSA.3.1** The TSF shall enforce the [*Auth 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 [*administrator*] to specify alternative initial values to override the default values when an object or information is created.

### 6.1.5.7.FMT_MSA.3 Static Attribute Initialization / C

**FMT_MSA.3.1** The TSF shall enforce the [*information flow 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 [*service users*] to specify alternative initial values to override the default values when an object or information is created.

### 6.1.5.8.FMT_MTD.1 Management of TSF Data

**FMT_MTD.1.1** The TSF shall restrict the ability to [change default, query, modify, delete, clear, [*none*]] the [*User , TSF Mediated and Meta data*] to [*administrator(NetAdmin) and authorized users*].

### 6.1.5.9.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*].

### 6.1.5.10.FMT_SMR.1 Security Roles

**FMT_SMR.1.1** The TSF shall maintain the roles [NetSystem,*user, and administrator*].

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

### 6.1.6. Class FPT: Protection of The TSF

#### 6.1.6.1.FPT_FLS.1 Failure with preservation of secure state / Log Fail

**FPT_FLS.1.1** The TSF shall preserve a secure state when the following types of failures occur: [

*-Database connection cannot be established while trying to perform a logging event*

*-An error occurred in database while trying to write*

].

#### 6.1.6.2.FPT_FLS.1 Failure with preservation of secure state / Meta Fail

**FPT_FLS.1.1** The TSF shall preserve a secure state when the following types of failures occur: [

- *inconsistency in meta file*

- *nonexisting component used in meta file*].

#### 6.1.6.3.FPT_ITT.1 Basic internal TSF data transfer protection

**FPT_ITT.1.1** The TSF shall protect TSF data from [disclosure, modification] when it is transmitted between separate parts of the TOE.

**Application Note:** Netigma, NetGIS Server and NetCad Base uses secure SSL communication while transferring of TSF data between these separate web applications.

#### 6.1.6.4.FPT_TDC.1 Inter-TSF basic TSF data consistency

**FPT_TDC.1.1** The TSF shall provide the capability to consistently interpret [*user data and other TSF data*] when shared between the TSF and another trusted IT product.

**FPT_TDC.1.2** The TSF shall use [*validation of TSF data by transferred class level, session management of authorized web service requester*] when interpreting the TSF data from another trusted IT product.

### 6.1.7. Class FRU: Resource Utilisation

#### 6.1.7.1.FRU_FLT.1 Degraded fault tolerance / Log fail

**FRU_FLT.1.1** The TSF shall ensure the operation of [*logs will be written in a file*] when the following failures occur: [-*Database connection cannot be established while trying to perform a logging event, -Any writing error occurred while writing into database*].

#### 6.1.7.2.FRU_FLT.1 Degraded fault tolerance / Meta fail

**FRU_FLT.1.1** The TSF shall ensure the operation of [-*asking for administrator's comfirmation to solve meta file problems and reload the meta data*] when the following failures occur: [

- *inconsistency in meta file*

- *non-existing component used in meta file*].

### 6.1.8. Class FTA: ToE Access

#### 6.1.8.1.FTA_MCS.1 Basic limitation on multiple concurrent sessions

**FTA_MCS.1.1** The TSF shall restrict the maximum number of concurrent sessions that belong to the same user.

**FTA_MCS.1.2** The TSF shall enforce, by default, a limit of [*administrator defined*] sessions per user.

#### 6.1.8.2.FTA_SSL.3 TSF-initiated termination

**FTA_SSL.3.1** The TSF shall terminate an interactive session after a [*administrator defined interval of* *user inactivity*].

### 6.1.8.3.FTA_SSL.4 User-initiated termination

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

### 6.1.8.4.FTA_TAH.1 TOE access history

**FTA_TAH.1.1** Upon successful session establishment, the TSF shall display the [date, time, method, location] of the last successful session establishment to the user.

**FTA_TAH.1.2** Upon successful session establishment, the TSF shall display the [date, time, method, location] of the last unsuccessful attempt to session establishment and the number of unsuccessful attempts since the last successful session establishment.

**FTA_TAH.1.3** The TSF shall not erase the access history information from the user interface without giving the user an opportunity to review the information.

### 6.1.8.5.FTA_TSE.1 TOE session establishment

FTA_TSE.1.1 The TSF shall be able to deny session establishment based on [*user id, location of user, IP range of user, time range*].

