AppGate SDP v5.2

Security Target

Version 1.2

2020-12-04

Prepared for:

AppGate

Kungsgatan 34
411 519 Göteborg
Sweden

Global Headquarters:
BAC Colonnade Office Towers
2333 Ponce De Leon Blvd, Suite 900
Coral Gables, FL 33134

Prepared by:

Common Criteria Testing Laboratory
6841 Benjamin Franklin Drive
Columbia, Maryland 21046

Revision History		
Date	**Author**	**Modifications**
2019-12-18	Leidos	Initial draft
2020-03-10	Leidos	Second draft
2020-06-15	Leidos	Updated per latest vendor information.
2020-07-15	Leidos	Updated per evaluator comments.
2020-08-25	Leidos	Updated company name and logo
2020-09-15	Leidos	Updated per evaluator comments.
2020-09-17	Leidos	Updated per latest vendor information.
2020-12-04	Leidos	Updated per the evaluator comments.

Table of Contents

List of Figures and Tables

1 Security Target Introduction

This section introduces the Target of Evaluation (TOE) and provides the Security Target (ST) and TOE identification, ST and TOE conformance claims, ST conventions, glossary and list of abbreviations.

The TOE is the AppGate SDP 5.2.0. AppGate SDP provides capabilities to control access of network-based users to network resources in physical, cloud-based and hybrid environments, using the approach to computer security known as the Software Defined Perimeter (SDP).

A Software-Defined Perimeter is built on three core principles:

- Identity-centric - It is designed around the user, addressing the perimeter-less enterprise. Users are authenticated before they are allowed to connect to a network.
- Zero-Trust - It enforces the "Zero Trust" model so that anyone attempting to access a resource must authenticate first. All unauthorized resources are invisible. This applies the principle of least privilege to the network and reduces the attack surface.

  By default, users are not allowed to connect to anything – this is the opposite of traditional corporate networks, where once a user is given an IP address, they have access to everything in the network. Instead, Zero Trust ensures that once proper access criteria are met, a dynamic one-to-one connection is generated from the user's machine to the specific resource needed. Everything else is completely invisible.
- Cloud-centric - The Software-Defined Perimeter (SDP) is built for the cloud and has no centralized network chokepoint. It is completely distributed and as scalable as the internet itself. An SDP is engineered to operate natively in cloud networks. It is not simply a modified perimeter-based device that has been placed into a virtual machine. In addition, it is compatible with existing corporate networks.

The principle of operation is that Gateways are deployed in front of networked resource (application and server) infrastructure, effectively making it invisible on the network. A Controller defines access rights for users and devices (collectively, the Clients) on an individual basis. A Client establishes a secure TLS tunnel to the Controller, which authenticates the user. This process is based on verifying user claims within each session—including device posture and identity—before issuing Entitlement tokens to the user. The Client passes the issued Entitlement tokens on to the Gateways, which provision a firewall instance just for that user. The Gateway then translates the Entitlements into a set of individualized firewall rules. For each packet received from the Client, the correct rules allow, conditionally allow or block access to the network resources protected by the Gateway.

This ST includes the following additional sections:

- TOE Description (Section 2)—provides an overview of the TOE and describes the physical and logical boundaries of the TOE
- Security Problem Definition (Section 3)—describes the threats and assumptions that define the security problem to be addressed by the TOE and its environment
- Security Objectives (Section 4)—describes the security objectives for the TOE and its operational environment necessary to counter the threats and satisfy the assumptions that define the security problem

- IT Security Requirements (Section 5)—specifies the security functional requirements (SFRs) and security assurance requirements (SARs) to be met by the TOE
- TOE Summary Specification (Section 6)—describes the security functions of the TOE and how they satisfy the SFRs
- Rationale (Section 7)—provides mappings and rationale for the security problem definition, security objectives, security requirements, and security functions to justify their completeness, consistency, and suitability.

1.1 Security Target, Target of Evaluation, and Common Criteria Identification

ST Title: AppGate SDP v5.2 Security Target

ST Version: 1.2

ST Date: 2020-12-04

TOE Identification: AppGate SDP v5.2.0

TOE Developer: AppGate

Evaluation Sponsor: AppGate

CC Identification: Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, April 2017

1.2 Conformance Claims

This ST and the TOE it describes are conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Components, Version 3.1 Revision 5, April 2017
  - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security Assurance Components, Version 3.1 Revision 5, April 2017
  - Part 3 Conformant.

This ST and the TOE it describes are conformant to the following package:

- EAL2 augmented (ALC_FLR.1 Basic flaw remediation).

1.3 Conventions

The following conventions are used in this document:

- Security Functional Requirements—Part 1 of the CC defines the approved set of operations that may be applied to functional requirements: iteration; selection; assignment; and refinement.
  - Iteration—allows a component to be used more than once with varying operations. In this ST, iteration is identified with a number in parentheses following the base component identifier. For example, iterations of FCS_COP.1 are identified in a manner similar to FCS_COP.1(1) (for the component) and FCS_COP.1.1(1) (for the elements).

  - Selection—allows the specification of one or more elements from a list. Selections are indicated using bold italics and are enclosed by brackets (e.g., [***selection***]).
  - Assignment—allows the specification of an identified parameter. Assignments are indicated using bold text and are enclosed by brackets (e.g., [**assignment**]). An assignment within a selection is identified in underlined italics and with embedded bold brackets (e.g., [***[selected-assignment]***]).
  - Refinement—allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …").
- Other sections of the ST—other sections of the ST use bolding and/or different fonts (such as `Courier`) to highlight text of special interest, such as captions, commands, or filenames specific to the TOE.

1.4 Glossary

This ST uses a number of terms that have a specific meaning within the context of the ST and the TOE. This glossary provides a list of those terms and how they are to be understood within this ST.

Table 1: Terms and Definitions

Term	Definition
Action	An element of an Entitlement, specifying an IP access protocol and network resources, along with any additional protocol specific options such as ports or types, and the action to be taken (Allow, Block, or Alert) on matching network traffic.
Appliance	"Appliance" refers to the virtual or physical instance on which the system is running. Each appliance is a stateless, configurable machine that can operate as a Controller, Gateway, or LogServer or some combination thereof in a single appliance.
Claims	Key-value pairs that relate to the identity and context of the user and device. Claims may be generated from the Client (e.g., username), by the Controller (e.g., Active Directory attributes), from the client device (e.g., operating system) or from 3rd parties (e.g., location).
Client	The TOE component responsible for interfacing with the user, passing tokens to the Gateway, and establishing secure connections with Gateways as required.
Client Profile	A Client profile includes the profile name plus the minimum set of information required for a user to be able to sign-in to an AppGate SDP Controller.
Client Profile Link	The profile link permits of the distribution the Client profile for use with any type of Client. The profile link provides the Clients with all the information needed to access the Controllers
Collective	A group of appliances (Controller(s), Gateways and LogServers) all configured and managed from the same Controller.
Controller	The central point of administration for an AppGate SDP deployment. The first appliance in the deployment must always be configured as a Controller. Trust between Client and appliances within the Collective is based on the CA [Certificate Authority]. A self-signed certificate is generated when the first Controller is started and this is used to establish the Controller as the trusted authority for all tokens, certificates and TLS connections. Appliance seed files include CA-generated temporary certificates. The Client profile link includes the CA fingerprint.

Term	Definition
Entitlement	An Entitlement defines the rules for controlling access to network resources on a particular Site.
Entitlement Token	An Entitlement token contains a list of all the user's Entitlements for a specific Site as well as a list of the Gateways associated with that Site and the defined weighting of each Gateway. The Entitlement token is created by the Controller following a request from the Client. The Controller uses the assignment criteria in Policies to find all services that the user is entitled to, and replies to the Client with Entitlement tokens - one per Site. The Client sends the relevant Entitlement token to the appropriate Gateway, once it is connected.
Gateway	The policy enforcement point in the AppGate SDP deployment. The Gateway is responsible for controlling user access to network resources. The Gateway uses Claims and Entitlement information from each user to manage firewall rules and provide real-time access control.
Live Entitlements	AppGate SDP replaces static access rules with Live Entitlements. Live Entitlements are dynamic, context-aware security attributes that confirm user identity while providing the flexibility necessary to adjust to changing variables, such as environmental/infrastructure changes, user location, time of day, and workload sensitivity. Live Entitlements maintain security without manual interactions, often needed when modifying traditional static firewall rules. In AppGate SDP, Live Entitlements automatically update based on monitoring changes to the user context (such as an opened service ticket) or changes associated with the dynamic nature of a public or hybrid cloud environment.
Policy	AppGate SDP uses policies to assign rights to a user or group of users. Rights consist of Entitlements, Device Security settings, and Admin Roles. AppGate SDP assigns policies using claims-based expressions that identify the users or user groups to whom the Policy can be assigned.
Port Knocking	Port knocking is a method of externally opening ports on a firewall by generating a connection attempt on a set of pre-specified closed ports. Once a correct sequence of connection attempts is received, the firewall rules are dynamically modified to allow the host that sent the connection attempts to connect over specific port(s). The primary purpose of port knocking is to prevent an attacker from scanning a system for potentially exploitable services by doing a port scan. Single Packet Authorization (SPA) provides only a single "knock", consisting of an encrypted packet.
Seed File	The required configuration settings are set in the Admin UI for a new appliance where the seed file is generated. The JSON (or ISO) seed file remains valid for 24 hours only. To activate the new appliance, the seed file is exported (a temporary initialization file containing information to allow trusted peer-to-peer communication to take place). After a new appliance has been seeded, it will establish communication towards the Controller. As part of the registration it will get a signed certificate and change its status from pending to active.
Seeding	The process of passing the seed file onto the appliance. After successful seeding, the new appliance will show as *Active* on the appliances console.
Site	A group of networks or resources reachable from a Gateway or a group of Gateways. For example, a Site could contain all the networks within a Virtual Private Cloud (and all the resources hosted there), and may also contain the Gateways themselves that are used to connect to those resources.

Term	Definition
Single Packet Authorization (SPA)	SPA is a technology, a version of port knocking, used to enforce the "authenticate-first, connect second" approach. SPA cloaks infrastructure so that it is invisible to port scans. It ensures that only authorized users can connect to network resources. This reduces the attack surface and improves security.

1.5 Abbreviations and Acronyms

Table 2: Abbreviations and Acronyms

Abbreviation	Definition
AES	Advanced Encryption Standard—a symmetric encryption algorithm
API	Application Programming Interface
GCM	Galois/Counter Mode—a mode of operation for AES
GPG	GNU Privacy Guard—a cryptographic software suite
HMAC	Keyed-Hash Message Authentication Code
HTTPS	Hypertext Transfer Protocol Secure
IOPS	Input/output operations per second—an input/output performance measurement used to characterize computer storage devices.
LDAP	Lightweight Directory Access Protocol
MFA	Multi-Factor Authentication
NTP	Network Time Protocol
OTP	One-Time Password
RADIUS	Remote Authentication Dial-In User Service
REST	Representational State Transfer—a software architectural style that defines a set of constraints to be used for creating Web services.
RSA	Rivest-Shamir-Adleman—an asymmetric cryptographic algorithm
SAML	Security Assertion Markup Language
SAR	Security Assurance Requirement
SDP	Software Defined Perimeter
SFR	Security Function
SHA	Secure Hash Algorithm
SPA	Single Packet Authorization
SQL	Structured Query Language—programming language used for managing data held in a relational database management system.
ST	Security Target
TLS	Transport Layer Security
TOE	Target of Evaluation
TCP	Transmission Control Protocol

Abbreviation	Definition
UDP	User Datagram Protocol
UI	User Interface

2 TOE Description

2.1 TOE Overview

The TOE is the AppGate SDP v5.2.0. AppGate SDP enables network administrators to establish a Software Defined Perimeter (SDP) to control access by network-based users to network resources in physical, cloud-based and hybrid environments.

AppGate SDP implements an information flow control policy that mediates network traffic between users and network resources, based on Claims (key-value pairs that relate to the identity and context of the user and the user's device) and Entitlements (the rules for controlling access to network resources).

AppGate SDP Gateways are deployed in front of networked resource infrastructure. An AppGate SDP Controller defines access rights for users and devices (collectively, the Clients). An AppGate SDP Client establishes a secure TLS tunnel to the Controller, which authenticates the user. The Controller verifies user claims and issues Entitlement tokens to the user. The Client submits the Entitlement tokens to the Gateways, which control the user's subsequent access to network resources based on the granted Entitlements.

All users are identified and authenticated by the Controller prior to being granted a Claims token. AppGate SDP provides a local password-based authentication mechanism and can be configured to support remote authentication of client users and administrators using LDAP, RADIUS and SAML.

AppGate SDP generates audit records of security relevant events and can be configured to export generated audit records over a TLS channel to an external syslog server for audit storage and review.

Communications between Controllers, Gateways and Clients are protected using TLS v1.2.

2.2 TOE Architecture

AppGate SDP comprises an appliance component and a client software component installed on a user's device, such as a workstation, laptop, or mobile platform.

- **Client** – runs on each user's device which makes access requests to the Controller.
  After (or at) installation, a Client profile must be used to configure the Client before it is able to connect to a Controller. A Client profile includes the profile name plus the minimum set of information required for a user to be able to sign-in to an AppGate SDP Controller. The resulting elements are:
    - a URL for the Controller(s)
    - an assigned identity provider against which the user will authenticate
    - the profile name (which appears in the Client)
    - SPA details including the key required to establish a TCP/UDP connection to the Controller
    - a means to verify the CA fingerprint to ensure the Controller is genuine

The first two items are what define a unique profile. If a unique profile is updated (changed name, SPA or CA), then it will REPLACE any existing profile. If another unique profile is created (different URL or IdP) then it will CO-EXIST with any existing unique profiles.

When a Client profile is provided to the AppGate SDP Client it is automatically configured and the users are taken via the acceptance form straight to the sign-in form.

