BREACH+

V.2.0

SECURITY TARGET V.1.4

Document History

Version	Description	Date	Author
0.1	First Draft	02/02/2024	Muhammad Haider Ali
0.2	ST-document update	02/12/2024	Osama Ellahi
0.3	ST-document update	03/07/2024	Osama Ellahi
0.4	ST-document update in accordance with security target evaluation in v 2.2	03/09/2024	Osama Ellahi
0.5	ST-document update in accordance with security target evaluation in v 3.3	03/11/2024	Osama Ellahi
0.6	ST-document update in accordance with security target evaluation in v 4.3	03/13/2024	Osama Ellahi
0.7	ST-document update	03/14/2024	Osama Ellahi
0.8	ST-document update	04/25/2024	Osama Ellahi
0.9	ST-document update	04/29/2024	Osama Ellahi
1.0	ST- Document update	05/06/2024	Osama Ellahi
1.1	ST- Document update on CB comments v 1.0	05/27/2024	Osama Ellahi
1.2	ST- Document update on CB comments	06/06/2024	Osama Ellahi
1.3	Terminology Consistency updated according to CB comments	07/01/2024	Muhammad Haider Ali
1.4	Updated version of guidance in ST document	07/16/2024	Osama Ellahi

Table of Contents

1 INTRODUCTION

1.1 REFERENCES

ST Title	Breach+ Security Target
ST Version	v.1.4
TOE Title	Breach+
TOE Version	v2.0
Assurance Level	EAL1
CC Identification	▪ Common Criteria Part 1 Version 3.1 Revision 5 ▪ Common Criteria Part 2 Version 3.1 Revision 5 ▪ Common Criteria Part 3 Version 3.1 Revision 5 ▪ Common Methodology for Information Technology Security Evaluation (CEM) version 3.1 Revision 5

Table 1: ST and TOE References

1.2 TOE OVERVIEW

1.2.1 TOE USAGE AND SECURITY FEATURE

Breach+ is an online tool that helps users do important tasks securely. It checks how well security controls work by saving public exploits and trying out new attack paths in a safe way. It goes through the whole process of a cyberattack, acting like a real attacker to see if security rules and protections hold up. In Breach+, there are three user categories: Administrator user, MSSP user, and Client user. Administrator user oversee user management, attack inventory, and plugin management. MSSP user is empowered to establish and oversee Client user accounts. Finally, Client user serve as end-users, utilizing the platform for security audits.

Breach+ is super helpful for organizations wanting to make sure their defenses are strong. In addition to its primary functions, Breach+ provides detailed insights into potential vulnerabilities and strengths in security setups. By simulating real-world cyber threats, it helps users understand where their systems might be weak and how to strengthen them. With Breach+, users can stay one step ahead in the ongoing battle against cyber threats, ensuring their systems are well-protected and resilient against potential attacks.

Major security features of the TOE are listed below.

- ✓ Security Audit
  - The TSF keeps track of everything happening in the system through its Security Audit feature. It creates logs of different events, helping users see what's been going on. These logs are like a diary of activities, making it easy to check for any suspicious or unusual behavior. With Breach+, users can stay on top of their system's security by keeping a close eye on its activity.
- ✓ Protection of security functionality
  - Keeping important security functions safe and available is a top priority for TSF. TSF make sure that no one modifies/misuses these functions without permission.
- ✓ User Data Protection
  - TSF takes safeguarding user data seriously. Through access privilege assignments, it ensures that only users can access sensitive information. This helps prevent unauthorized access and keeps user data safe and secure.
- ✓ Identification and Authentication
  - In Breach+, only users can access what they need once they prove who they are by entering their email address, secret, and multifactor authentication code if it is enabled. This ensures that only the right people can get into the system and use its features, keeping everything secure.
- ✓ Security Management
  - Modify/Allocate Access Privileges: Authorized Administrator user can adjust and assign access privileges as needed. This means they can control who has access to what parts of the system, ensuring that sensitive information remains protected.

- Edit Authentication Data: Users can also edit authentication data. This allows for the maintenance and updating of user credentials, enhancing security by ensuring that only authorized individuals can access the system.

✓ TOE Access

- The Trusted Security Function (TSF) or users themselves can end active sessions. This ensures that sessions are terminated promptly when necessary, reducing the risk of unauthorized access.
- Breach+ prevents the establishment of sessions for users with inactive(disable) status. By denying access to those who shouldn't be logging in, it enhances security by only allowing authorized individuals to access the TOE.