## 6.1.9. CLASS FTP: TRUSTED PATHS

### 6.1.9.1.FTP_TRP.1 Trusted Path

**FTP_TRP.1.1** The TSF shall provide a communication path between itself and [remote] 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, disclosure].

**FTP_TRP.1.2** The TSF shall permit [remote 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.2 – Flaw reporting procedures |

| | |
|---|---|
| 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 | Dependency Met? |
|---|---|---|
| FAU_GEN.1 | FPT_STM.1 | Time stamps will be provided by operational environment. |
| FAU_GEN.2 | FAU_GEN.1<br>FIA_UID.1 | YES (FAU_GEN.1 ,FIA_UID.2 (is hierarchical to FIA_UID.1)) |
| FAU_SAR.1 | FAU_GEN.1 | YES (FAU_GEN.1) |
| FAU_SAR.2 | FAU_SAR.1 | YES (FAU_SAR.1) |
| FAU_SAR.3 | FAU_SAR.1 | YES(FAU_SAR.1) |
| FAU_SEL.1 | FAU_GEN.1<br>FMT_MTD.1 | YES (FAU_GEN.1, FMT_MTD.1) |
| FAU_STG.1 | FAU_GEN.1 | YES (FAU_GEN.1) |
| FAU_STG.3 | FAU_STG.1 | YES (FAU_STG.1) |
| FCS_CKM.1/ AES | [FCS_CKM.2, or FCS_COP.1]<br>FCS_CKM.4 | YES (FCS_COP.1/AES ,FCS_CKM.4/AES) |
| FCS_CKM.4/ AES | [FDP_ITC.1, or FDP_ITC.2, or FCS_CKM.1] | YES(FCS_CKM.1/AES) |

| | | |
|---|---|---|
| FCS_COP.1 / AES | [FDP_ITC.1, or<br>FDP_ITC.2, or<br>FCS_CKM.1 ]<br>FCS_CKM.4 | YES(FCS_CKM.1/AES,<br>FCS_CKM.4/AES) |
| FCS_COP.1 / Salted SHA-2 | [FDP_ITC.1, or<br>FDP_ITC.2, or<br>FCS_CKM.1 ]<br>FCS_CKM.4 | Since SHA-2 is a hashing algorithm and a one way function it doesn't need any key generation or destruction. There for the dependencies are not applicable. |
| FDP_ACC.1 / A | FDP_ACF.1 | YES (FDP_ACF.1/A) |
| FDP_ACC.1 / B | FDP_ACF.1 | YES (FDP_ACF.1/B) |
| FDP_ACF.1 / A | FDP_ACC.1<br>FMT_MSA.3 | YES (FDP_ACC.1/A) |
| FDP_ACF.1 / B | FDP_ACC.1<br>FMT_MSA.3 | YES (FDP_ACC.1/B) |
| FDP_IFC.1 | FDP_IFF.1 | YES (FDP_IFF.1) |
| FDP_IFF.1 | FDP_IFC.1<br>FMT_MSA.3 | YES (FDP_IFC.1, FMT_MSA.3/C) |
| FDP_ITC.2 | [FDP_ACC.1, or<br>FDP_IFC.1 ]<br>[FTP_ITC.1 , or<br>FTP_TRP.1 ] | YES<br>(FDP_ACC.1/B,FDP_ACC.1/A,FDP_IFC.1, FTP_TRP.1, FPT_TDC.1) |

| | | |
|---|---|---|
| | FPT_TDC.1 | |
| FDP_ETC.2 | [FDP_ACC.1, or<br>FDP_IFC.1] | YES(FDP_ACC.1/A, FDP_ACC.1/ B, FDP_IFC.1) |
| FDP_SDI.2 | - | - |
| FIA_AFL.1 | FIA_UAU.1 | YES (FIA_UAU.2 (is hierarchical to FIA_UAU.1)) |
| FIA_ATD.1 | - | - |
| FIA_SOS.1 | - | - |
| FIA_UAU.2 | FIA_UID.1 | YES (FIA_UID.2 (is hierarchical to FIA_UID.1)) |
| FIA_UAU.5 | - | - |
| FIA_UID.2 | - | - |
| FMT_MOF.1 | FMT_SMR.1<br>FMT_SMF.1 | YES (FMT_SMR.1, FMT_SMF.1) |
| FMT_MSA.1 / B | [FDP_ACC.1 or FDP_IFC.1 ] | YES (FDP_ACC.1/ |