The AppGate SDP appliance is a stateless, configurable component that can operate in the following roles:

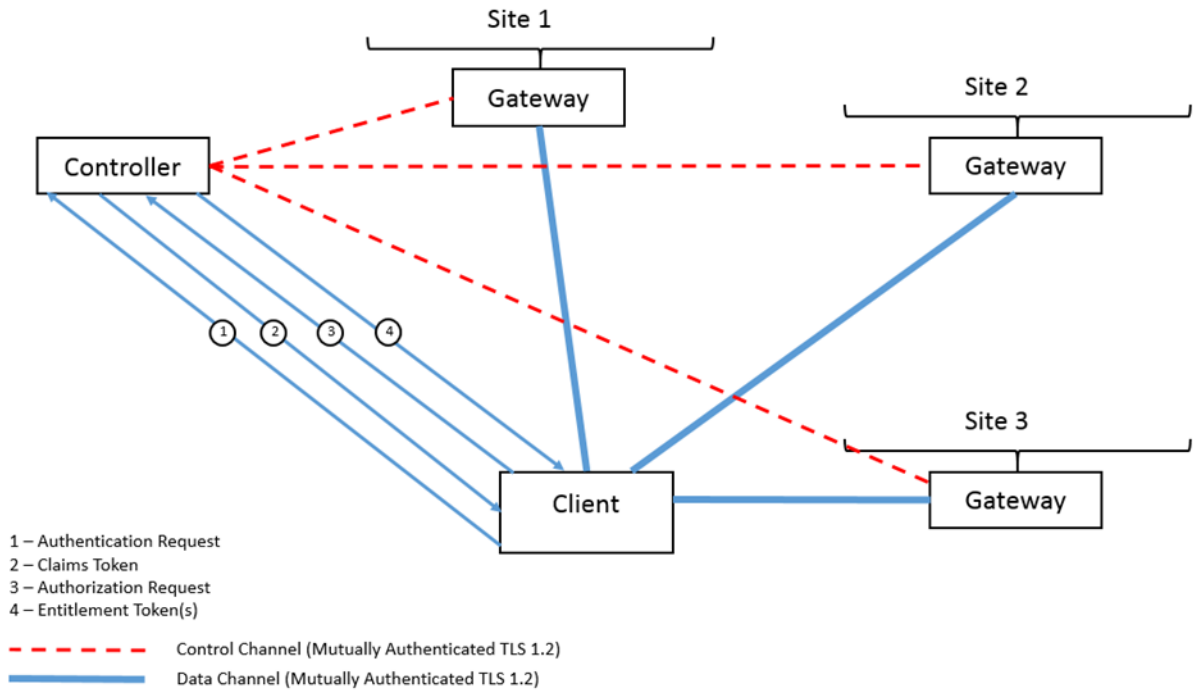
- **Controller**—the central point of administration for the AppGate SDP deployment. It includes an internal database for the storage of system configuration data and provides the following capabilities:
  - Certificate Authority (CA) for the deployment
  - Creation and signing of tokens used for authentication, authorization, and Policy distribution
  - Authentication of administrators logging in via the Admin User Interface (UI) and REST API[1], and users logging in via the Client
  - Assignment of Policies to users and creation of the list of Entitlements for each user
  - Assignment of roles to administrators and enforcement of privileges
- **Gateway**—The Gateway is the enforcement point, responsible for controlling user access to protected resources. After seeding it registers with the Controller and will then be listed as available to receive Client connections. Once registered it runs as a stateless appliance only needing to receive the token revocation list from the Controller. Even after a loss of connectivity to the Controller (maybe due to network problems) it will continue to serve both existing and new connections for a period of 12 hours, after which user access will no longer be allowed through that Gateway.
  The Gateway uses the Claims and Entitlement tokens from each user to manage firewall rules and provide real-time access control. User interactions allow the Gateway to elevate privilege levels on-demand. Name resolvers automatically resolve any firewall rules based on real-time information from the Cloud environments. After a user's access session has ended (eg. their Entitlement token has been revoked or has expired), no data about the user, their Entitlements, claims, firewall rules, etc., is retained.
  A Gateway can be enabled on the same appliance as the Controller, or on a separate appliance.
- **LogServer**—collects logs from the Controllers and Gateways to provide an audit trail of actions and user access. A LogServer is typically configured on an existing Controller appliance but can also be stand-alone where Controller performance is critical. Alternatively, log files can be exported using `rsyslog` to an external log server. The audit logs are not intended to remain on or be viewed on the appliance, so in the evaluated configuration they are exported to an external audit server using `rsyslog`. Note, the LogServer role is excluded from the evaluated configuration.

An AppGate SDP deployment can consist of a single appliance configured to perform all of these roles, or multiple appliances with at least one Controller and at least one Gateway (note, a multi-appliance

[1] The REST API comprises a set of Controller APIs, accessible via the peer interface (port 444 by default) of a Controller appliance.

deployment is known as a Collective[2]). The first appliance in the Collective must be configured as a Controller and will host the Admin UI. The Controller holds policy and configuration data in an internal SQL database. Additional appliances can be added to the Collective using the Admin UI. At least one Gateway is required to control access to a Site, and multiple Gateways can be deployed on a single Site to improve performance and availability. Figure 1 depicts an example deployment of AppGate SDP.

Figure 1: Example TOE Deployment



AppGate SDP appliances protect themselves using Single Packet Authorization (SPA). This effectively makes the appliances invisible on the network. In order to establish a connection to an appliance (Controller or Gateway), a Client must first send an SPA packet to the appliance. The SPA packet includes a time stamp to protect against replay attacks and is encoded using a pre-shared key known only to the Client and the appliance. Once the appliance verifies the SPA, a TLS tunnel is established between the Client and appliance.

Before the first authentication, the Client will have a URL to one of the Controllers, and the Certificate Authority ("CA") certificate, either because these were automatically installed with the Client, or because the user entered the Controller URL manually and the Client downloaded the CA certificate. The

[2] Although a Collective can support up to 6 Controllers (for purposes of High Availability), the scope of the evaluation is restricted to a single Controller deployment.

Client will use the CA certificate in all communications to verify the authenticity of the Controller or Gateway.

1. The Client device makes an authentication request to the Controller using SPA ("1" in Figure 1). The Controller verifies the credentials (e.g., username and password) submitted in the authentication request against a user directory (e.g., LDAP, not depicted in Figure 1). The Client will generate a random ID for itself if it does not already have one. This random ID will be sent along with the credentials.
2. On a successful authentication, the Controller replies with a Claims token ("2" in Figure 1). The Claims token is issued to a Distinguished Name ("DN") which includes both the user's ordinary DN and the randomly-generated Client ID. This new DN will identify the session throughout the system.
3. The Client uses the Claims token to request Entitlement tokens ("3" in Figure 1).
4. The Controller queries its internal database to find all services that the user is entitled to, and replies to the Client with Entitlement tokens ("4" in Figure 1), one per Site.
5. The Client generates an RSA key pair and sends a certificate signing request to the Controller, based on the Client's Claims token and public key. The Controller replies with a certificate generated by the Certificate Authority (CA) hosted by the Controller and issued to the DN in the Claims token.
6. The Client establishes a TLS tunnel to one Gateway at each Site, using the Client certificate for Client-side TLS authentication, and sends the Claims token, a number of device claims, and a site-specific Entitlement token.
7. Each Gateway translates the Claims token, device claims and Entitlement token into a list of firewall rules, which apply to that user on that Client computer at that specific time.
8. The Client can now send IP packets to endpoint servers through the TLS tunnel, provided that this is allowed by the firewall rules.

AppGate SDP uses **Policies** to assign rights to a user or group of users. These include **Entitlements**, **Device Security** settings, and any **Admin Roles**. Policy assignment is done using claims-based expressions identifying who the Policy should be assigned to. The Controller manages this process and the resulting configurations are then granted to the user in the form of:

- Site-based Entitlement token(s), comprising the list of Entitlements and **Conditions**
- Device Security settings, which are pushed to the connected Client
- An Admin token (for an administrator), comprising the list of Admin Roles allowed.

Each Entitlement defines the rules for controlling access to network resources on a particular Site. The main elements of an Entitlement are the Site, **Actions** (traffic protocols, target hosts, ports), and any Conditions that must be met for those Actions to be allowed by the Gateway.

Typically, Entitlements are used to <ALLOW> user traffic to a network resource. However, an Entitlement can also <BLOCK> traffic, or be used to trigger an <ALERT>.

Conditions can also be used to create **remedy methods**—alternative criteria that would unblock access if the user's other claims don't satisfy the requirements. For example, if working from home then providing multi-factor authentication could be an alternative method for gaining access to particular network resources.

The Gateway uses the Entitlements and Conditions to configure firewall rules for each user. If the user's claims match the access criteria in the Condition, then the Entitlement will become live. If the claims don't meet the criteria then the Condition will not pass. In this situation, any remedy method that has been set up in the Condition will be triggered. The Gateway regularly re-evaluates the Conditions and reconfigures firewall rules in response to changes in the user's claims. A Condition can be used to force additional re-evaluations to ensure the Gateway responds quickly to relevant changes.

2.3 Physical Boundaries

2.3.1 TOE Components

AppGate SDP comprises an appliance software component and a client software component installed on a user's device, such as a workstation, laptop, or mobile platform. In its evaluated configuration, the AppGate SDP appliance is available as a virtual appliance. This is typically supplied as installation ISO images that can be downloaded from the AppGate website and run in the customer's chosen environment.

2.3.1.1 Virtual Appliance

The virtual AppGate SDP appliance is supplied as an installation ISO image that can be downloaded from the AppGate website and run in the customer's chosen environment. The AppGate SDP appliance is currently supported on VMware ESXI 5.5 or higher, VMware workstation, Microsoft Hyper-V, and Oracle VirtualBox.

AppGate recommends the following minimum specifications for platforms supporting the virtual AppGate SDP appliance:

- Memory size—each appliance has some built-in checks for available memory and this is compared to the roles selected to see if there is enough. Where there is insufficient memory, a Dashboard Admin Message will be displayed—either an Info message if the memory is > 85% of the recommended minimum values or an Error message if < 85%:
  - Standalone Controller or Gateway—minimum 2 GB
  - Controller with Gateway—minimum 3 GB

  It can be expected that a well-used production system is likely to require 4GB for an appliance. Controllers are transactional so only use memory while issuing tokens. Since they can only handle some 10s of transactions simultaneously, memory usage will never grow. For Gateways, memory requirements depend on the number of users and the number of Actions per user. Assume 1MB (for < 10 Actions) to 5MB (for > 1000 Actions) per user. For a standalone 2,000 user Gateway, 300 Actions per user would be the limit for a 4GB Gateway. A 5,000 user Gateway with 2,000 Actions per user would need 16GB.
- Disk size—20GB disk space should be sufficient for both the Controller and the Gateway.

2.3.1.2 Client Software

The client software is supported on the following platforms:

- Windows 7 SP1 and later, with .NET 4.5 and later
- Apple OSX/macOS 10.12.3 or newer
- Ubuntu 16.04 and later with network manager 1.2.4 or later and gnome-keyring
- Fedora 30 & 31 + gnome-keyring. Uses broad SELinux settings (affects default security)
- iOS 11 or later
- Android 6 or later
- Chrome OS 78 and later.

Client software for Windows, Apple OSX/macOS, Ubuntu, and Fedora is downloadable from the AppGate support web site, while the mobile client is available from the appropriate app store.

The client software is also available in a single sign-on (SSO) deployment for Windows and in headless deployments for Windows, Ubuntu, Fedora, and RHEL7/CentOS.

2.3.2 Operational Environment Components

AppGate SDP requires the following components in its operational environment:

- NTP server—provides time synchronization to the Controllers and Gateways in a Collective
- Syslog server—provides storage and review capabilities for audit records generated and transmitted by Collectors and Gateways.

AppGate SDP also supports the following optional components in its operational environment:

- DNS server—provides host name resolution services for AppGate SDP appliances
- LDAP, RADIUS and SAML identity providers.

Administrators access the Admin UI remotely using a web browser that supports TLS v1.2.

Note, the AppGate SDP virtual appliance also provides a command line interface (CLI), accessible via Secure Shell (SSH), that administrators use for initial installation and configuration. It is not intended for use once AppGate SDP is in its evaluated configuration. As such, use of the CLI and of SSH to access the CLI are excluded from the scope of evaluation.

2.4 Logical Boundaries

This section summarizes the security functions provided by the TOE.

2.4.1 Security Audit

AppGate SDP generates audit records of security relevant events occurring on the Collector and Gateways. Each audit log includes the following information: date and time of the event; type of event; subject identity; and the outcome of the event.

The generated audit records are stored locally on the appliance, which can be configured to export the audit records to an external syslog server over a secure communications channel. AppGate SDP will overwrite the oldest stored audit records if the local storage space is exhausted.

2.4.2 Cryptographic Support

AppGate SDP incorporates OpenSSL 1.0.2u with OpenSSL FIPS Object Module 2.0.16 to support its TLS implementation, including symmetric encryption and decryption using AES in GCM mode, digital signature generation and verification using RSA, cryptographic hashing using SHA-256 and SHA-384, and keyed-hash message authentication using HMAC-SHA-256 and HMAC-SHA-384. The OpenSSL FIPS Object Module destroys cryptographic keys when they are no longer needed.

AppGate SDP also incorporates the Bouncy Castle Java API v1.65, which provides the cryptographic operations to encrypt token data and sign tokens and generate appliance and Client certificates.

2.4.3 User Data Protection

AppGate SDP implements an information flow control policy that mediates information flow between Clients (comprising an identified user together with a provisioned client device) and Destinations (comprising network-based resources), based on configured Entitlements and Claims.

All communications between Clients and Gateways are protected using mutually authenticated TLS 1.2.

2.4.4 Identification and Authentication

AppGate SDP requires all users to be successfully identified and authenticated before allowing them to perform any other activities. AppGate SDP provides a local password-based authentication mechanism and can be configured to support remote authentication of client users and administrators using LDAP, RADIUS and SAML.