1.2.2 TOE TYPE

Breach+ works as a system that includes both a website for user control and a special software agent that helps in checking security automatically. From the web interface, Administrator user have comprehensive control over user management and exploit handling functionalities. They can manage users and deal with security tests from there, making sure Breach+ runs smoothly and safely.

The special part of Breach+ is its agent software, which works on Windows and Linux computers. This agent is key for automatically checking if the security measures work well. It does this by downloading, running, and then reporting on different security tests without needing a person to do each step. This makes the whole process faster and better.

By including this agent in what we consider part of Breach+, we show that it's not just a website but also this smart agent working together. The website lets Client user control things easily, and the agent does the security checks by itself. This mix helps in protecting against security threats in a strong and efficient way.

1.2.3 FIRMWARE/HARDWARE/SOFTWARE REQUIRED BY TOE

The TOE requires two environments to operate. One is a browser to access the user portal and the second is an agent which must be on a Client user premises in any Windows or Linux operating system. In addition to requiring services from the environment to achieve its main goal, the environment also maintains a secure posture so that the application cannot be compromised by factors out of the TSF Scope of Control.

The following requirements are for TOE Windows Agent in Client user premises.

	Disk	Memory	CPU	Network
Recommended	100GB	8GB	4	1 network interface
Minimum	50GB	4GB	2	1 network interface

Table 2: HW/SW/FW Required by TOE Windows Agent

Platform	Version
Windows	Windows 10 Windows 11 Windows Server 2016 Windows Server 2019 Windows Server 2022

Table 3: HW/SW/FW Required by TOE Windows Agent

In addition to this the following requirements are for Breach+ Linux Agent in Client user premises.

	Disk	Memory	CPU	Network
Recommended	50GB	4GB	2	1 network interface
Minimum	50GB	2GB	2	1 network interface

Table 4: HW/SW/FW Required by TOE Linux Agent

Platform	Version
Centos	Centos 7- 8
Red Hat Enterprise	RHEL 7 - 8
Ubuntu	Ubuntu 18.04 or above
Debian	Debian 9 or above

Table 5: HW/SW/FW Required by TOE Linux Agent

In addition to this the following requirements are for Breach+ MAC Agent on the Client user premises.

	Disk	Memory	CPU	Network
Recommended	120GB	8GB	4	1 network interface
Minimum	80GB	4GB	2	1 network interface

Table 6: HW/SW/FW Required by TOE MAC Agent

Platform	Version
MacOS	MacOS 11

	Big Sur MacOS 12 Monterey MacOS 13 Ventura

Table 7: HW/SW/FW Required by TOE MAC Agent

Type	Details
Environment	Kubernetes 1.28.3
Operating System	Ubuntu Server 22.04
Web Server	NGINX
Database	Mongo DB 5.0.19
Server system	4 virtual CPU or higher processor, 2.4 GHz and 14 GB or more RAM
Client user system	1.6 GHz or higher processor, 4 GB or more RAM

Table 8: TOE Operational Environment

1.3 TOE DESCRIPTION

1.3.1 PHYSICAL SCOPE OF TOE

TOE works as a system that includes both a web application for user control and a special software agent that helps in checking security automatically. The TOE is hosted as a web application on a server, and its components are illustrated in figure 1. Additionally, the TOE includes an agent that operates on Windows and Linux systems, automating the process of checking and ensuring the security measures are effective. This agent plays a crucial role in the overall security framework by performing automated tasks to test and report on the system's defenses without manual intervention. As the portal is accessible through TOE URL, Client user can see all the steps (guide walkthrough) on web portal for using the TOE and for downloading and installing the agent. Client user can download the agent directly from the website. There is no need to physically handover the TOE to Client user.

1.3.1.1 Software:

The TOE is the following software:

a. **Web portal version 2.8** can be accessed by users from the identified link: https://apt.cytomate.net
b. **Agent version 2.0.1** (BreachPlusAgent.exe for windows and BreachPlusAgent.deb for Linux) installers can be downloaded from web portal link from the navbar.

1.3.1.2 Guidance Documents:

The TOE includes the following guidance documents:

a. Technical guide document of TOE **Guidance Document v 0.9.pdf**.

1.3.1.3 NON-TOE Components:

Following are the non-TOE operational requirements:

a. Endpoint Server.
b. Database.
c. All Kubernetes jobs (as illustrated in diamond shape in figure 1).

Web portal and endpoint server are accessible publicly. Web portals are used by all types of authorized users to access through any browser. TOE agent can be downloaded from web portal and must be installed in Client user premises where the security controls are to be assessed. TOE agent communicates with Web portal and endpoint server through API key. While Web portal starts Kubernetes job against each kind of assessment as illustrated in diamond shape in figure 1.