| | | |
|---|---|---|
| | FMT_SMR.1<br>FMT_SMF.1 | B,FMT_SMR.1,FMT_SMF.1) |
| FMT_MSA.1 / A | [FDP_ACC.1 or FDP_IFC.1 ]<br>FMT_SMR.1<br>FMT_SMF.1 | YES (FDP_ACC.1/A,FMT_SMR.1,FMT_SMF.1) |
| FMT_MSA.1 / C | [FDP_ACC.1 or FDP_IFC.1 ]<br>FMT_SMR.1<br>FMT_SMF.1 | YES (FDP_IFC.1,FMT_SMR.1,FMT_SMF.1) |
| FMT_MSA.3 / A | FMT_MSA.1<br>FMT_SMR.1 | YES (FMT_MSA.1 / A,FMT_SMR.1 ) |
| FMT_MSA.3 / B | FMT_MSA.1<br>FMT_SMR.1 | YES (FMT_MSA.1 / B,FMT_SMR.1 ) |
| FMT_MSA.3 / C | FMT_MSA.1<br>FMT_SMR.1 | YES (FMT_MSA.1 /C,FMT_SMR.1 ) |
| FMT_MTD.1 | FMT_SMR.1<br>FMT_SMF.1 | YES (FMT_SMR.1,FMT_SMF.1) |
| FMT_SMF.1 | - | - |
| FMT_SMR.1 | FIA_UID.1 | YES (FIA_UID.2 (is hierarchical to |

| | | |
|---|---|---|
| | | FIA_UID.1)) |
| FPT_FLS.1 / Log Fail | - | - |
| FPT_FLS.1 / Meta Fail | | |
| FPT_ITT.1 | - | - |
| FPT_TDC.1 | - | - |
| FRU_FLT.1 / Log Fail | FPT_FLS.1 | YES (FPT_FLS.1/Log Fail) |
| FRU_FLT.1 / Meta Fail | FPT_FLS.1 | YES (FPT_FLS.1/Meta FAİL) |
| FTA_MCS.1 | FIA_UID.1 | YES (FIA_UID.2 (is hierarchical to FIA_UID.1)) |
| FTA_SSL.3 | - | - |
| FTA_SSL.4 | - | - |
| FTA_TAH.1 | - | - |
| FTA_TSE.1 | - | - |
| FTP_TRP.1 | - | - |

**Table 4. Security Functional Requirements Dependency Rationale**

### 6.3.2. Security Functional Requirements Rationale

| **Objectives** | **SFRs** | **Rationale** |
|---|---|---|
| **O. AUDIT** | **FAU_GEN.1,**<br>**FAU_SAR.1,**<br>**FAU_SAR.2,**<br>**FAU_SAR.3,**<br>**FAU_GEN.2,**<br>**FAU_SEL.1,**<br>**FAU_STG.1,**<br>**FAU_STG.3** | 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. FAU_SAR.1 provides the users of the TOE with a human-readable interface to the audit records while<br>FAU_SAR.2 prohibits normal users to review audit logs, only administrator is able to see audit logs<br>FAU_SAR.3 introduces an ability to TOE, with which audit records can be shown to the user in a selectable format.<br>FAU_SEL.1 provides administrator configurable selective audit<br>FAU_STG.1 provides protection of audit trail storage.<br>FAU_STG.3 provides notifications for administrator when the audit storage is near to exceed configurable limits. |