AppGate SDP will lock a local user out of their account for a one minute interval following an administrator-configurable number of consecutive failed authentication attempts.

2.4.5 Security Management

AppGate SDP provides a built-in System Administrator role that has full system administration privileges. The System Administrator manages AppGate SDP via a browser-based Admin UI that provides the System Administrator capabilities to manage AppGate SDP's security functionality.

2.4.6 Protection of the TSF

All communications between distributed AppGate SDP components (clients, Controller, Gateways) is over TLS 1.2. Communications between the client and Controller is TLS (with initial authentication request using HTTPS), while communications between clients and Gateways and between Controller and Gateways is over mutually authenticated TLS. Communications from Client to Controller and from client to Gateways also uses SPA.

In order to operate correctly, AppGate SDP appliances need to be able to establish connections to external NTP servers. Synchronized/coordinated time is critical for AppGate SDP operation because of the use of time-stamped tokens. AppGate SDP currently uses NTP v4.2.8 (RFC 5905).

AppGate SDP provides a trusted update mechanism consistent with the trusted update requirements in the Network Device collaborative Protection Profile (although the TOE is not claiming conformance to this PP.) The digital signature used is GPG (Gnu PG), configured to use RSA with a 2048 bit key.

2.4.7 TOE Access

Users who access AppGate SDP are presented with a warning banner prior to being granted access to the system. Additionally, users who have logged into the TOE are able to terminate their sessions.

2.4.8 Trusted Path/Channels

AppGate SDP provides a trusted path for administrators to communicate with the Controller. The trusted path is implemented using HTTPS for access to the Admin UI. Administrators initiate the trusted path by establishing an HTTPS connection using a supported web browser. The trusted path is used for initial authentication and all subsequent administrative actions. The use of HTTPS ensures all communication over the trusted path is protected from disclosure and modification.

AppGate SDP is able to establish trusted channels with external IT entities to support remote authentication of users and export of audit records to an audit server. These trusted channels are implemented using TLS v1.2.

2.5 TOE Documentation

The guidance documentation included in the TOE is available on-line at the following URLs:

https://sdphelp.appgate.com/adminguide/v5.2/introduction.html

The user guide for Clients can be found here:

https://sdphelp.appgate.com/userguide/v5.2/

3 Security Problem Definition

This section defines the security problem to be addressed by the TOE, in terms of threats to be countered by the TOE or its operational environment, and assumptions about the intended operational environment of the TOE.

3.1 Assumptions

This section contains assumptions regarding the operational environment and the intended usage of the TOE.

A.MANAGE There will be one or more competent individuals assigned to manage the TOE and the security of the information it contains.

A.PROTECT The TOE components critical to security policy enforcement will be protected from unauthorized physical modification.

A.HOSTNAME_RESOLUTION The operational environment must ensure that security measures are in place to protect DNS hostname resolution.

3.2 Threats

This section identifies and describes the threats to be countered by the TOE and its operational environment.

T.BRUTE_FORCE An unauthorized user may gain access to the TOE through repeated password-guessing attempts.

T.INAPPROPRIATE_USE Authorized users perform inappropriate actions on the TOE due to ignorance of their responsibilities or operational policies and procedures.

T.NETWORK_ACCESS An attacker attempts to gain access to networked resources and compromise the data or services provided by those resources.

T.NETWORK_COMPROMISE An unauthorized user may monitor the enterprise network in an attempt to obtain sensitive data, such as passwords, or to modify transmitted data.

T.NO_ACCOUNTABILITY Authorized users of the TOE perform adverse actions on the TOE, or attempt to perform unauthorized actions, which go undetected.

T.UNATTENDED_SESSION An unauthorized user gains access to the TOE via an unattended authorized user session.

T.UNAUTHORIZED_ACCESS An unauthorized user may gain access to the TOE security functions and data.

T.UNAUTHORIZED_ACTIVITY	Authorized users perform unauthorized actions on the TOE.
T.UPDATE_COMPROMISE	An attacker may attempt to provide a compromised update of the software or firmware that undermines the security functionality of the device.

4 Security Objectives

This section identifies the security objectives for the TOE and its operational environment. The security objectives identify the responsibilities of the TOE and its environment in addressing the security problem defined in Section 3.

4.1 Security Objectives for the TOE

The following are the TOE security objectives:

O.AUDIT	The TOE shall be able to generate audit records of security-relevant events, identifying users causing the events as applicable.
O.AUDIT_STORAGE	The TOE shall protect stored audit records from unauthorized modification and deletion and shall preserve the most recent stored audit records in the event audit storage is exhausted. The TOE additionally shall provide a means to securely export audit records to an external audit server for long term storage and review.
O.CRYPTOGRAPHY	The TOE shall perform cryptographic operations to support protocols used to protect data in transit.
O.I_AND_A	The TOE shall require all users of the TOE to be identified and authenticated before gaining access to TOE services.
O.INFORMATION_FLOW	The TOE shall mediate the transmission of network packets between users and network resources based on user identity, and authorization to access network resources based on IP address, protocol and port number.
O.LOGON_BANNER	The TOE shall be able to display an advisory warning message to potential users pertaining to appropriate use of the TOE.
O.PROTECTED_COMMS	The TOE shall protect communications between its distributed components, and between itself and external entities, from disclosure and modification.
O.SECURITY_MANAGEMENT	The TOE shall restrict the ability to perform security management functions on the TOE to authorized administrators.
O.SESSION_TERMINATION	The TOE shall provide mechanisms to terminate a user session at the request of the user.

O.THROTTLE	The TOE shall limit the rate at which consecutive unsuccessful authentication attempts can be performed.
O.TRUSTED_UPDATE	The TOE shall apply cryptographic means to verify the integrity and authenticity of software or firmware updates prior to applying those updates.

4.2 Security Objectives for the Operational Environment

The following are the security objectives for the operational environment of the TOE:

OE.PERSONNEL	Those responsible for the TOE must ensure that personnel working as authorized administrators have been carefully selected and trained for proper operation of the TOE.
OE.PHYSICAL	Those responsible for the TOE must ensure that those parts of the TOE critical to security policy are protected from any physical attack.
OE.HOSTNAME_RESOLUTION	The operational environment is responsible for DNS hostname resolution. As such, strategies in the operational environment must be implemented to provide security protections.

5 IT Security Requirements

This ST defines the following extended components for use within this ST:

5.1 Extended Components Definition

This ST defines the following extended components for use within this ST:

- FAU_EXP.1—Audit data export
- FPT_TUD.1—Trusted update.

5.1.1 Security Audit (FAU)

5.1.1.1 Audit Data Export (FAU_EXP.1)

Family Behavior

This family defines requirements for the TSF to be able to securely transmit audit data to an external IT entity.

Component Leveling

At FAU_EXP.1 Audit data export, the TSF provides the capability to export audit data to an external IT entity.

Management: FAU_EXP.1

The following actions could be considered for the management functions in FMT:

a) configuration of the trusted channel for exporting audit records.

Audit: FAU_EXP.1

There are no auditable events foreseen.

FAU_EXP.1 Audit data export

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation
FTP_ITC.1 Inter-TSF trusted channel

FAU_EXP.1.1 The TSF shall be able to transmit generated audit data to an external IT entity using a trusted channel.

5.1.2 Protection of the TSF (FPT)

5.1.2.1 Trusted Update (FPT_TUD.1)

Family Behavior

This family defines the requirements for securely updating the TOE firmware and/or software.

Component Leveling

FPT_TUD: Trusted update	1

At FPT_TUD.1 Trusted update, the TSF provides mechanisms for determining the current TOE version, checking a trusted source for TSF updates, verifying the integrity of updates, and initiating the update process.

Management: FPT_TUD.1

The following actions could be considered for the management functions in FMT:

a) Ability to update the TOE and to verify the updates.

Audit: FPT_TUD.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Initiation of the update process.

b) Minimal: Any failure to verify the integrity of the update.

FPT_TUD.1 Trusted update

Hierarchical to: No other components.

Dependencies: FCS_COP.1 Cryptographic operation
FMT_SMR.1 Security roles

FPT_TUD.1.1 The TSF shall provide [**assignment: *the authorized identified roles***] the ability to query the currently executing version of the TOE firmware/software.

FPT_TUD.1.2 The TSF shall provide [**assignment: *the authorized identified roles***] the ability to manually initiate updates to TOE firmware/software and [**selection: *support automatic checking for updates*, *support automatic updates*, *no other update mechanism***].

FPT_TUD.1.3 The TSF shall provide means to authenticate firmware/software updates to the TOE using a [**selection: *digital signature mechanism*, *published hash***] prior to installing those updates.

5.2 TOE Security Functional Requirements

This section specifies the security functional requirements (SFRs) for the TOE. SFRs were drawn from Part 2 of the Common Criteria v3.1 Revision 5, and from the extended components defined in Section 5.1 above.

Table 3: TOE Security Functional Components

Requirement Class	Requirement Component
FAU: Security audit	FAU_GEN.1 Audit data generation
	FAU_GEN.2 User identity association
	FAU_EXP.1 Audit data export
	FAU_STG.1 Protected audit trail storage
	FAU_STG.4 Prevention of audit data loss
FCS: Cryptographic support	FCS_CKM.1 Cryptographic key generation
	FCS_CKM.4 Cryptographic key destruction
	FCS_COP.1 Cryptographic operation
FDP: User data protection	FDP_IFC.1 Subset information flow control
	FDP_IFF.1 Simple security attributes
	FDP_ITT.1 Basic internal transfer protection
FIA: Identification and authentication	FIA_AFL.1 Authentication failure handling
	FIA_UAU.2 User authentication before any action
	FIA_UAU.5 Multiple authentication mechanisms
	FIA_UID.2 User identification before any action
FMT: Security management	FMT_MSA.1 Management of security attributes
	FMT_MSA.3 Static attribute initialisation
	FMT_REV.1 Revocation
	FMT_SAE.1 Time-limited authorisation
	FMT_SMF.1 Specification of Management Functions
	FMT_SMR.1 Security roles
FPT: Protection of the TSF	FPT_ITT.1 Basic internal TSF data transfer protection
	FPT_STM.1 Reliable time stamps
	FPT_TUD.1 Trusted update
FTA: TOE Access	FTA_SSL.4: User-initiated termination
	FTA_TAB.1: Default TOE access banners
FTP: Trusted path/channels	FTP_ITC.1 Inter-TSF trusted channel
	FTP_TRP.1 Trusted path

5.2.1 Security Audit (FAU)

FAU_GEN.1 Audit data generation

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

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

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

c) [**the following auditable events:**

- **Appliance started (equivalent to startup of the audit function)**
- **Appliance deactivated (equivalent to shutdown of the audit function)**
- **Administrator authorization succeeded and failed**
- **User authentication succeeded and failed**
- **Local user account locked out due to consecutive authentication failures**
- **User authorization succeeded and failed**
- **User blacklisted**
- **User blacklist entry removed**
- **Claims token accepted, rejected, expired**
- **Entitlement token, accepted, rejected, expired**
- **Remedy action triggered**
- **Remedy action authentication succeeded and failed**
- **Token revoked**
- **Client attempt to access Destination**
- **Tunnel from Gateway to Client established, closed**
- **Client session created, reconnected, removed**
- **System entity (Appliance, Condition, Entitlement, Identity Provider, Local User, Policy, etc.) created, modified, deleted, viewed**
- **Global settings updated**
- **Device on-boarded, deleted from system.**

].

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, 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, [**none**].

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.

FAU_EXP.1 Audit data export

FAU_EXP.1.1 The TSF shall be able to transmit generated audit data to an external IT entity using a trusted channel.

FAU_STG.1 Protected audit trail storage

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

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

FAU_STG.4 Prevention of audit data loss

FAU_STG.4.1 The TSF shall [***"overwrite the oldest stored audit records"***] and [**take no other action**] if the audit trail is full.

5.2.2 Cryptographic Support (FCS)

FCS_CKM.1(1) Cryptographic key generation (RSA key pairs)

FCS_CKM.1.1(1) The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [**IFC Key Pair Generation**] and specified cryptographic key sizes [**4096 bits**] that meet the following: [**FIPS 186-4, Appendix B.3**].

FCS_CKM.1(2) Cryptographic key generation (AES keys)

FCS_CKM.1.1(2) The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [**Counter DRBG (AES)**] and specified cryptographic key sizes [**128 bits, 256 bits**] that meet the following: [**ISO/IEC 18031:2011**].

FCS_CKM.4 Cryptographic key destruction

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [**overwriting with zeroes**] that meets the following: [**FIPS 140-2**].

FCS_COP.1(1) Cryptographic operation (symmetric cryptography)

FCS_COP.1.1(1) The TSF shall perform [**symmetric encryption and decryption**] in accordance with a specified cryptographic algorithm [**AES operating in GCM**] and cryptographic key sizes [**128 bits, 256 bits**] that meet the following: [**ISO 18033-3 (AES), ISO 19772 (GCM)**].

FCS_COP.1(2) Cryptographic operation (asymmetric cryptography)

FCS_COP.1.1(2) The TSF shall perform [**digital signature generation and verification**] in accordance with a specified cryptographic algorithm [**RSA**] and cryptographic key sizes [**2048 bits, 3072 bits, 4096 bits**] that meet the following: [**FIPS PUB 186-4**].

FCS_COP.1(3) Cryptographic operation (cryptographic hashing)

FCS_COP.1.1(3) The TSF shall perform [**cryptographic hashing**] in accordance with a specified cryptographic algorithm [**SHA-256, SHA-384, SHA-512**] and ~~cryptographic key~~ **message digest** sizes [**256 bits, 384 bits, 512 bits**] that meet the following: [**ISO/IEC 10118-3:2004**].