Endpoint component consists of threat libraries categorized for threat intelligence validation, adversary emulation and advance exploits. WAF (Web Application Firewall) consists of payloads from basic to advance to test the security of firewall of Client user. Email Gateway component consists of public and customized payload in form of docx, pdf etc. These are malicious documents which are sent from TOE email server to Client user email server to test the security, furthermore the email gateway also has a collection of phishing links to test security. Network component consists of malicious network traffic, which is simulated in the Client user network to test network firewall.

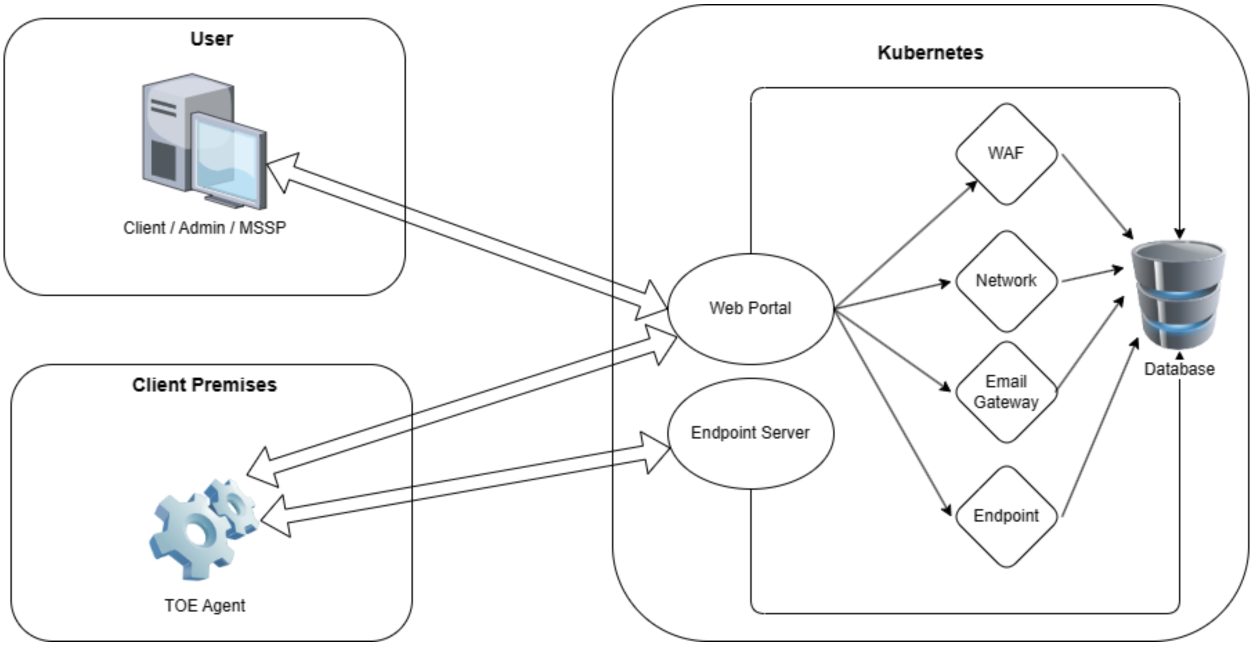

Figure 1: TOE architecture.

1.3.2 LOGICAL SCOPE OF TOE

The logical scope of the TOE is described through the security functionality as follows.

Security Audit: The TSF diligently generates audit logs, various auditable events, as mandated by FAU_GEN.1 Audit data generation. These logs include timestamps, user IDs, actions executed by users and object of events as mentioned in Table 11. Moreover, related to FAU_STG.1, the TSF ensures the protection of stored audit records in the audit trail from deletion. Additionally, as per FAU_STG.1.2, the TSF is designed to prevent unauthorized modifications to the stored audit records, thereby preserving the integrity and reliability of the audit trail.

Protection of Security Functionality:

The TOE ensures the security of its critical functionalities in the following manner:

- Basic Internal TSF Data Transfer Protection: The TOE employs mechanisms to protect the internal transfer of data between its security functions, ensuring that critical security information remains confidential and intact according to FPT_ITT.1.
- Reliable Time Stamps: To maintain the integrity and traceability of security-relevant events, the TOE generates reliable time stamps, enabling accurate auditing and forensic analysis of system activities in accordance with FPT_STM.1.

User Data Protection:

In accordance with the SFRs detailed in Section 6.1.2, the TOE prioritizes the protection of user data. By thoroughly implementing access privilege assignments, the TSF ensures that sensitive information remains accessible only to users. This implementation effectively prevents unauthorized access, strengthening the safety and security of user data.