| Objectives | SFRs | Rationale |
|---|---|---|
| O. MANAGE | **FMT_MOF.1,**<br>**FMT_MSA.1 / A,**<br>**FMT_MSA.1 /B,**<br>**FMT_MSA.1 /C,**<br>**FMT_MSA.3 /A,**<br>**FMT_MSA.3 /B,**<br>**FMT_MSA.3 / C,**<br>**FMT_MTD.1,**<br>**FMT_SMF.1,**<br>**FAU_SAR.1,**<br>**FAU_SAR.2,**<br>**FAU_SAR.3,**<br>**FAU_SEL.1,**<br>**FAU_STG.3,**<br>**FDP_ACC.1 / A,**<br>**FDP_ACC.1 /** B,<br>**FDP_ACF.1 /A,**<br>**FDP_ACF.1/B,**<br>**FMT_SMR.1** | FMT_MOF.1 restricts the ability to manage security features to the authorized identified roles. FMT_MSA.1 iterations applies the specified policy to manage security attributes to authorized users. FMT_MSA.3 iterations limits to be able to manage default values for attributes according to a specified policy. FMT_MTD.1 allows authorized users to manage TSF data within the specified rules. FMT_SMF.1 and FMT_SMR.1 determines the management functions and roles.<br>FAU_SAR.1 provides the users of the TOE with a human-readable interface to the audit records while<br>FAU_SAR.2 prohibits normal users to review audit logs, only administrator is able to see and manage audit logs<br>FAU_SAR.3 introduces an ability to TOE, with which audit records can be shown to the user in a selectable format.<br>FAU_SEL.1 provides managable selective audit<br>FAU_STG.3 provides protection of audit trail storage.<br>FDP_ACC.1, FDP_ACF.1 iterations specify access control policy details, information and rules on the management functions. |

| Objectives | SFRs | Rationale |
|---|---|---|
| O. DATA_PROTECTION | FDP_IFC.1,<br>FDP_IFF.1,<br>FDP_ACC.1 / A,<br>FDP_ACC.1 / B,<br>FDP_ACF.1 /A,<br>FDP_ACF.1/B,<br>FDP_ITC.2,<br>FDP_ETC.2,<br>FPT_TDC.1,<br>FDP_SDI.2,<br>FPT_FLS.1 / Log Fail,<br>FPT_FLS.1 /Meta Fail<br>FPT_ITT.1,<br>FRU_FLT.1 /Log Fail,<br>FRU_FLT.1/ Meta Fail,<br>FCS_COP.1/ Salted SHA-2 | FDP_ACC.1, FDP_ACF.1 iterations specify access control policy details, information and rules. On the other hand, FDP_IFF.1, FDP_IFC.1 are the components that details information flow control policy rules. FDP_ITC.2 provide a functionality for imported data with consistency provided by FPT_TDC.1 verification as FDP_ETC.2 is a functionality that applies some security measures for exported data.<br>FDP_SDI.2 provides detection of data integrity errors<br>FPT_FLS.1 iterations provide preserving secure state when software errors occur<br>FPT_ITT.1 protects data from disclosure and modification during it is transmitted between parts of TOE<br>FRU_FLT.1 iterations provide operation of all critical TOE functions when a software error occurs.<br>FCS_COP.1 / Salted SHA-2provides salting and encryption of user password before it is stored in database. |
| O. SEC_COMM | FTP_TRP.1 | FTP_TRP.1 helps to establish a secure channel from the user's browser to GIS application protecting the user data from disclosure and modification. |
| O. AUTHORIZATION | FIA_UAU.2,<br>FIA_UAU.5,<br>FIA_UID.2, | Before performing any action, FIA_UAU.2 forces TOE users to authenticate as well as identify provided by FIA_UID.2. FIA_UAU.5 provides multiple authentication mechanism for users. FIA_AFL.1 protects the TOE against |

| Objectives | SFRs | Rationale |
|---|---|---|
| | **FIA_AFL.1,**<br>**FIA_ATD.1,**<br>**FIA_SOS.1,**<br>**FCS_CKM.1/AES,**<br>**FCS_CKM.4/AES,**<br>**FCS_COP.1 / AES,**<br>**FTA_MCS.1,**<br>**FTA_SSL.3,**<br>**FTA_SSL.4,**<br>**FTA_TAH.1,**<br>**FMT_SMR.1,**<br>**FTA_TSE.1** | 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 due to the fact that by this component TSF defines the rules for secrets which contribute to the measures taken against unauthorized access. FCS_CKM.1/AES and FCS_CKM.4/AES provides cryptographic key management which is used to encrypt / decrypt user session keys. FCS_COP.1 / AES provides encryption & decryption of user session variables.<br>FTA_MCS.1 limits users of starting multiple concurrent sessions.<br>FTA_SSL.3 provides session termination after a defined period of inactivity.<br>FTA_SSL.4 allows users to terminate their own session.<br>FTA_TAH.1 provides users history of successful and unsuccessful login attempts.<br>FTA_TSE.1 provides the capability to deny user sessions according to selected criteria such as banned user id, user’s IP range, user’s location etc.<br>FMT_SMR.1 associates users with role groups that include static & dynamic authorizations. |