FCS_COP.1(4) Cryptographic operation (keyed-hash message authentication)

FCS_COP.1.1(4) The TSF shall perform [**keyed-hash message authentication**] in accordance with a specified cryptographic algorithm [**HMAC-SHA-256, HMAC-SHA-384**] and ~~cryptographic key~~ **message digest** sizes [**256 bits, 384 bits**] that meet the following: [**ISO/IEC 9797-2:2011**].

5.2.3 User Data Protection (FDP)

FDP_IFC.1 Subset information flow control

FDP_IFC.1.1 The TSF shall enforce the [**SDP Information Flow Policy**] on [

- **Subjects: Clients (comprising Users+Devices), Destinations**
- **Information: Network Traffic**
- **Operations: Transmit**

].

FDP_IFF.1 Simple security attributes

FDP_IFF.1.1 The TSF shall enforce the [**SDP Information Flow Policy**] based on the following types of subject and information security attributes: [

- **Subjects: Clients, Destinations**
- **Information: Network Traffic**
- **Subject security attributes: Distinguished Name, Entitlements, Claims**
- **Information security attributes: Destination IP Address, Destination Protocol, Destination Port**

].

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: [**A Client can Transmit Network Traffic to a Destination if the Network Traffic's Destination IP Address, Destination Protocol and Destination Port match an 'Allow' rule associated with the Client's Distinguished Name and derived from the Client's Entitlements**].

FDP_IFF.1.3 The TSF shall enforce the [**specific Condition associated with the 'Allow' rule, based on the Client's Entitlements and Claims**].

FDP_IFF.1.4 The TSF shall explicitly authorise an information flow based on the following rules: [**no explicit authorization rule**].

FDP_IFF.1.5 The TSF shall explicitly deny an information flow based on the following rules: [**the Network Traffic's Destination IP Address, Destination Protocol and Destination Port match a 'Block' rule associated with the Client's Distinguished Name**].

FDP_ITT.1 Basic internal transfer protection

FDP_ITT.1.1 The TSF shall enforce the [**SDP Information Flow Policy**] to prevent the [***disclosure***, ***modification***] of user data when it is transmitted between physically-separated parts of the TOE.

5.2.4 Identification and Authentication (FIA)

FIA_AFL.1 Authentication failure handling

FIA_AFL.1.1 The TSF shall detect when [***an administrator configurable positive integer within [1...99]***] unsuccessful authentication attempts occur related to [**local user password authentication**].

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been [***met***], the TSF shall [**lock the user account for 1 minute**].

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.

FIA_UAU.5 Multiple authentication mechanisms

FIA_UAU.5.1 The TSF shall provide [

- **a local password mechanism**
- **support for remote authentication via LDAP, RADIUS or SAML**
- **support for remote certificate-based authentication via LDAP**
- **support for additional Multi Factor Authentication**

] to support user authentication.

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

- **Locally defined users are authenticated using the password associated with the user's account**
- **Users defined in the administrator configured Client Profile can use an external LDAP or RADIUS server and are authenticated using the password associated with the externally managed account**
- **A user defined in the administrator configured Client Profile can use a valid certificate to authenticate via LDAP using the certificate**
- **Users defined in the administrator configured Client Profile can use a SAML identity provider to enter credentials and initiate the SAML authentication request. The Controller checks the Authentication, Attribute and Authorization Decision assertions in the SAML response to determine if authentication was successful**
- **Where configured, Multi Factor Authentication can be triggered:**
  - **By a remedy method in a Condition in a Client's Entitlement**
  - **As part of System Administrator authentication**
  - **During Device on-boarding**

].

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.

5.2.5 Security Management (FMT)

FMT_MSA.1 Management of security attributes

FMT_MSA.1.1 The TSF shall enforce the [**SDP Information Flow Policy**] to restrict the ability to [***modify***] the security attributes [**Entitlements**] to [**System Administrator**].

FMT_MSA.3 Static attribute initialisation

FMT_MSA.3.1 The TSF shall enforce the [**SDP Information Flow Policy**] to provide [***restrictive***] default values for security attributes that are used to enforce the SFP.

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

FMT_REV.1 Revocation

FMT_REV.1.1 The TSF shall restrict the ability to revoke [**Entitlements**] associated with the [***subjects***] under the control of the TSF to [**System Administrator**].

FMT_REV.1.2 The TSF shall enforce the rules [**as soon as the subject's Entitlements token is revoked**].

FMT_SAE.1 Time-limited authorisation

FMT_SAE.1.1 The TSF shall restrict the capability to specify an expiration time for [**Claims, Entitlements, Admin access**] to [**System Administrator**].

FMT_SAE.1.2 For each of these security attributes, the TSF shall be able to [

- **For Claims: Request user re-authentication**
- **For Entitlements: Update Entitlements**
- **For Admin access: Terminate administrator session**

] after the expiration time for the indicated security attribute has passed.

FMT_SMF.1 Specification of Management Functions

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

- **Manage Policies**
- **Manage Entitlements**
- **Manage Conditions**
- **Manage Sites**
- **Manage Identity Providers**
- **Manage Users**

- **Manage local user lockout policy**
- **Change local user passwords**
- **Configure token lifetimes**
- **Renew tokens**
- **Blacklist user**
- **Remove device**
- **Configure Appliances**

].

FMT_SMR.1 Security roles

FMT_SMR.1.1 The TSF shall maintain the roles [**System Administrator**].

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

5.2.6 Protection of the TSF (FPT)

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.

FPT_STM.1 Reliable time stamps

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

FPT_TUD.1 Trusted update

FPT_TUD.1.1 The TSF shall provide [**System Administrator**] the ability to query the currently executing version of the TOE firmware/software.

FPT_TUD.1.2 The TSF shall provide [**System Administrator**] the ability to manually initiate updates to TOE firmware/software and [***no other update mechanism***].

FPT_TUD.1.3 The TSF shall provide means to authenticate firmware/software updates to the TOE using a [***digital signature mechanism***] prior to installing those updates.

5.2.7 TOE Access (FTA)

FTA_SSL.4 User-initiated termination

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

FTA_TAB.1 Default TOE access banners

FTA_TAB.1.1 Before establishing a user session, the TSF shall display and advisory warning message regarding unauthorized use of the TOE.

5.2.8 Trusted Path/Channels (FTP)

FTP_ITC.1 Inter-TSF trusted channel

FTP_ITC.1.1 The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

FTP_ITC.1.2 The TSF shall permit [***the TSF***] to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for [**export of audit records to external audit server, transmission of user authentication data to external authentication server**].

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, [all subsequent user interactions]***].

5.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are the EAL2 augmented with ALC_FLR.1 Basic flaw remediation components as specified in Part 3 of the Common Criteria. No operations are applied to the assurance components.

Table 4: Assurance Components

Requirement Class	Requirement Component
ADV: Development	ADV_ARC.1: Security architecture description
	ADV_FSP.2: Security-enforcing functional specification
	ADV_TDS.1: Basic design
AGD: Guidance documents	AGD_OPE.1: Operational user guidance
	AGD_PRE.1: Preparative procedures
ALC: Life-cycle support	ALC_CMC.2: Use of a CM system
	ALC_CMS.2: Parts of the TOE CM coverage
	ALC_DEL.1: Delivery procedures
	ALC_FLR.1: Basic flaw remediation

Requirement Class	Requirement Component
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: Tests	ATE_COV.1: Evidence of coverage
	ATE_FUN.1: Functional testing
	ATE_IND.2: Independent testing – sample
AVA: Vulnerability assessment	AVA_VAN.2: Vulnerability analysis

5.3.1 Development (ADV)

ADV_ARC.1 Security architecture description

ADV_ARC.1.1D The developer shall design and implement the TOE so that the security features of the TSF cannot be bypassed.

ADV_ARC.1.2D The developer shall design and implement the TSF so that it is able to protect itself from tampering by untrusted active entities.

ADV_ARC.1.3D The developer shall provide a security architecture description of the TSF.

ADV_ARC.1.1C The security architecture description shall be at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE design document.

ADV_ARC.1.2C The security architecture description shall describe the security domains maintained by the TSF consistently with the SFRs.

ADV_ARC.1.3C The security architecture description shall describe how the TSF initialisation process is secure.

ADV_ARC.1.4C The security architecture description shall demonstrate that the TSF protects itself from tampering.

ADV_ARC.1.5C The security architecture description shall demonstrate that the TSF prevents bypass of the SFR-enforcing functionality.

ADV_ARC.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.2 Security-enforcing functional specification

ADV_FSP.2.1D The developer shall provide a functional specification.

ADV_FSP.2.2D The developer shall provide a tracing from the functional specification to the SFRs.

ADV_FSP.2.1C The functional specification shall completely represent the TSF.

ADV_FSP.2.2C The functional specification shall describe the purpose and method of use for all TSFI.

ADV_FSP.2.3C The functional specification shall identify and describe all parameters associated with each TSFI.

ADV_FSP.2.4C For each SFR-enforcing TSFI, the functional specification shall describe the SFR-enforcing actions associated with the TSFI.

ADV_FSP.2.5C For each SFR-enforcing TSFI, the functional specification shall describe direct error messages resulting from processing associated with the SFR-enforcing actions.

ADV_FSP.2.6C The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

ADV_FSP.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.2.2E The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

ADV_TDS.1 Basic design

ADV_TDS.1.1D The developer shall provide the design of the TOE.

ADV_TDS.1.2D The developer shall provide a mapping from the TSFI of the functional specification to the lowest level of decomposition available in the TOE design.

ADV_TDS.1.1C The design shall describe the structure of the TOE in terms of subsystems.

ADV_TDS.1.2C The design shall identify all subsystems of the TSF.

ADV_TDS.1.3C The design shall describe the behavior of each SFR-supporting or SFR non-interfering TSF subsystem in sufficient detail to determine that it is not SFR-enforcing.

ADV_TDS.1.4C The design shall summarise the SFR-enforcing behavior of the SFR-enforcing subsystems.

ADV_TDS.1.5C The design shall provide a description of the interactions among SFR-enforcing subsystems of the TSF, and between the SFR-enforcing subsystems of the TSF and other subsystems of the TSF.

ADV_TDS.1.6C The mapping shall demonstrate that all TSFIs trace to the behavior described in the TOE design that they invoke.

ADV_TDS.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_TDS.1.2E The evaluator shall determine that the design is an accurate and complete instantiation of all security functional requirements.

5.3.2 Guidance Documents (AGD)

AGD_OPE.1 Operational user guidance

AGD_OPE.1.1D The developer shall provide operational user guidance.

AGD_OPE.1.1C The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2C The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3C The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4C The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5C The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AGD_OPE.1.6C The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfil the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7C The operational user guidance shall be clear and reasonable.

AGD_OPE.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AGD_PRE.1 Preparative procedures

AGD_PRE.1.1D The developer shall provide the TOE including its preparative procedures.

AGD_PRE.1.1C The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer’s delivery procedures.

AGD_PRE.1.2C The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

AGD_PRE.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AGD_PRE.1.2E The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

5.3.3 Life-cycle Support (ALC)

ALC_CMC.2 Use of a CM system

ALC_CMC.2.1D The developer shall provide the TOE and a reference for the TOE.

ALC_CMC.2.2D The developer shall provide the CM documentation.

ALC_CMC.2.3D The developer shall use a CM system.

ALC_CMC.2.1C The TOE shall be labelled with its unique reference.

ALC_CMC.2.2C The CM documentation shall describe the method used to uniquely identify the configuration items.

ALC_CMC.2.3C The CM system shall uniquely identify all configuration items.

ALC_CMC.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_CMS.2 – Parts of the TOE CM coverage

ALC_CMS.2.1D The developer shall provide a configuration list for the TOE.

ALC_CMS.2.1C The configuration list shall include the following: the TOE itself; the evaluation evidence required by the SARs; and the parts that comprise the TOE.

ALC_CMS.2.2C The configuration list shall uniquely identify the configuration items.

ALC_CMS.2.3C For each TSF relevant configuration item, the configuration list shall indicate the developer of the item.

ALC_CMS.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_DEL.1 Delivery procedures

ALC_DEL.1.1D The developer shall document and provide procedures for delivery of the TOE or parts of it to the consumer.

ALC_DEL.1.2D The developer shall use the delivery procedures.

ALC_DEL.1.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to the consumer.

ALC_DEL.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_FLR.1 Basic flaw remediation

ALC_FLR.1.1D The developer shall document and provide flaw remediation procedures addressed to TOE developers.

ALC_FLR.1.1C The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

ALC_FLR.1.2C The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC_FLR.1.3C The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC_FLR.1.4C The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.

ALC_FLR.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.4 Security Target Evaluation (ASE)

ASE_CCL.1 Conformance claims

ASE_CCL.1.1D The developer shall provide a conformance claim.

ASE_CCL.1.2D The developer shall provide a conformance claim rationale.

ASE_CCL.1.1C The conformance claim shall contain a CC conformance claim that identifies the version of the CC to which the ST and the TOE claim conformance.

ASE_CCL.1.2C The CC conformance claim shall describe the conformance of the ST to CC Part 2 as either CC Part 2 conformant or CC Part 2 extended.

ASE_CCL.1.3C The CC conformance claim shall describe the conformance of the ST to CC Part 3 as either CC Part 3 conformant or CC Part 3 extended.

ASE_CCL.1.4C The CC conformance claim shall be consistent with the extended components definition.

ASE_CCL.1.5C The conformance claim shall identify all PPs and security requirement packages to which the ST claims conformance.

ASE_CCL.1.6C The conformance claim shall describe any conformance of the ST to a package as either package-conformant or package-augmented.

ASE_CCL.1.7C The conformance claim rationale shall demonstrate that the TOE type is consistent with the TOE type in the PPs for which conformance is being claimed.

ASE_CCL.1.8C The conformance claim rationale shall demonstrate that the statement of the security problem definition is consistent with the statement of the security problem definition in the PPs for which conformance is being claimed.

ASE_CCL.1.9C The conformance claim rationale shall demonstrate that the statement of security objectives is consistent with the statement of security objectives in the PPs for which conformance is being claimed.