Identification and Authentication:

The authentication process validates users via their account email and secret, ensuring only authorized individuals access sensitive TOE functions. This setup minimizes unauthorized entry and misuse. Additionally, security is implemented through multifactor authentication to align with FIA_UAU.5.2, including OTPs sent via email. In handling Identification and Authentication, the system needs to maintain a record of security information for every user, including their roles, secrets, and Multi-Factor Authentication (MFA) status, as outlined in FIA_ATD.1. This process enhances the precision of user identification and authentication. Furthermore, the TSF ensures that every action is checked to verify if the request is authenticated or if it is an illegitimate request according to FIA_UAU.2.

Upon each request, authentication and authorization processes are carried out to grant access to users only, supporting FIA_AFL.1 Authentication failure handling. This mechanism ensures that only legitimate users can interact with the TOE's sensitive functions, reducing the potential for unauthorized access and misuse of its capabilities.

There is a mechanism implemented where the TSF ensures that every secret, whether created by a new user or updated by an existing user, complies with the parameters defined in FIA_SOS.1.

Security Management:

Within FMT_MSA.1, the TOE's security management revolves around modifying and allocating access privileges to different plugins. This functionality enables Administrator user, granting them the authority to finely tune user permissions. Administrator user can thereby restrict or grant access to various TOE resources based on the roles and responsibilities of individual users. This comprehensive approach ensures alignment with established security protocols, enabling access control mechanisms and reinforcing the TOE's overall security architecture. An MSSP user has the capability to seamlessly switch into their Client user accounts at any time and execute tasks on behalf of the Client user. Conversely, an

Administrator user does not possess the ability to switch accounts; they must log in separately to perform any administrative tasks.

Toe Access:

The TOE access feature encompasses the management of user sessions, allowing for session termination by both the TOE's Security Functions (TSF) and Client user. This capability enhances the control Administrator user have over user interactions with the TOE. By terminating sessions, the TOE reduces the window of opportunity for unauthorized access and potential security breaches, contributing to the overall robustness of its security architecture. Each user session is terminated after 10 hours maximum (if active) and 15 minutes (if in-active).

2 CONFORMANCE CLAIM

2.1 CC CONFORMANCE CLAIM

This ST claims conformance to

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

The

- Common Methodology for Information Technology Security Evaluation, Evaluation Methodology; CCMB-2017-04-004, Version 3.1, Revision 5, April 2017, [4]

must be taken into account.

2.2 PP CLAIM

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

2.3 PACKAGE CLAIM

Evaluation Assurance Level is EAL1

3 SECURITY OBJECTIVES

3.1 SECURITY OBJECTIVES FOR TOE

The security objectives for the TOE are described in below:

O.AUTH The TOE must provide measures to uniquely identify and authenticate users before granting access to its protected functions or resources. Once users are confirmed and logged in, they should be able to utilize functions or resources aligned with their assigned roles. If a user exceeds five requests, their account should be temporarily inactive(disable) until an Administrator user re-enables it. The security system should stay to specific access control rules to regulate access. These rules dictate who can access various resources, such as user data and system modules, and what actions they can perform, like viewing or deleting. Access permissions are assigned based on user roles and permissions. The system should ensure that secrets meet specified requirements and that users can terminate their own sessions when desired. The TOE should implement TSF-initiated termination, this ensures that interactive sessions are automatically terminated by the TSF after a designated time, such as 15 minutes of inactivity or 10 hours of continuous activity, enhancing security and preventing unauthorized access to sensitive resources. Overall, these measures enhance the security and effectiveness of OAuth implementation.

O.AUDIT The TOE must audit data access, access to system functionalities, and all security-related operations, saving these logs. These logs should be monitored constantly, and it is allowed to review them when needed. This ensures that the audit trail securely stores all generated audit data and prevents deletion and unauthorized modification of these records, maintaining the integrity and reliability of the audit trail FAU_STG.1.

O.MGMT TOE must provide all necessary means and functions in order that an Administrator user manages the system securely and effectively. TOE shall restrict using these means and functions against unauthorized use and take necessary precautions.

O.LEGIT_USE The TOE should assess the domains on which the users have verified their ownership. The TOE should generate a API key from web portal and Client user must add this API key in to the domain’s TXT record. The ownership over domain must be authenticated through web portal by mapping the portal generated TXT record with Client user ID and then query domain for DNS TXT record internally through web portal.

O.MFA There must be multifactor authentication in place to provide an additional layer of security on top of the basic authentication mechanisms that are defined in *FIA_UAU.5.*

O.SAFE_EXEC The TOE must ensure that only production safe malware is executed in the premises of user environment. This objective must be ensured by secure transferring of tags within the TOE from web portal to agent. The tags decide execution of malware.