**Table5. Security Functional Requirements Rationale**

### 6.3.3. Security Functional Requirements Rationale Tables

| SFR s | O. AUTHO-RIZATION | O. AUDIT | O. MA-NAGE | O. DATA_PROTECTION | O. SEC_COMM |
|---|---|---|---|---|---|
| FAU_GEN.1 | | ✓ | | | |
| FAU_GEN.2 | | ✓ | | | |
| FAU_SAR.1 | | ✓ | ✓ | | |
| FAU_SAR.2 | | ✓ | ✓ | | |
| FAU_SAR.3 | | ✓ | ✓ | | |
| FAU_SEL.1 | | ✓ | ✓ | | |
| FAU_STG.1 | | ✓ | | | |
| FAU_STG.3 | | ✓ | ✓ | | |
| FCS_CKM.1/ AES | ✓ | | | | |
| FCS_CKM.4/ AES | ✓ | | | | |
| FCS_COP.1 / AES | ✓ | | | | |
| FCS_COP.1 / Salted SHA-2 | | | | ✓ | |
| FDP_ACC.1 / A | | | ✓ | ✓ | |
| FDP_ACC.1 / B | | | ✓ | ✓ | |
| FDP_ACF.1 / A | | | ✓ | ✓ | |

| | | | | | |
|---|---|---|---|---|---|
| **FDP_ACF.1 / B** | | | ✓ | ✓ | |
| **FDP_IFC.1** | | | | ✓ | |
| **FDP_IFF.1** | | | | ✓ | |
| **FDP_ITC.2** | | | | ✓ | |
| **FDP_ETC.2** | | | | ✓ | |
| **FDP_SDI.2** | | | | ✓ | |
| **FIA_AFL.1** | ✓ | | | | |
| **FIA_ATD.1** | ✓ | | | | |
| **FIA_SOS.1** | ✓ | | | | |
| **FIA_UAU.2** | ✓ | | | | |
| **FIA_UAU.5** | ✓ | | | | |
| **FIA_UID.2** | ✓ | | | | |
| **FMT_MOF.1** | | | ✓ | | |
| **FMT_MSA.1 / A** | | | ✓ | | |
| **FMT_MSA.1 / B** | | | ✓ | | |
| **FMT_MSA.1 / C** | | | ✓ | | |
| **FMT_MSA.3 / A** | | | ✓ | | |
| **FMT_MSA.3 / B** | | | ✓ | | |
| **FMT_MSA.3 / C** | | | ✓ | | |
| **FMT_MTD.1** | | | ✓ | | |
| **FMT_SMF.1** | | | ✓ | | |
| **FMT_SMR.1** | ✓ | | ✓ | | |
| **FPT_FLS.1 / Log Fail** | | ✓ | | ✓ | |
| **FPT_FLS.1 / Meta Fail** | | ✓ | | ✓ | |
| **FPT_ITT.1** | | | | ✓ | |
| **FPT_TDC.1** | | | | ✓ | |

| | | | | | |
|---|---|---|---|---|---|
| **FRU_FLT.1 / Log Fail** | | ✓ | | ✓ | |
| **FRU_FLT.1 / Meta Fail** | | ✓ | | ✓ | |
| **FTA_MCS.1** | ✓ | | | | |
| **FTA_SSL.3** | ✓ | | | | |
| **FTA_SSL.4** | ✓ | | | | |
| **FTA_TAH.1** | ✓ | | | | |
| **FTA_TSE.1** | ✓ | | | | |
| **FTP_TRP.1** | | | | | ✓ |

**Table6. 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.2 is chosen to provide additional quality assurance to the TOE.

# 7. TOE Summary Specification

## 7.1. TOE Security Functions

### 7.1.1. Log Generation and Review

TOE generates audit logs in order to provide accountability for the administrators and system users. Administrators have the capability to review the audit logs.

When a log is created after a user's action, that log is associated with that user.

Audit logs can be reviewed by administrators using "Log Server" application and can be filtered and ordered according to criteria with logical relations.

TOE also have capability to generate selective audit according to configuration made by administrators.

TOE protects audit storage considering OE_PROPER_ADMIN operational environment constraint. TOE prevents  unauthorized users to make modification of logs and it protects audit records from unauthorized deletion.

TOE ignores audit events and notify administrators when audit storage exceeds administrator configurable limit.