ASE_CCL.1.10C The conformance claim rationale shall demonstrate that the statement of security requirements is consistent with the statement of security requirements in the PPs for which conformance is being claimed.

ASE_CCL.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_ECD.1 Extended components definition

ASE_ECD.1.1D The developer shall provide a statement of security requirements.

ASE_ECD.1.2D The developer shall provide an extended components definition.

ASE_ECD.1.1C The statement of security requirements shall identify all extended security requirements.

ASE_ECD.1.2C The extended components definition shall define an extended component for each extended security requirement.

ASE_ECD.1.3C The extended components definition shall describe how each extended component is related to the existing CC components, families, and classes.

ASE_ECD.1.4C The extended components definition shall use the existing CC components, families, classes, and methodology as a model for presentation.

ASE_ECD.1.5C The extended components shall consist of measurable and objective elements such that conformance or nonconformance to these elements can be demonstrated.

ASE_ECD.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_ECD.1.2E The evaluator shall confirm that no extended component can be clearly expressed using existing components.

ASE_INT.1 ST introduction

ASE_INT.1.1D The developer shall provide an ST introduction.

ASE_INT.1.1C The ST introduction shall contain an ST reference, a TOE reference, a TOE overview and a TOE description.

ASE_INT.1.2C The ST reference shall uniquely identify the ST.

ASE_INT.1.3C The TOE reference shall uniquely identify the TOE.

ASE_INT.1.4C The TOE overview shall summarise the usage and major security features of the TOE.

ASE_INT.1.5C The TOE overview shall identify the TOE type.

ASE_INT.1.6C The TOE overview shall identify any non-TOE hardware/software/firmware required by the TOE.

ASE_INT.1.7C The TOE description shall describe the physical scope of the TOE.

ASE_INT.1.8C The TOE description shall describe the logical scope of the TOE.

ASE_INT.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_INT.1.2E The evaluator shall confirm that the TOE reference, the TOE overview, and the TOE description are consistent with each other.

ASE_OBJ.2 Security objectives

ASE_OBJ.2.1D The developer shall provide a statement of security objectives.

ASE_OBJ.2.2D The developer shall provide a security objectives rationale.

ASE_OBJ.2.1C The statement of security objectives shall describe the security objectives for the TOE and the security objectives for the operational environment.

ASE_OBJ.2.2C The security objectives rationale shall trace each security objective for the TOE back to threats countered by that security objective and OSPs enforced by that security objective.

ASE_OBJ.2.3C The security objectives rationale shall trace each security objective for the operational environment back to threats countered by that security objective, OSPs enforced by that security objective, and assumptions upheld by that security objective.

ASE_OBJ.2.4C The security objectives rationale shall demonstrate that the security objectives counter all threats.

ASE_OBJ.2.5C The security objectives rationale shall demonstrate that the security objectives enforce all OSPs.

ASE_OBJ.2.6C The security objectives rationale shall demonstrate that the security objectives for the operational environment uphold all assumptions.

ASE_OBJ.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_REQ.2 Derived security requirements

ASE_REQ.2.1D The developer shall provide a statement of security requirements.

ASE_REQ.2.2D The developer shall provide a security requirements rationale.

ASE_REQ.2.1C The statement of security requirements shall describe the SFRs and the SARs.

ASE_REQ.2.2C All subjects, objects, operations, security attributes, external entities and other terms that are used in the SFRs and the SARs shall be defined.

ASE_REQ.2.3C The statement of security requirements shall identify all operations on the security requirements.

ASE_REQ.2.4C All operations shall be performed correctly.

ASE_REQ.2.5C Each dependency of the security requirements shall either be satisfied, or the security requirements rationale shall justify the dependency not being satisfied.

ASE_REQ.2.6C The security requirements rationale shall trace each SFR back to the security objectives for the TOE.

ASE_REQ.2.7C The security requirements rationale shall demonstrate that the SFRs meet all security objectives for the TOE.

ASE_REQ.2.8C The security requirements rationale shall explain why the SARs were chosen.

ASE_REQ.2.9C The statement of security requirements shall be internally consistent.

ASE_REQ.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_SPD.1 Security problem definition

ASE_SPD.1.1D The developer shall provide a security problem definition.

ASE_SPD.1.1C The security problem definition shall describe the threats.

ASE_SPD.1.2C All threats shall be described in terms of a threat agent, an asset, and an adverse action.

ASE_SPD.1.3C The security problem definition shall describe the OSPs.

ASE_SPD.1.4C The security problem definition shall describe the assumptions about the operational environment of the TOE.

ASE_SPD.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_TSS.1 TOE summary specification

ASE_TSS.1.1D The developer shall provide a TOE summary specification.

ASE_TSS.1.1C The TOE summary specification shall describe how the TOE meets each SFR.

ASE_TSS.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_TSS.1.2E The evaluator shall confirm that the TOE summary specification is consistent with the TOE overview and the TOE description.

5.3.5 Tests (ATE)

ATE_COV.1 Evidence of coverage

ATE_COV.1.1D The developer shall provide evidence of the test coverage.

ATE_COV.1.1C The evidence of the test coverage shall show the correspondence between the tests in the test documentation and the TSFIs in the functional specification.

ATE_COV.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_FUN.1 Functional testing

ATE_FUN.1.1D The developer shall test the TSF and document the results.

ATE_FUN.1.2D The developer shall provide test documentation.

ATE_FUN.1.1C The test documentation shall consist of test plans, expected test results and actual test results.

ATE_FUN.1.2C The test plans shall identify the tests to be performed and describe the scenarios for performing each test. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.3C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.4C The actual test results shall be consistent with the expected test results.

ATE_FUN.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2 Independent testing – sample

ATE_IND.2.1D The developer shall provide the TOE for testing.

ATE_IND.2.1C The TOE shall be suitable for testing.

ATE_IND.2.2C The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

ATE_IND.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2.2E The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

ATE_IND.2.3E The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

5.3.6 Vulnerability Assessment (AVA)

AVA_VAN.2 Vulnerability analysis

AVA_VAN.2.1D The developer shall provide the TOE for testing.

AVA_VAN.2.1C The TOE shall be suitable for testing.

AVA_VAN.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VAN.2.2E The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.2.3E The evaluator shall perform an independent vulnerability analysis of the TOE using the guidance documentation, functional specification, TOE design and security architecture description to identify potential vulnerabilities in the TOE.

AVA_VAN.2.4E The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

6 TOE Summary Specification

This section describes the following security functions implemented by the TOE to satisfy the SFRs claimed in Section 5.2:

- Security Audit
- Cryptographic Support
- User Data Protection
- Identification and Authentication
- Security Management
- Protection of the TSF
- Trusted Path/Channels.

6.1 Security Audit

AppGate SDP has two different types of log records: debug logs and audit logs. Debug logs are typically used to examine the workings of the AppGate SDP system itself and audit logs are used to record the actions performed by the system.

All appliances generate debug and audit logs. Both types are held together on the appliance in a 4GB rolling store that is protected from unauthorized access by file permissions—only subjects with root privilege are able to access the log store. The `journald` daemon manages the storage and when the 4GB is fully used it removes the oldest records. Although debug logs and audit logs are held together on each appliance, they are handled differently by the AppGate SDP system.

Audit logs are written to disk locally within each appliance using `logd`. The `logd` daemon uses a database to manage the writing of audit logs to disk and can be configured to write these logs in one of 3 ways: Default; Performance; or Guaranteed. These three options can be selected in **Settings > Global settings**.

- In Default Logging mode, `logd` batches records in 1 second lots. This ensures that at any time the number of unwritten (potentially lost) logs will never exceed 1 second + any write delay in the disk sub-system.
- Performance Logging works exactly the same way as Default, but instead of writing to a physical disk, RAM is used. This increases the writing speed and reduces the IOPS loading on the disk sub-subsystem. The System Administrator needs to ensure the appliance has sufficient RAM (maximum 1GB) before enabling this option.
- If Guaranteed Logging is enabled, then all the daemons in the AppGate SDP appliance will demand that each audit log record is written before allowing whatever action. The daemons will wait for an 'ack' from `logd` before proceeding. The daemons attempt to connect to `logd` multiple times and `logd` attempts to write to disk multiple times. The effect of this is that the writing of each audit log record is essentially guaranteed. The downside of using Guaranteed Logging is that on a slow system daemons may be waiting for up to 5 seconds (the give-up waiting time) before continuing. Clearly, this can have a significant performance impact on the system. For this reason, it is recommended only enabling this feature if there is a specific audit or security requirement mandating it. It is also recommended to use this feature only on where there is some control over the disk system performance (e.g., can specify SSD with high IOPS).

The audit logs are not intended to remain on or be viewed on the appliance, so in the evaluated configuration they are exported to an external audit server using `rsyslog`. The `rsyslog` destination for an appliance is configured in **General Administration → Admin UI → System → Appliances → Configure Appliances → RSYSLOG**, as is export over a TLS-protected channel. Audit record export happens in real time so there should never be a significant amount of unsent audit logs on an appliance at any time. There may be situations when the audit logs cannot be sent for a period of time, such as when the `rsyslog` destination is taken offline (for instance, during an upgrade) or during a network outage. In these situations, the appliances buffer up to 1GB of audit log records, which will be exported as soon as the problem is resolved. In the event the 1GB buffer space is exhausted, newly generated audit records are lost.

AppGate SDP generates audit records of the following auditable events:

- Appliance started (equivalent to startup of the audit function)
- Appliance deactivated (equivalent to shutdown of the audit function)
- Administrator authorization succeeded and failed
- User authentication succeeded and failed
- Local user account locked out due to consecutive authentication failures
- User authorization succeeded and failed
- User blacklisted
- User blacklist entry removed
- Claims token accepted, rejected, expired
- Entitlement token, accepted, rejected, expired
- Remedy action triggered
- Remedy action authentication succeeded and failed
- Token revoked
- Client attempt to access Destination
- Tunnel from Gateway to Client established, closed
- Client session created, reconnected, removed
- System entity (Appliance, Condition, Entitlement, Identity Provider, Local User, Policy, etc.) created, modified, deleted, viewed
- Global settings updated
- Device on-boarded, deleted from system.

Each audit record includes the following information: date and time of the event; type of event; subject identity; and the outcome of the event. Where the auditable event results from the actions of an identified user, AppGate SDP includes the user identity in the audit record. This will be the Distinguished Name, associated with the user, comprising the device identifier, username, and identity provider name. This is the unique identifier which will be included in all audit records no matter how many Controllers or Gateways are being used by the user/device. "Collective ID" is also recorded in each audit record, so when logs have been consolidated from several different Collectives it is possible to easily identify the source.

The Security Audit security function satisfies the following security functional requirements:

- FAU_GEN.1—the TOE generates audit records for security relevant events. Generated records include the date and time of the event, type of event, subject identity, and outcome of the event.
- FAU_GEN.2—the TOE associates each auditable event resulting from actions of identified users with the identity of the user that caused the event.
- FAU_EXP.1—the TOE can be configured to export audit records to an external syslog server over a TLS-protected channel.
- FAU_STG.1—the TOE protects stored audit records from unauthorized modification and deletion.
- FAU_STG.4—the TOE will overwrite the oldest stored audit records if the local storage space is exhausted.

6.2 Cryptographic Support

AppGate SDP incorporates OpenSSL 1.0.2u with OpenSSL FIPS Object Module 2.0.16 to support its TLS implementation, including symmetric encryption and decryption using AES in GCM mode, digital signature generation and verification using RSA, cryptographic hashing using SHA-256 and SHA-384, and keyed-hash message authentication using HMAC-SHA-256 and HMAC-SHA-384. The OpenSSL FIPS Object Module destroys cryptographic keys when they are no longer needed.

AppGate SDP also incorporates the Bouncy Castle Java API v1.65, which provides the cryptographic operations to encrypt token data and sign tokens and generate RSA key pairs for appliance and Client certificates.

Table 5: Cryptographic Capabilities

Functions	Standards	Usage
Key Generation		
RSA key pairs with 4096 bit modulus	FIPS 186-4	Appliance and Client certificates
Deterministic Random Bit Generation		
Counter DRBG: AES-128 and AES-256 modes	ISO/IEC 18031:2011	Generating AES keys
Symmetric Encryption and Decryption		
AES: 128, 256 bits in GCM mode	ISO 18033-3 (AES) ISO 19772 (GCM)	TLS Ciphersuites Single Packet Authorization Claims and Entitlement token encryption
Digital Signature Generation and Verification Services		
RSA: 2048, 3072, 4096 bits	FIPS 186-4	TLS Ciphersuites Generation and verification of certificate and token signatures

Functions	Standards	Usage
Cryptographic Hashing		
SHA-256 SHA-384 SHA-512	ISO/IEC 10118-3:2004	TLS Ciphersuites (SHA-256, SHA-384) Certificate signature generation and verification (SHA-512)
Keyed-Hash Message Authentication Code		
HMAC-SHA-256 HMAC-SHA-384	ISO/IEC 9797-2:2011	TLS Ciphersuites

By default, the Controller operates as the Certificate Authority (CA) in an AppGate SDP deployment. The AppGate SDP system uses a self-signed root CA certificate and certificates generated from this CA certificate to establish trusted communications between appliances (peer interface – port 444) and from Clients to appliances (Client interface – ports 443 and 53 when SPA is used in UDP-TCP mode).

For Admin UI and REST API access (HTTPS on port 8443 by default), the System Administrator can upload an external certificate that the browser can validate, so the administrator will see a trusted connection on their browser.

The Controller generates appliance certificates using 4096-bit RSA with SHA-512 as the signature algorithm.

The Cryptographic Support security function satisfies the following security functional requirements:

- FCS_CKM.1(*)—the TOE uses its implementation of DRBG using AES in CTR mode to generate symmetric AES keys and RSA key pairs.
- FCS_CKM.4—the TOE destroys keys that are no longer required by overwriting with zeroes.
- FCS_COP.1(*)—the TOE performs cryptographic operations as specified in Table 5 above, using the identified algorithms in accordance with the identified standards.