3.2 SECURITY OBJECTIVES FOR OPERATIONAL 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 and block denial-of-service attacks.

OE.SEC_ENV Operational environment of the web portal ensures physical and environmental security of the TOE. Unauthorized access is restricted and all components in the operational environment are secured. Only specifically authorized people are allowed to access critical components. All credentials stored by the endpoint agent are hashed, ensuring that email addresses and secrets cannot be stolen from systems where the agent is installed.

OE.CRED Those responsible for the TOE must ensure that all access credentials, such as secrets 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 user is non-hostile, well-trained, and follows all user guidance, installation guidance and configuration guidance.

4 EXTENDED COMPONENT DEFINITION

There is no extended components

5 SECURITY REQUIREMENTS

SFR Formatting

This section defines the security requirements satisfied by the TOE. Each requirement has been extracted from version 3.1 of the Common Criteria, Part 2 providing functional requirements and Part 3 providing assurance requirements.

Part 2 of the Common Criteria defines an approved set of operations that may be applied to security functional requirements. Following are the approved operations and the document conventions that are used within this ST to depict their application.

- **Assignment:** The assignment operation provides the ability to specify an identified parameter within a requirement. Assignments are depicted using **bolded text** and are surrounded by square brackets as follows [**assignment**].
- **Selection:** The selection operation allows the specification of one or more items from a list. Selections are depicted using *italics text* and are surrounded by square brackets as follows [*selection*].
- **Refinement:** The refinement operation allows the addition of extra detail to a requirement. Refinements are indicated using **bolded text**, for additions, and ~~strike-through~~, for deletions.

- **Iteration:** The iteration operation allows a component to be used more than once with varying operations. Iterations are depicted by an identifier at the end of the component identifier as follows FDP_ACC.1/IDENTIFIER

5.1 SECURITY FUNCTIONAL REQUIREMENTS (SFR)

This section specifies the security functional requirements for the TOE. It organizes the SFRs by the CC classes.

Requirement Class	Requirement Component
FAU: Security Audit	FAU_GEN.1: Audit Data Generation
	FAU_GEN.2: User identity association
	FAU_SAR.1: Audit Review
	FAU_STG.1: Protected Audit Trail Storage
FDP: User Data Protection	FDP_ACC.1: Subset Access Control
	FDP_ACF.1: Security Attribute Based Access Control
FIA: Identification and Authentication	FIA_ATD.1: User attribute definition
	FIA_AFL.1: Authentication failure handling
	FIA_UID.2: User identification before any action
	FIA_UAU.2: User authentication before any action
	FIA_UAU.5: Multiple authentication mechanisms
	FIA_SOS.1: Verification of secrets
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.3: TSF-initiated termination
	FTA_SSL.4: User-initiated termination
	FTA_TSE.1: TOE session establishment

FPT: Protection of Security Functionality	FPT_ITT.1: Basic Internal TSF data transfer protection
	FPT_STM.1: Reliable time stamps

Table 10: Security Functional Requirements

5.1.1 Security Audit

FAU_GEN.1 Audit data generation

Hierarchical to: No other components.

Dependencies: FPT_STM.1 Reliable time stamps

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 [*basic]* 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 ~~PP~~/ST, [**information specified in column two of Table 7, below**].

Component	Auditable Event
FAU_SAR.1	Reading of information from the audit records.
FDP_ACF.1	All requests to perform an operation on an object covered by the SFP.
FIA_AFL.1	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_UID.2	All use of the user identification mechanism, including the user identity provided

FIA_UAU.2	All use of the authentication mechanism
FIA_UAU.5	The result of each activated mechanism together with the final decision.
FIA_SOS.1	Rejection or acceptance by the TSF of any tested secret.
FMT_MSA.1	All modifications of the values of security attribute
FMT_MSA.3	All modifications of the initial values of security attribute.
FMT_SMF.1	Use of the management functions
FMT_SMR.1	modifications to the group of users that are part of a role
FTA_SSL.3	Termination of an interactive session by the session locking mechanism.
FTA_SSL.4	Termination of an interactive session by the user
FTA_TSE.1	All attempts at establishment of a user session.
FPT_STM.1	Changes to the time.

Table 11: Auditable Events

FAU_GEN.2 User identity association

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation

FIA_UID.1 Timing of identification

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.

FAU_SAR.1 Audit review

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation

FAU_SAR.1.1 The TSF shall provide [**Administrator user**] with the capability to read [**all audit information**] 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.