Functional Requirement Satisfied: FAU_GEN.1, FAU_GEN.2,  FAU_SAR.1, FAU_SAR.2, FAU_SAR.3, FAU_GEN.2, FAU_SEL.1, FAU_STG.1, FAU_STG.3

### 7.1.2. Cryptographic Key Management & Operations

TOE provides mechanisms for encryption and decryption of session data and salting and encryption of user passwords. Cryptographic key generation and destruction mechanisms are also provided for keys used in session variables encryption & decryption.

Functional Requirement Satisfied: FCS_CKM.1/AES, FCS_CKM.4/AES, FCS_COP.1 / AES, FCS_COP.1 / Salted SHA-2

### 7.1.3. USER LOGIN AND AUTHENTICATION

TOE successfully identifies, authenticates and authorizes its users and it provides user identification & authentication before any action. Authentication failures are handled by locking user account until it is unlocked by administrator or the defined time interval expires. There are multiple mechanisms provided to authenticate users. Authentication of users are done via login interface, authentication of web services are done using user name, password and application security key which are provided by requester as parameters for web services. TOE also has the ability to deny session establishment based on user id, location of user, time range and IP range of user. After successful authentication TOE presents users successful and unsuccessful access history containing date, time, method and location of login attempts.

TOE restricts multiple concurrent sessions of the same user according to an administrator defined limit.

TOE maintains user id, name, e-mail, password, user role attributes belong to individual users. There is also a mechanism provided to verify user secrets meet the properties explained in 6.1.4.3 .

TOE terminates inactive user session after an administrator defined period and it also allows users to terminate their own session.

Functional Requirement Satisfied: FIA_UAU.2, FIA_UAU.5, FIA_UID.2, FIA_AFL.1, FIA_ATD.1, FIA_SOS.1, FTA_MCS.1, FTA_SSL.3, FTA_SSL.4, FTA_TAH.1, FTA_TSE.1

### 7.1.4. Protection Of User, TSF Mediated and Meta Data

TOE provides confidentiality and integrity of user and TSF data during import/export of data to/from third parties and it enforces Auth SFP ,Data Access Control SFP and Information Flow Control SFP for import and export of user data, meta data and TSF mediated data. For service users TOE enforces Information Flow Control SFP on all subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP using SSL secure channel

TOE preserves a secure state when application failures and user failures occur and  it ensures the operation of logs will be written in a file when database connection cannot be established while trying to perform a logging event. Furthermore, TOE ensures the operation of asking for administrator’s comfirmation to solve meta file problems and reload the meta data while there is inconsistency in meta file or nonexisting component used in meta file.

Moreover, TOE controls integrity of data stored in containers. Administrators may run integrity check on existing data via administrator user interface.

Functional Requirement Satisfied: FDP_ACC.1 /B, FDP_ACC.1 / A, FDP_ACF.1 /B , FDP_ACF.1 / A, FDP_IFC.1, FDP_IFF.1, FDP_ITC.2, FDP_ETC.2, FDP_SDI.2, FPT_FLS.1 / Log fail , FPT_FLS.1 /  Meta fail , FPT_ITT.1, FPT_TDC.1, FRU_FLT.1 / Log fail , FRU_FLT.1 /  Meta fail , FTP_TRP.1

### 7.1.5. User Roles and Security Rules

This feature contains managing security functions and data for different situations. Security roles, rules and conditions are identified and management is supplied according to roles, rules and conditions and only authorized people access the TOE.

The TSF enforces auth and data access control SFP, information flow control SFP to restrict the ability to modify, delete or unauthorized existing user, change user role, create, modify, delete user roles, and modify authorization of roles and the security attributes user id, name, password, user role to administrator user.

Administrator users can specify alternative initial values to override the default values when an object or information is created.

Only administrator user change_default, query, modify, delete or clear critical TSF data such as application configuration parameters, user data.

The TSF allows to create, delete, modify, and read security attributes defined in FIA_ATD.1 only to administrator users.

Functional Requirement Satisfied: FMT_MOF.1 , FMT_MSA.1 /A ,FMT_MSA.1 / B, FMT_MSA.1 /C ,FMT_MSA.3 /A , FMT_MSA.3 / B, FMT_MSA.3 /C , FMT_MTD.1, FMT_SMF.1, FMT_SMR.1 , FDP_ACC.1 /A , FDP_ACC.1 /B , FDP_ACF.1 /A , FDP_ACF.1 / B