6.3 User Data Protection

AppGate SDP implements the SDP Information Flow Policy to mediate network traffic between Clients (comprising an identified user together with a provisioned client device) and Destinations (comprising network-based resources), based on configured Entitlements and Claims.

Gateways are deployed in front of networked resource infrastructure (Sites). The Controller defines access rights for Clients. A Client establishes a secure TLS tunnel to the Controller, which authenticates the user and issues a Claims token to the Client. The Client then uses the Claims token to request Entitlement tokens from the Controller. The Controller verifies the user claims and issues Entitlement tokens to the Client, based on configured Policies. The Client submits the Entitlement tokens to the Gateways, which control the user's subsequent access to Destinations based on the granted Entitlements.

Claims tokens have a lifetime (default is 1,440 minutes (24 hours)). When a Claims token expires, the Controller requests re-authentication, but the user experience depends on which identity provider the user has authenticated against. If the Client has cached the user credentials, the Claims token renewal

should be transparent to the user. Otherwise (e.g., if the user has authenticated using SAML), the user will be prompted to re-authenticate in order to renew the token.

AppGate SDP uses **Policies** to assign rights to a user or group of users. Policies include **Entitlements**, **Device Security** settings, and any **Admin Roles**. Policy assignment is done using claims-based expressions identifying who the Policy should be assigned to. The Controller manages this process and the resulting configurations are then granted to the user in the form of:

- Site-based Entitlement token(s), comprising the list of Entitlements and **Conditions**
- Device Security settings, which are pushed to the connected Client
- an Admin token (for an administrator), comprising the list of Admin Roles allowed.

The default firewall rule within the Gateway will Block (deny) access. The System Administrator creates and modifies Entitlements to specifically define which traffic is to be Allowed. Each Entitlement defines the rules for controlling access to network resources on a particular Site. The main elements of an Entitlement are the Site, **Actions** (traffic protocols, target hosts, ports), and any Conditions that must be met for those Actions to be allowed by the Gateway.

As an example, the following Entitlement allows the user IP Access to 10.0.0.1 server on port 80 only if the time is between 09.00 and 17.00:

<Site> "Net01"<Actions> "TCP up to 10.0.0.1 on port 80" <Conditions> "Office Hours"

Conditions are included in Entitlements to control conditional access to network resources. Conditions are claims-based expressions that define the criteria that need to be <True> for an Entitlement to be allowed. For example, access may only be allowed if the user meets the Condition of working from an office-based computer. The default Condition for an Entitlement is `Always`, i.e., unrestricted unless a specific Condition has been included. If all Conditions are removed from an Entitlement, the Entitlement is never applied by a Gateway and is effectively disabled.

Typically, Entitlements are used to <ALLOW> user traffic to network resources. However, an Entitlement can <BLOCK> traffic to a service, or be used to trigger an <ALERT> if traffic is sent to a particular host.

Conditions can also be used to create **remedy methods**—alternative criteria that would unblock access if the user's other claims don't satisfy the requirements. For example, if working from home, then providing multi-factor authentication could be an alternative method for gaining access to particular network resources. Remedy methods allow the user to remedy (change) the situation by satisfying an alternative access criterion, e.g., by providing multi-factor authentication, as in the example just cited.

AppGate SDP supports the following types of remedy method:

- Display Message—provides the user with information, such as "Please use the wired network for access"
- Require MFA—the user must satisfy multi-factor authentication in order to access the requested network resources
- Provide Reason—the user needs to type in text before continuing
- Password—the user needs to provide a password in order to continue.

Each type of remedy method has a corresponding User Interaction—a message card that is displayed by the Client to prompt the user for the required data.

If an Entitlement is blocked because a Condition is false and a remedy method has been configured, the Gateway will communicate the User Interaction to the Client. The Client manages the user's response as appropriate. The Controller will update the claims token, and the Gateway will re-evaluate the Conditions in the Entitlement to produce new firewall rules.

The Gateway uses the Entitlements and Conditions to configure firewall rules for each user. If the user's claims match the access criteria in the Condition, then the Entitlement will become live. If the claims don't meet the criteria then the Condition will not pass. In this situation, any remedy method that has been set up in the Condition will be triggered. The Gateway regularly re-evaluates the Conditions and reconfigures firewall rules in response to changes in the user's claims. A Condition can be used to force additional re-evaluations to ensure the Gateway responds quickly to relevant changes.

The administrator can configure when and how often a remedy method is triggered for a particular user:

- Using re-evaluation time intervals in the Condition configuration to eliminate unnecessary re-authentication triggers
- Setting a specific time within the Condition
- Using specific claim names to control access for a particular user or network resource
- Using general claim names to avoid re-triggering a remedy method unnecessarily.

Entitlement tokens are valid until one of the following events occurs:

- the user session ends (i.e., the user logs out)
- the Entitlement token expires (the default lifetime for an Entitlement token is 1440 minutes)
- the Entitlement token is renewed.

Changes to Policies and Entitlements are propagated automatically by the Controller when the user's Entitlement token expires and is renewed, or when the user signs-out and signs-in again. The System Administrator can also propagate changes in real time by renewing tokens for a specified active device or by renewing all tokens in the system.

Actions in Entitlements may overlap in terms of subnet, protocol, ports and types, so that a packet matches more than one action. The actions may be in the same Entitlement, or in different Entitlements. The overlap may be a mistake, or it may be deliberate, e.g., adding exceptions to a range by mixing <ALLOW> and <BLOCK> actions.

In any case, if a packet matches more than one action, only the action with the highest precedence is used. The other actions are not logged, do not trigger Remedy Actions, do not raise Alerts, and do not influence whether the packet is allowed or blocked.

The rules of action precedence are as follows:

1. The action with a smaller subnet has higher precedence—for instance, "block 172.23.23.0/24 port 80" has precedence over "allow 172.23.0.0/16 port 80", so that a packet to 172.23.23.1:80

is blocked, despite it matching both actions. This is because a /24 subnet is smaller than a /16 subnet.

2. If the subnets are equal, the action with a smaller port (or type) range has higher precedence—for instance, "block 172.23.0.0/16 port 80" has precedence over "allow 172.23.0.0/16 port 1-65535", so that a packet to 172.23.0.1:80 is blocked, despite it matching both actions. This is because the range "80" is smaller than the range "1-65535".
3. If the subnets are equal, and the port (or type) ranges are equally small, the action with a higher starting port (or type) has higher precedence—for instance, "block 172.23.0.0/16 port 11-100" has precedence over "allow 172.23.0.0/16 port 1-90", so that a packet to 172.23.0.1:80 is blocked, despite it matching both actions. This is because the range "11-100" starts higher than "1-90".
4. If the subnets and port (or type) ranges are equal, an allowing action wins over a blocking action, where "allowing" means an action marked "allow" in an Entitlement that is currently available, and "blocking" means either an action marked "block" or "alert", or an action marked "allow" in an Entitlement for which the Conditions are currently not fulfilled. For instance, "allow 172.23.0.0/16 port 80" has precedence over "block 172.23.0.0/16 port 80". Additionally, if two Entitlements have the same "allow 172.23.0.0/16 port 80" action, where one Entitlement is already available, but the other would require a Remedy Action, then the already available Entitlement will be used and a Remedy Action will not be requested.
5. If two Actions in an Entitlement are identical, meaning that subnets, port (or type) ranges and "allowing/blocking" are equal, the precedence is unpredictable, meaning that either action may be used in the Entitlement. This definition of identical includes the case where a specific block action and an allow action with a condition that has not been met are present. While this precedence has no practical implications for the outcome of the Entitlement, the unpredictability has implications for Audit Logs. The Audit Log will only note the applied Action. For example, suppose an Entitlement `foo` has `allow tcp-up to intranet port 80`, and `bar` has the same Action, `allow tcp-up to intranet port 80`. The user connects to `intranet:80 and the access will be allowed. However the log will note either `foo` or `bar` but not both.

AppGate SDP appliances protect themselves using Single Packet Authorization (SPA). This effectively makes the appliances invisible on the network. In order to establish a connection to an appliance (Controller or Gateway), a client must first send an SPA packet to the appliance. The SPA packet includes a time stamp to protect against replay attacks and is encoded using a pre-shared key known only to the client and the appliance. Once the appliance verifies the SPA, a TLS tunnel is established between the client and appliance. The System Administrator configures the SPA settings for the Client port (443), which can be one of the following modes (TCP mode is enabled by default):

- Disable—SPA is not require and any Client can connect to the appliance on port 443 and establish a TCP session
- TCP—valid Clients are required to present a TLS ClientHello packet comprising a specially crafted custom extension to allow TLS connections to be established

- UDP-TCP—valid Clients are required to present a specially crafted UDP packet (SPA-DTLS packet on 443 or SPA-DNS on 53). The appliance's firewall is then updated to allow 443 access for UDP (for DTLS) or TCP (for TLS). For TLS tunnels the SPA-TCP process is then performed.

All communications between Clients and Gateways are protected using mutually authenticated TLS 1.2, using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289 as the cipher suite.

The User Data Protection function satisfies the following security functional requirements:

- FDP_IFC.1/FDP_IFF.1—the TOE enforces the SDP Information Flow Policy to mediate network traffic between users and network-based resources, based on configured Entitlements and Claims
- FDP_ITT.1—the TOE protects user data transmitted between physically separated parts of the TOE from disclosure and modification using TLS
- FMT_MSA.1—the TOE restricts the ability to modify Entitlements to the System Administrator
- FMT_MSA.3—the TOE provides restrictive default values for security attributes used to enforce the SDP Information Flow Policy and allows the System Administrator to specify alternative initial values to override the restrictive default values.
- FMT_SAE.1—the TOE will request re-authentication when a Claims token expires and will update Entitlements when an Entitlements token expires.

6.4 Identification and Authentication

After (or at) installation, a Client profile must be used to configure the Client before it is able to connect to a Controller.

A Client profile includes the profile name plus the minimum set of information required for a user to be able to sign-in to an AppGate SDP Controller. The resulting elements are:

- a URL for the Controller(s)
- an assigned identity provider against which the user will authenticate – Users cannot choose an Identity Provider or accept an unknown certificate.
- the profile name (which appears in the Client)
- SPA details including the key required to establish a TCP/UDP connection to the Controller
- a means to verify the CA fingerprint to ensure the Controller is genuine

The first two items are what define a unique profile. If a unique profile is updated (changed name, SPA or CA), then it will REPLACE any existing profile. If another unique profile is created (different URL or IdP) then it will CO-EXIST with any existing unique profiles.

Profiles can be exported to streamline the process of on-boarding Clients. This is done from the admin UI in **Settings > Client Connections**. Installation of the desktop Clients includes the profile link, so users have everything they need all in one place.

In AppGate SDP, the profile link provides the Clients with all the information needed to access the Controllers. The tokens from the Controllers include settings which can be used to configure (and enforce the use of) routes and firewall rules on the connecting device. The tokens also provide all the

information for each Gateway to setup and manage the user's firewall rules. Clients manage their own connectivity and are free to access any of the available Controllers or Gateways (on allowed Sites).

AppGate SDP requires all users to be successfully identified and authenticated before allowing them to perform any other activities. AppGate SDP provides a local password-based authentication mechanism and can be configured to support remote authentication of client users and administrators using LDAP, RADIUS, and SAML identity providers. AppGate SDP additionally supports a Multi Factor Authentication mechanism that can be triggered in various circumstances.

Local password-based authentication

“Local users” are user identities that have been added to the local (Controller) system database. The System Administrator (username **admin**) local user account is pre-configured in the local database to enable full system administration and cannot be deleted.

AppGate SDP is able to detect when an administrator-configurable number (in the range from 1 to 99) of unsuccessful authentication attempts has occurred related to authentication of local users. When the defined number of unsuccessful authentication attempts has been met, AppGate SDP locks the user account for a period of 1 minute.

LDAP/RADIUS identity providers

AppGate SDP supports the use of LDAP and RADIUS to authenticate users connecting through the Client, and also to authenticate administrators logging into the Admin UI.

LDAP certificate identity providers

AppGate SDP supports LDAP certificate-based authentication through standard enterprise identity providers (IdP) such as Active Directory. Authentication is supported for just Windows Clients and the Linux headless Client. Authentication for administrators using the admin UI is not supported.

AppGate SDP supports only RSA-based certificates that specify clientAuthentication key usage, have a valid subjectAlternativeName field, and include the private key. Certificates must be located in **Certificates-Current User > Personal > Certificates** on the Windows Client. The Controller will map attributes based on the subjectAlternativeName to populate user claims and issue Claims and Entitlement tokens.

SAML

AppGate SDP supports single sign-on authentication using SAML 2.0 identity providers such as ADFS, OKTA, OneLogin and Ping. SAML can be used to authenticate a user connecting through the Client, and also to authenticate administrators logging into the Admin UI. When user launches the Client and is provided with a profile link that includes a SAML Identity Provider, the Client initiates the SAML authentication request and uses the default browser on the user's device to handle the SAML flow and session cookies. The SAML response is picked up by the Client and sent to the Controller, which checks the Authentication, Attribute and Authorization Decision assertions. If authentication has been successful, the Controller uses the SAML attribute assertions to populate AppGate SDP user claims so they can be used to issue Claims and Entitlement tokens as appropriate.

For administrator authentication at the Admin UI, when a SAML identity provider is selected from the drop-down list, the username and password fields will vanish. Clicking **<Sign in>** will redirect the user to the identity provider website to enter their credentials. The SAML response is then handled by the Controller similarly to the Client user.