FAU_STG.1 Protected audit trail storage

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation

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.

5.1.2 User Data Protection

FDP_ACC.1 Subset access control

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute-based access control

FDP_ACC.1.1 The TSF shall enforce the [**Access Control Policy**] on [**subjects: Administrator user, MSSP user, Client user; object: user data, TOE Modules, secrets, User Attributes, Attack Inventory, installable Agents; operations: read, delete, execute**]

FDP_ACF.1 Security attribute-based access control

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control

FMT_MSA.3 Static attribute initialization

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

Security Attributes:

- **MSSP user: user identity, access control rules**
- **Administrator user: user role**

- **Client user: Client user identity, access control rules for the Client user, ownership of Client user and/or group membership**

Object Attributes: no additional security attributes].

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [**user is explicitly granted access to a function or resource if he/she belongs to a user group which has been granted access with following rules:**

- **MSSP user**
  - **Have access right for all TOE modules.**
  - **Read/modify user attributes.**
  - **Modify secrets of each user.**
  - **Create/update/delete user data for Client user.**
- **Administrator user can**
  - **Have access right for all TOE modules.**
  - **Read/modify user attributes.**
  - **Modify secrets of each user/MSSP user.**
  - **Create/update/delete user data for Client user/MSSP user.**
  - **Can create/update/delete attack inventory.**
  - **Can create/update/delete installable agents.**
- **Client user can**
  - **Read his/her own user attribute, assigned role.**
  - **Create/Modify his/her own secret.**
  - **Create/Modify/Delete access levels/roles for Client user groups**].

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**]

5.1.3 Identification and Authentication

FIA_AFL.1 Authentication failure handling

Hierarchical to: No other components.

Dependencies: FIA_UAU.1 Timing of authentication

FIA_AFL.1.1 The TSF shall detect when [**[*5*]]** unsuccessful authentication attempts occur related to [**user attempting to authenticate basis on user attributes**].

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been [*met*], the TSF shall [**inactive(disable) the user status until Administrator user enables it**]

FIA_ATD.1 User attribute definition

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users: [**associated roles of user, secrets, and Multi-Factor Authentication (MFA) status**]

FIA_SOS.1 Verification of secrets

Hierarchical to: No other components.

Dependencies: No dependencies.

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

a) Should contain at least one uppercase letter,

b) Should contain at least one lowercase letter,

c) Should contain at least one number,

d) Should contain at least one special character,

e) Should be at least 8 characters long]

FIA_UAU.2 User authentication before any action

Hierarchical to: FIA_UAU.1 Timing of authentication

Dependencies: FIA_UID.1 Timing of identification

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.

FIA_UAU.5 Multiple authentication mechanisms

Hierarchical to: No other components

Dependencies: No other components

FIA_UAU.5.1 The TSF shall provide [**email address and secret authentication, OTP verification using mail, Sign-in using Microsoft, domain verification using TXT record**] to support user authentication.

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

- **The TOE first verifies the email and secret and then verifies the six-digit OTP which was sent to the user's email address. If each verification has been successfully performed, further TSF-mediated actions are allowed.**
- **The TOE verifies that email type is organizational account and verifies MFA through MICROSOFT APIs and upon verification it also checks if email address is in TOE database.**
- **The TOE verifies if the key is saved by requesting user's provided domain TXT record and TOE also makes sure that one domain is not being used in any other user's account.]**

FIA_UID.2 User identification before any action

Hierarchical to: FIA_UID.1 Timing of identification

Dependencies: No dependencies.

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.

5.1.4 Security Management

FMT_MSA.1 Management of security attributes

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or

FDP_IFC.1 Subset information flow control]

FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions

FMT_MSA.1.1 The TSF shall enforce the [**Access Control Policy**] to restrict the ability to [[**switch**]] the security attributes [**account information**] to [**MSSP user**].

FMT_MSA.3 Static attribute initialization

Hierarchical to: No other components.

Dependencies: FMT_MSA.1 Management of security attributes

FMT_SMR.1 Security roles

FMT_MSA.3.1 The TSF shall enforce the [**Access Control Policy**] to provide [*restrictive*] default values for ~~security attributes~~ **access policy attributes defined in FDP_ACC.1** that are used to enforce the SFP.

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

FMT_SMF.1 Specification of Management Functions

Hierarchical to: No other components.

Dependencies: No dependencies.

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions: [**create, delete, modify, and read secret and MFA status; and switch to Client user account**].

FMT_SMR.1 Security roles

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification

FMT_SMR.1.1 The TSF shall maintain the roles [**Administrator user, Client user, MSSP user**].

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

5.1.5 TOE Access

FTA_SSL.3 TSF-initiated termination

Hierarchical to: No other components.

Dependencies: No dependencies.

FTA_SSL.3.1 The TSF shall terminate an interactive session after a [**time period of 15 minutes**]. **Every active session will be terminated by TSF automatically after 10 hours and if user is inactive, TSF will terminate the session after time of 15 minutes.**

FTA_SSL.4 User-initiated termination

Hierarchical to: No other components.

Dependencies: No dependencies.

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

FTA_TSE.1 TOE session establishment

Hierarchical to: No other components.

Dependencies: No dependencies.

FTA_TSE.1.1 The TSF shall be able to deny session establishment based on [**account status of the user**].

5.1.6 Protection of Security Functionality

FPT_ITT.1 Basic internal TSF data transfer protection

Hierarchical to: No other components.

Dependencies: No dependencies.

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

FPT_STM.1 Reliable time stamps

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps.

5.2 SECURITY ASSURANCE REQUIREMENTS (SAR)

The TOE meets the security assurance requirements for EAL1. The following table is the summary for the requirements.

Assurance Class	Assurance Components	Dependency	Dependency Met?
ADV: Development	ADV_FSP.1 Basic functional specification	-	-
AGD: Guidance documents	AGD_OPE.1 Operational user guidance	ADV_FSP.1 Informal functional specification	Yes
	AGD_PRE.1 Preparative procedures	-	-

ALC: Life-cycle support	ALC_CMC.1 Labeling of the TOE	ALC_CMS.1 TOE CM coverage	Yes
	ALC_CMS.1 TOE CM coverage	-	-
ASE: Security Target evaluation	ASE_CCL.1 Conformance claims	ASE_INT.1 ST introduction ASE_ECD.1 Extended components definition ASE_REQ.1 Stated Security Requirements	Yes
	ASE_ECD.1 Extended components definition	-	-
	ASE_INT.1 ST introduction	-	-
	ASE_OBJ.1 Security objectives for the operational environment	-	-
	ASE_REQ.1 Stated Security Requirements	ASE_ECD.1 Extended components definition	Yes
	ASE_TSS.1 TOE summary specification	ASE_INT.1 ST introduction ASE_REQ.1 Stated Security Requirements ADV_FSP.1 Basic functional specification	Yes
ATE: Tests	ATE_IND.1 Independent testing - conformance	ADV_FSP.1 Basic functional specification AGD_OPE.1 Operational user guidance. AGD_PRE.1 Preparative procedures	Yes
AVA: Vulnerability Assessment	AVA_VAN.1 Vulnerability survey	ADV_FSP.1 Basic functional specification AGD_OPE.1 Operational user guidance.	Yes

		AGD_PRE.1 Preparative procedures	

Table 12: Security Assurance Requirements

5.3 SECURITY REQUIREMENTS RATIONALE

5.3.1 SFR RATIONALE

SFR Dependency Rationale

The table below lists each SFR to which the TOE claims conformance with a dependency and indicates whether the dependent requirement was included.

SFR	Dependency	Dependency Met?
FAU_GEN.1	FPT_STM.1	YES
FAU_GEN.2	FAU_GEN.1 FIA_UID.1	YES YES (FIA_UID.2 is hierarchical to FIA_UID.1)
FAU_SAR.1	FAU_GEN.1	YES
FAU_STG.1	FAU_GEN.1	YES
FDP_ACC.1	FDP_ACF.1	YES
FDP_ACF.1	FDP_ACC.1 FMT_MSA.3	YES YES
FIA_ATD.1	-	-
FIA_UID.2	-	-
FIA_UAU.5	-	-
FIA_UAU.2	FIA_UID.1	YES (FIA_UID.2 is hierarchical to FIA_UID.1)
FIA_AFL.1	FIA_UAU.1	YES (FIA_UAU.2 is hierarchical to FIA_UAU.1)
FIA_SOS.1	-	-
FMT_MSA.1	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1, YES

	FMT_SMR.1 FMT_SMF.1	YES
FMT_MSA.3	FMT_MSA.1 FMT_SMR.1	YES, YES
FMT_SMF.1	-	-
FMT_SMR.1	FIA_UID.1	YES (FIA_UID.2 is hierarchical to FIA_UID.1)
FTA_SSL.3	-	-
FTA_SSL.4	-	-
FTA_TSE.1	-	-
FPT_ITT.1	-	-
FPT_STM.1	-	-

Table 13: SFR Dependency Table

5.3.2 SAR RATIONALE

The chosen assurance level is appropriate with the threats defined for the environment. The threats that were chosen are consistent with attacker of low attack motivation, therefore EAL1 was chosen for this ST. Table 12 indicates that all SAR dependencies have been fulfilled.