AppGate SDP has the following requirements for SAML authentication:

- The SAML response issued by the SAML identity provider must be a base64 encoded, XML assertion
- The Status code must be "urn:oasis:names:tc:SAML:2.0:status:Success"
- At least one assertion must be present in the response. It can be an encrypted assertion if there is a decryption key configured in the identity provider.
- Only the first assertion will be validated and used. The rest will be discarded.
- Assertion.Subject.NameID must exist. This value is mapped by the Controller to the claim-name <username>
- The attribute NameID will also be mapped to the claim-name <samlNameId>
- The assertion must have an 'AuthnStatement'
- The assertion must have an 'AttributeStatement'
- The assertion must have audience defined under Assertion.Conditions.AudienceRestrictions[0].Audiences[0].AudienceURI and the value must match the value of <Audience> entered in the Identity Provider configuration.
- The assertion must have an issuer tag (URL) and its value must match the value of <Issuer> entered in the Identity Provider configuration.
- The assertion lifetime conditions 'NotBefore' and 'NotOnOrAfter' must be valid compared to the system time.
- The assertion signature must be validated by the public certificate entered in the Identity Provider Configuration.

Multi Factor Authentication

AppGate SDP can be configured to require additional authentication using Multi Factor Authentication (MFA) providers in the following circumstances:

- As part of a remedy method in a Condition in a Client's Entitlement
- As part of System Administrator authentication
- During Device on-boarding.

AppGate SDP can be configured to use its built in Default Time-Based OTP Provider or an external RADIUS Provider.

The built-in OTP provider works with OATH time-based Authenticator apps such as Google Authenticator. The app can be downloaded and auto-initialized the first time the user interaction is triggered for that user. The user is guided through the download and setup process by the Client.

The external RADIUS support includes pre-emptive, RADIUS based and challenge-response modes.

The Identification and Authentication function satisfies the following security functional requirements:

- FIA_AFL.1—the TOE is able to detect when an administrator-configurable positive integer of unsuccessful authentication attempts occur related to user authentication. When the defined number of unsuccessful authentication attempts has been met, the TOE locks the user account for 1 minute.
- FIA_UAU.2—the TOE requires each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.
- FIA_UAU.5—the TOE supports multiple authentication mechanisms: local password; LDAP; RADIUS; certificate-based LDAP; SAML; and Multi Factor Authentication.
- FIA_UID.2—the TOE requires each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

6.5 Security Management

AppGate SDP is managed using the Admin UI. To access the Admin UI, administrators need to be able to resolve and access the IP address or hostname of the Controller, have an account in the Local Database or a configured Identity provider, and be assigned the appropriate privileges via an Admin Role. The Admin Roles for each administrator will be listed in their Admin token once they have logged into the Admin UI, and will be valid until the session ends (the administrator logs out), the Admin token expires, or the Admin token is renewed. Token expiry and token renewal will terminate an active session and the administrator will have to log in again.

The System Administrator account (username **admin**) is pre-configured in AppGate SDP with full system administration privileges.

AppGate SDP also provides the ability for the System Administrator to create additional administrative user accounts and to define Admin Roles. Admin Roles enable administrators to manage some or all aspects of the system from the Admin UI. However, only the System Administrator account is included within the scope of the evaluation.

The Admin UI provides the System Administrator with the following management capabilities:

- Manage Policies—the System Administrator uses the **Operations > Policies** section of the Admin UI to create, view, enable, disable, edit, and delete Policies
- Manage Entitlements—the System Administrator uses the **Operations > Entitlements** section of the Admin UI to create, view, enable, disable, edit and delete Entitlements
- Manage Conditions—the System Administrator uses the **Operations > Conditions** section of the Admin UI to create, view, edit, and delete Conditions
- Manage Sites—the System Administrator uses the **System > Sites** section of the Admin UI to create, view, edit, and delete Sites
- Manage Identity Providers—the System Administrator uses the **System > Identity Providers** section of the Admin UI to create, view, edit, and delete identity providers
- Manage Users—the System Administrator uses the **System > Identity Providers** section of the Admin UI to create, view, edit, and delete local user accounts
- Manage local user lockout policy—the System Administrator uses the **System > Identity Providers** section of the Admin UI to configure the number of consecutive failed authentication attempts that will cause a local user account to be locked

- Change local user passwords—the System Administrator uses the **System > Identity Providers** section of the Admin UI to change the password on an existing local user account
- Configure token lifetimes—the System Administrator uses the **Settings > Global Settings** section of the Admin UI to configure lifetimes for Claims tokens, Entitlement tokens, and Administration tokens
- Renew tokens—the System Administrator uses the **Users and Devices > Active Devices** section of the Admin UI to renew all tokens for a single device or for all devices in the deployment. This will revoke the relevant tokens and generate new ones and provides a mechanism for the System Administrator to propagate changes in Policies, Entitlements and Admin Roles to Clients in real time rather than waiting for tokens to expire and refresh automatically)
- Blacklist user—the System Administrator uses the **Users and Devices > Active Devices** section of the Admin UI to blacklist a user, which immediately revokes all of the user's tokens, terminates any currently active sessions, and prevents the user from subsequently logging in
- Remove device—the System Administrator uses the **Users and Devices > On-boarded Devices** section of the Admin UI to remove a device from the system
- Configure Appliances—the System Administrator uses the **System > Appliances** section of the Admin UI to configure new appliances to be added to the Collective, including specifying the appliance's roles, configuring network interfaces, and selecting the Site the appliance will belong to if it is in a Gateway role.

The Security Management function satisfies the following security functional requirements:

- FMT_REV.1—the TOE restricts the ability to revoke a Client's Entitlements to the System Administrator. The TOE revokes the Client's Entitlements as soon as the user is blacklisted.
- FMT_SAE.1—the TOE restricts the ability to configure expiration times for Claims, Entitlement and Admin tokens to the System Administrator. The TOE will terminate the administrator's session when an Admin token expires.
- FMT_SMF.1—the TOE provides the capabilities necessary to manage the security of the TOE.
- FMT_SMR.1—the TOE defines a built-in System Administrator role with full system administration privileges.

6.6 Protection of the TSF

All communications between distributed AppGate SDP components is over TLS v1.2, as follows:

- Communications between a Client and Controller, with initial authentication request using HTTPS, can use either of the following cipher suites, as defined in RFC 5289:
  - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- Communications between Controllers and Gateways is mutually-authenticated TLS using
  - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,
  - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384, as defined in RFC 5289.

The AppGate SDP appliances are virtual appliance images installed on virtualization platforms that include a hardware-based real-time clock. AppGate SDP's embedded operating system manages the clock and exposes administrator clock-related functions. In order to operate correctly, AppGate SDP appliances need to be able to establish connections to external NTP servers. Synchronized/coordinated

time is critical for AppGate SDP operation because of the use of time-stamped tokens. AppGate SDP currently uses NTP v4.2.8 (RFC 5905).

AppGate SDP provides the System Administrator with the ability to determine the current running version of AppGate SDP. The **Dashboard > Appliances** section of the Admin UI displays details of all appliances deployed in the Collective, including the version of software running on each appliance.

In order to update the AppGate SDP software, the System Administrator downloads the update image from the AppGate support website. The update image has a GPG digital signature, configured to use RSA with a 2048 bit key, which is verified prior to initiating the update. The update can be launched from the System Administrator's local system using an upgrade script, or by uploading the image to each appliance in the Collective. The advantage of using the upgrade script is that it performs an orchestrated upgrade of all appliances in the Collective, including verification of the update image prior to launching.

The Protection of the TSF security function satisfies the following security functional requirements:

- FPT_ITT.1—the TOE uses TLS to protect TSF data from disclosure and modification when it is transmitted between distributed parts of the TOE.
- FPT_STM.1—the TOE is able to provide reliable time stamps, based on its own internal clock (for hardware-based components) or a time source in its operational environment. The TOE components can also be configured to synchronize with NTP servers.
- FPT_TUD.1—the TOE enables the administrator to query the currently executing version of the TOE software and initiate software/firmware updates for the TOE. The integrity of TOE updates is verified using a digital signature.

6.7 TOE Access

AppGate SDP provides users with the ability to terminate their sessions once they have logged in to the TOE.

Prior to logging in all user are presented with a message warning against unauthorized use of the TOE.

The TOE Access security function satisfies the following security functional requirements:

- FTA_SSL.4—the TOE allows user-initiated termination of the user's own interactive session.
- FTA_TAB.1—the TOE displays an advisory warning message regarding unauthorized use of the TOE.

6.8 Trusted Path/Channels

AppGate SDP provides a trusted path for administrators to communicate with the Controller. The trusted path is implemented using HTTPS (i.e., HTTP over TLS) for access to the Admin UI. Administrators initiate the trusted path by establishing an HTTPS connection using a supported web browser. The trusted path implementation supports TLS v1.2 with the following cipher suites, as defined in RFC 5289:

- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256.

The trusted path is used for initial authentication and all subsequent administrative actions. The use of HTTPS ensures all communication over the trusted path is protected from disclosure and modification.

AppGate SDP supports communications via trusted channels with trusted IT products in its operational environment for the following functions:

- Export of audit records to an external audit server—the administrator can configure AppGate SDP to export audit records to an external audit server that has `rsyslog` (8.16.0 or higher) and the `rsyslog-gnutls` plugin installed.
- Remote authentication of Client and Admin UI users via LDAP or RADIUS.

The trusted channels to these external IT entities are initiated by the AppGate SDP appliance, using TLS v1.2. In this circumstance, AppGate SDP supports the set of cipher suites enabled by default on the Java platform.

The Trusted Path/Channels security function satisfies the following security functional requirements:

- FTP_ITC.1—the TOE provides a trusted channel to transmit securely audit records to an external syslog server and user authentication requests to external authentication servers.
- FTP_TRP.1—the TOE provides a trusted path for administrators to communicate with the TOE, using HTTPS to access the Admin UI.

7 Rationale

This section provides the rationale for completeness and consistency of the ST. The rationale addresses the following areas:

- Security Objectives
- Security Functional Requirements
- Security Assurance Requirements
- Requirement Dependencies
- TOE Summary Specification.

7.1 Security Objectives Rationale

This section shows that all secure usage assumptions and threats are completely covered by security objectives for the TOE or operational environment. In addition, each objective counters or addresses at least one assumption or threat.

Table 6: Security Problem Definition to Security Objective Correspondence

	T.BRUTE_FORCE	T.INAPPROPRIATE_USE	T.NETWORK_ACCESS	T.NETWORK_COMPROMISE	T.NO_ACCOUNTABILITY	T.UNATTENDED_SESSION	T. UNAUTHORIZED_ACCESS	T.UNAUTHORIZED_ACTIVITY	T.UPDATE_COMPROMISE	A. MANAGE	A.PROTECT	A.HOSTNAME_RESOLUTION
O.AUDIT					X							
O.AUDIT_STORAGE					X							
O.CRYPTOGRAPHY				X								
O.I_AND_A							X					
O.INFORMATION_FLOW			X									
O.LOGON_BANNER		X										
O.PROTECTED_COMMS				X								
O.SECURITY_MANAGEMENT								X				
O.SESSION_TERMINATION						X						
O.THROTTLE	X											
O.TRUSTED_UPDATE									X			
OE.PERSONNEL										X		

	T.BRUTE_FORCE	T.INAPPROPRIATE_USE	T.NETWORK_ACCESS	T.NETWORK_COMPROMISE	T.NO_ACCOUNTABILITY	T.UNATTENDED_SESSION	T. UNAUTHORIZED_ACCESS	T.UNAUTHORIZED_ACTIVITY	T.UPDATE_COMPROMISE	A. MANAGE	A.PROTECT	A.HOSTNAME_RESOLUTION
OE.PHYSICAL											**X**	
OE.HOSTNAME_RESOLUTION												**X**

T.BRUTE_FORCE

An unauthorized user may gain access to the TOE through repeated password-guessing attempts.

This threat is countered by the following security objective:

- O.THROTTLE—addresses this threat by limiting the rate at which an unauthorized user is able to make attempts to guess the password of an authorized user.

T.INAPPROPRIATE_USE

Authorized users perform inappropriate actions on the TOE due to ignorance of their responsibilities or operational policies and procedures.

This threat is countered by the following security objective:

O.LOGON_BANNER—addresses this threat by displaying an advisory warning message to potential users pertaining to appropriate use of the TOE.

T.NETWORK_ACCESS

An attacker attempts to gain access to networked resources and compromise the data or services provided by those resources.

This threat is countered by the following security objective:

- O.INFORMATION_FLOW—addresses this threat by mediating the transmission of network packets between users and network resources based on user identity and authorization to access network resources.

T.NETWORK_COMPROMISE

An unauthorized user may monitor the enterprise network in an attempt to obtain sensitive data, such as passwords, or to modify transmitted data.

This threat is countered by the following security objectives:

- O.PROTECTED_COMMS—addresses this threat by ensuring all communications between distributed parts of the TOE, and between the TOE and external entities, are protected using cryptographic protocols.
- O.CRYPTOGRAPHY—supports O.PROTECTED_COMMS by ensuring the TOE implements the cryptographic algorithms necessary to support the cryptographic protocols that satisfy O.PROTECTED_COMMS.

T.NO_ACCOUNTAILITY

Authorized users of the TOE perform adverse actions on the TOE, or attempt to perform unauthorized actions, which go undetected.

This threat is countered by the following security objectives:

- O.AUDIT—addresses this threat by ensuring the TOE is able to generate audit records of security relevant events.
- O.AUDIT_STORAGE—supports O.AUDIT in addressing the threat by ensuring the TOE protects stored audit records from unauthorized modification and deletion.

T.UNATTENDED_SESSION

An unauthorized user gains access to the TOE via an unattended authorized user session.

This threat is countered by the following security objectives:

- O.SESSION_TERMINATION—addresses this thread by providing users with a mechanism to terminate their interactive sessions with the TOE.

T.UNAUTHORIZED_ACCESS

An unauthorized user may gain access to the TOE security functions and data.

This threat is countered by the following security objective:

- O.I_AND_A—addresses this threat by ensuring all users of the TOE are identified and authenticated prior to gaining further access to the TOE and its services.