6 TOE SUMMARY SPECIFICATION

6.1.1 Security Audit

The TOE generates audit logs that consist of various auditable events at the basic level or actions taken by the MSSP user, Client user and Administrator user. These logs that are associated to users, are produced with a reliable time stamp provided by TSF. The TOE provides the capability for Administrator user to read and view all the recorded logs. The TSF prevents anyone from modifying or deleting audit logs.

TOE Security Functional Requirement Satisfied: *FAU_GEN.1, FAU_GEN.2, FAU_SAR.1, FAU_STG.1*

6.1.2 User Data Protection

Users can access with defined scopes over defined modules. Users can perform different actions allowed within scope of their permissions. Client user, MSSP user and Administrator user can modify their secrets. Administrator user can assign and revoke permissions to the users in their tenant. The Administrator user has the authority to enable and inactive(disable) the account of any Client user and MSSP user.

TOE Security Functional Requirement Satisfied: *FDP_ACC.1, FDP_ACF.1*

6.1.3 Identification and Authentication

The Identification and Authentication security function provides the TOE with the ability to govern access by user. The TOE ensures that a user (or Administrator user) identity is established and verified before access to the TOE is allowed. Prior to allowing access, the TOE requires Administrator user and users to be identified using an email address, secret, and a multifactor authentication mechanism according to *FIA_UAU.2*. Security is implemented through multifactor authentication, including OTPs sent via email, aligning with the requirement for *FIA_UAU.5*. Before successful completion of the security function, a user is unable to perform any of the relevant functions according to *FIA_AFL.1*. To manage Identification and Authentication, it requires the system to keep a list of security details for each user, such as their roles, secrets, and Multi-Factor Authentication (MFA) status according to *FIA_ATD.1*. This helps ensure accurate identification and authentication of users. There's a system in place where the TSF checks every secret to make sure it follows the rules set in FIA_SOS.1, whether it's created by a new user or updated by an existing one. Upon five consecutive failed authentication attempts, users' access privileges are automatically restricted until reauthorization is granted by Administrator user.

TOE Security Functional Requirements Satisfied: *FIA_AFL.1, FIA_ATD.1, FIA_UAU.2, FIA_UID.2, FIA_SOS.1, FIA_UAU.5*

6.1.4 Security Management

The TOE provides mechanisms to govern which users can access with resources or functions. The Security Management function allows the Administrator user to properly configure this functionality. Administrator user can assign access privileges to users by user level based on the functions or resources that they are allowed to perform or access. Administrator user can assign and revoke permissions to the users in their tenant. In FMT_MSA.1, the way the system manages security is by changing and giving access rights to different plugins. This feature lets Administrator user adjust user permissions in detail. It means Administrator user can limit or allow access to different parts of the system depending on what each user needs to do according to *FMT_MSA.3*. This thorough method makes sure the system follows security rules properly. It helps control who can access what and strengthens the overall security setup of the system. Users can perform management functions as defined in *FMT_SMF.1* as applicable to their role *FMT_SMR.1*. An MSSP user can smoothly switch into their Client user accounts whenever necessary to carry out tasks on behalf of the Client user. In contrast, an Administrator user lacks the capability to switch accounts and must log in separately to perform administrative tasks.

TOE Security Functional Requirements Satisfied: *FMT_MSA.1, FMT_MSA.3, FMT_SMF.1, FMT_SMR.1*

6.1.5 TOE Access

The TSF provides a method for controlling the establishment of a user's session based on a termination of session after a specified period of user inactivity. Sessions are logged out after 10 hours (if active) and 15 minutes (if in active) automatically. The TOE also allows user-initiated termination of the user's own interactive session. The TOE can deny session establishment of users with inactive(disable) status. To authenticate again, the Administrator user allows the user to change the user's status.

TOE Security Functional Requirements Satisfied: *FTA_SSL.3, FTA_SSL.4, FTA_TSE.1*

6.1.6 Protection of Security Functionality

The TOE protects the private server keys used for encryption and signing purposes by an unauthorized entity. The TOE stores all user's secrets in non-plaintext form preventing them from reading. Data transferred internally to TOE is secured with authentication and by preventing public exposure. Each event is recorded with timestamps to be referenced in future if required.

TOE Security Functional Requirements Satisfied: FPT_STM.1, FPT_ITT.1

REFERENCES

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

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

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

[4] Common Criteria for Information Technology Security Evaluation, Evaluation Methodology; Version 3.1, Revision 5, CCMB-2017-04-004