T.UPDATE_COMPROMISE

An attacker may attempt to provide a compromised update of the software or firmware that undermines the security functionality of the device.

This threat is countered by the following security objective:

- O.TRUSTED_UPDATE—addresses this threat by ensuring the TOE is able to verify the integrity and authenticity of software or firmware updates prior to applying those updates

A.MANAGE

There will be one or more competent individuals assigned to manage the TOE and the security of the information it contains.

This assumption is satisfied by the following security objective:

- OE.PERSONNEL—this objective satisfies the assumption by ensuring those assigned as authorized administrators are properly trained in operating the TOE.

A.PROTECT

The TOE components critical to security policy enforcement will be protected from unauthorized physical modification.

This assumption is satisfied by the following security objective:

- OE.PHYSICAL—this objective satisfies the assumption by ensuring the TOE is protected from physical attack.

A.HOSTNAME_RESOLUTION

The operational environment must ensure that security measures are in place to protect DNS hostname resolution.

This assumption is satisfied by the following security objective:

- OE.HOSTNAME_RESOLUTION The operational environment is responsible for DNS hostname resolution. As such, strategies in the operational environment must be implemented to provide security protections.

7.2 Security Functional Requirements Rationale

All security functional requirements identified in this ST are fully addressed in this section and each is mapped to the objective it is intended to satisfy. Table 7 summarizes the correspondence of functional requirements to TOE security objectives.

Table 7: Objectives to Requirement Correspondence

	O.AUDIT	O.AUDIT_STORAGE	O.CRYPTOGRAPHY	O.I_AND_A	O.INFORMATION_FLOW	O.LOGON_BANNER	O.PROTECTED_COMMS	O.SECURITY_MANAGEMENT	O.SESSION_TERMINATION	O.THROTTLE	O.TRUSTED_UPDATES
FAU_GEN.1	X										
FAU_GEN.2	X										
FAU_EXP.1		X									
FAU_STG.1		X									
FAU_STG.4		X									
FCS_CKM.1(*)			X				X				
FCS_CKM.4			X								
FCS_COP.1(*)			X				X				X
FDP_IFC.1					X						
FDP_IFF.1					X						
FDP_ITT.1							X				
FIA_AFL.1										X	
FIA_UAU.2				X							
FIA_UAU.5				X							
FIA_UID.2				X							
FMT_MSA.1					X						
FMT_MSA.3					X						
FMT_REV.1								X			
FMT_SAE.1					X			X			
FMT_SMF.1								X			
FMT_SMR.1								X			
FPT_ITT.1							X				

	O.AUDIT	O.AUDIT_STORAGE	O.CRYPTOGRAPHY	O.I_AND_A	O.INFORMATION_FLOW	O.LOGON_BANNER	O.PROTECTED_COMMS	O.SECURITY_MANAGEMENT	O.SESSION_TERMINATION	O.THROTTLE	O.TRUSTED_UPDATES
FPT_STM.1	X				X						
FPT_TUD.1											X
FTA_SSL.4									X		
FTA_TAB.1						X					
FTP_ITC.1							X				
FTP_TRP.1							X				

O.AUDIT

The TOE shall be able to generate audit records of security-relevant events, identifying users causing the events as applicable.

The following security functional requirements contribute to satisfying this security objective:

- FAU_GEN.1—the ST includes FAU_GEN.1 to specify the capability to generate audit records of security-relevant events, and to specify the specific events to be audited and the content of generated audit records of those events.
- FAU_GEN.2—the ST includes FAU_GEN.2 to specify the capability to associate auditable events with the identity of the user that caused the event.
- FPT_STM.1—the ST includes FPT_STM.1 to support FAU_GEN.1 by ensuring the TOE provides a reliable time stamp for inclusion in generated audit records.

O.AUDIT_STORAGE

The TOE shall protect stored audit records from unauthorized modification and deletion and shall preserve the most recent stored audit records in the event audit storage is exhausted. The TOE additionally shall provide a means to securely export audit records to an external audit server for long term storage and review.

The following security functional requirements contribute to satisfying this security objective:

- FAU_STG.1—the ST includes FAU_STG.1 to specify the capability to protect stored audit records from unauthorized modification and deletion.
- FAU_STG.4—the ST includes FAU_STG.4 to specify the capability to overwrite the oldest stored audit records with new audit records in the event the space available for audit record storage is exhausted.
- FAU_EXP.1—the ST includes FAU_EXP.1 to specify the capability to export audit records to an external IT entity over a trusted channel.

O.CRYPTOGRAPHY

The TOE shall perform cryptographic operations to support protocols used to protect data in transit.

The following security functional requirements contribute to satisfying this security objective:

- FCS_COP.1(*)—the ST includes iterations of FCS_COP.1 to specify the cryptographic algorithms for symmetric and asymmetric encryption and decryption, cryptographic hashing, and keyed-hash message authentication necessary to support the protocols used to protect data in transit.
- FCS_CKM.1(*)—the ST includes iterations of FCS_CKM.1 to specify the capability to generate symmetric and asymmetric cryptographic keys necessary to support symmetric and asymmetric encryption and decryption.
- FCS_CKM.4—the ST includes FCS_CKM.4 to support the iterations of FCS_CKM.1 by ensuring cryptographic keys are securely destroyed when no longer needed by the TOE.

O.I_AND_A

The TOE shall require all users of the TOE to be identified and authenticated before gaining access to TOE services.

The following security functional requirements contribute to satisfying this security objective:

- FIA_UID.2, FIA_UAU.2—the ST includes FIA_UID.2 and FIA_UAU.2 to specify that users must be successfully identified and authenticated by the TOE before being able to perform any other TSF-mediated actions.
- FIA_UAU.5—the ST supports FIA_UAU.2 by including FIA_UAU.5 to specify multiple authentication mechanisms that are supported by the TOE.

O.INFORMATION_FLOW

The TOE shall mediate the transmission of network packets between users and network resources based on user identity, and authorization to access network resources based on IP address, protocol and port number.

The following security functional requirements contribute to satisfying this security objective:

- FDP_IFC.1, FDP_IFF.1—the ST includes FDP_IFC.1 and FDP_IFF.1 to specify the information flow control policy enforced by the TOE to control the communication of network traffic between users and network resources.
- FMT_MSA.1, FMT_MSA.3— the ST includes FMT_MSA.1 and FMT_MSA.3 to specify restrictions on the management of security attributes used to control the flow of network traffic.
- FMT_SAE.1—the ST supports FDP_IFC.1 and FDP_IFF.1 by including FMT_SAE.1 to specify the behavior of the TSF when certain security attributes used to control the flow of network traffic expire.
- FPT_STM.1—the ST includes FPT_STM.1 to support FMT_SAE.1 by ensuring the TOE provides a reliable time stamp for determining when specified security attributes have expired.

O.LOGON_BANNER

The TOE shall be able to display an advisory warning message to potential users pertaining to appropriate use of the TOE.

The following security functional requirements contribute to satisfying this security objective:

- FTA_TAB.1—The ST includes FTA_TAB.1 to specify the capability to display an advisory warning message regarding unauthorized use of the TOE.

O.PROTECTED_COMMS

The TOE shall protect communications between its distributed components, and between itself and external entities, from disclosure and modification.

The following security functional requirements contribute to satisfying this security objective:

- FDP_ITT.1, FPT_ITT.1—the ST includes FDP_ITT.1 and FPT_ITT.1 to specify that communications of user data and TSF data between distributed parts of the TOE will be protected from disclosure and modification.
- FTP_ITC.1, FTP_TRP.1—the ST includes FTP_ITC.1 and FTP_TRP.1 to specify that communications between the TOE and external entities will be protected from disclosure and modification.

O.SECURITY_MANAGENENT

The TOE shall restrict the ability to perform security management functions on the TOE to authorized administrators.

The following security functional requirements contribute to satisfying this security objective:

- FMT_SMF.1, FMT_SMR.1, FMT_REV.1, FMT_SAE.1—the ST includes these requirements to specify the security management functions to be provided by the TOE (FMT_SMF.1), to specify security management roles (FMT_SMR.1), and to specify TOE behavior when security attributes are revoked (FMT_REV.1) or expire (FMT_SAE.1).

O.SESSION_TERMINATION

The TOE shall provide mechanisms to terminate a user session at the request of the user.

The following security functional requirements contribute to satisfying this security objective:

- FTA_SSL.4—The ST includes this requirement to specify the capability for users to terminate their own interactive session.

O.THROTTLE

The TOE shall limit the rate at which consecutive unsuccessful authentication attempts can be performed.

The following security functional requirement contributes to satisfying this security objective:

- FIA_AFL.1—the ST includes FIA_AFL.1 to specify the capability to limit the rate at which consecutive failed authentication attempts (which may indicate a password-guessing attack) can be made.

O.TRUSTED_UPDATE

The TOE shall apply cryptographic means to verify the integrity and authenticity of software or firmware updates prior to applying those updates.

The following security functional requirement contributes to satisfying this security objective:

- FPT_TUD.1—the ST includes FPT_TUD.1 to specify the capability for the TSF to verify the integrity and authenticity of software or firmware updates using a digital signature mechanism, and for the System Administrator to apply such updates to the TOE.

7.3 Security Assurance Requirements Rationale

EAL 2 was selected as the assurance level because the TOE is a commercial product whose users require a low to moderate level of independently assured security. The TOE is intended for use in an environment with good physical access security where it is assumed that attackers will have Basic attack potential. The target assurance level of EAL 2 is appropriate for such an environment. Augmentation was chosen to provide the added assurance that is gained by defining flaw remediation procedures. Therefore, the target assurance level of EAL 2 augmented with ALC_FLR.1 is appropriate for such an environment.

7.4 Requirement Dependency Rationale

The following table identifies the SFRs claimed in the ST, their dependencies as defined in CC Part 2 or the extended components definition (Section 5.1 of this ST), and how the dependency is satisfied in the ST. It can be seen that all dependencies have been satisfied, either by inclusion in the ST of the appropriate dependent SFRs, or by functionality provided by the operational environment.

Table 8: Requirement Dependencies

Requirement	Dependencies	How Satisfied
FAU_GEN.1	FPT_STM.1	FPT_STM.1
FAU_GEN.2	FAU_GEN.1	FAU_GEN.1
	FIA_UID.1	FIA_UID.2 (hierarchical to FIA_UID.1)
FAU_EXP.1	FAU_GEN.1, FTP_ITC.1	FAU_GEN.1, FTP_ITC.1
FAU_STG.1	FAU_GEN.1	FAU_GEN.1
FAU_STG.4	FAU_STG.1	FAU_STG.1
FCS_CKM.1(*)	[FCS_CKM.2 or FCS_COP.1], FCS_CKM.4	FCS_COP.1, FCS_CKM.4
FCS_CKM.4	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1
FCS_COP.1(*)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1], FCS_CKM.4	FCS_CKM.1, FCS_CKM.4
FDP_IFC.1	FDP_IFF.1	FDP_IFF.1
FDP_IFF.1	FDP_IFC.1, FMT_MSA.3	FDP_IFC.1, FMT_MSA.3
FDP_ITT.1	[FDP_ACC.1 or FDP_IFC.1]	FDP_IFC.1
FIA_AFL.1	FIA_UAU.1	FIA_UAU.2 (hierarchical to FIA_UAU.1)
FIA_UAU.2	FIA_UID.1	FIA_UID.2 (hierarchical to FIA_UID.1)
FIA_UAU.5	None	None
FIA_UID.2	None	None
FMT_MSA.1	FMT_SMR.1, FMT_SMF.1	FMT_SMR.1, FMT_SMF.1
	FDP_ACC.1 or FDP_IFC.1	FDP_IFC.1
FMT_MSA.3	FMT_SMR.1, FMT_MSA.1	FMT_SMR.1, FMT_MSA.1
FMT_REV.1	FMT_SMR.1	FMT_SMR.1
FMT_SAE.1	FMT_SMR.1, FPT_STM.1	FMT_SMR.1, FPT_STM.1
FMT_SMF.1	None	None
FMT_SMR.1	FIA_UID.1	FIA_UID.2 (hierarchical to FIA_UID.1)
FPT_ITT.1	None	None
FPT_STM.1	None	None
FPT_TUD.1	FCS_COP.1, FMT_SMR.1	FCS_COP.1, FMT_SMR.1
FTA_SSL.4	None	None

Requirement	Dependencies	How Satisfied
FTA_TAB.1	None	None
FTP_ITC.1	None	None
FTP_TRP.1	None	None

7.5 TOE Summary Specification Rationale

Section 6, the TOE Summary Specification, describes how the security functions of the TOE meet the claimed SFRs. The following table provides a mapping of the SFRs to the security function descriptions to support the TOE Summary Specification.

Table 9: Security Functions vs. Requirements Mapping

	Security audit	Cryptographic support	User data protection	Identification and authentication	Security management	Protection of the TSF	TOE Access	Trusted path/channels
FAU_GEN.1	X							
FAU_GEN.2	X							
FAU_EXP.1	X							
FAU_STG.1	X							
FAU_STG.4	X							
FCS_CKM.1(*)		X						
FCS_CKM.4		X						
FCS_COP.1(*)		X						
FDP_IFC.1			X					
FDP_IFF.1			X					
FDP_ITT.1			X					
FIA_AFL.1				X				
FIA_UAU.2				X				
FIA_UAU.5				X				
FIA_UID.2				X				
FMT_MSA.1			X					

	Security audit	Cryptographic support	User data protection	Identification and authentication	Security management	Protection of the TSF	TOE Access	Trusted path/channels
FMT_MSA.3			X					
FMT_REV.1					X			
FMT_SAE.1			X		X			
FMT_SMF.1					X			
FMT_SMR.2					X			
FPT_ITT.1						X		
FPT_STM.1						X		
FPT_TUD.1						X		
FTA_SSL.4							X	
FTA_TAB.1							X	
FTP_ITC.1								X
FTP_TRP.1								X