US Government Protection Profile

Authorization Server

For

Basic Robustness Environments

Information

Assurance

Directorate

July 25, 2007

Version 1.1

Foreword

This Protection Profile "*US Government Protection Profile Authorization Server for Basic Robustness Environments*" (PP) was updated using Version 3.1 of the Common Criteria (CC).

Editor's note: The purpose of this update was to bring the PP up to the new CC 3.1 standard without changing the authors' original meaning or purpose of the documented requirements. The original PP was developed using version 2.x of the CC. The CC version 2.3 was the final version 2 update that included all international interpretations. CC version 3.1 used the final CC version 2.3 Security Functional Requirements (SFR)s as the new set of SFRs for version 3.1. Some minor changes were made to the SFRs in version 3.1, including moving a few SFRs to Security Assurance Requirements (SAR)s. There may be other minor differences between some SFRs in the version 2.3 PP and the new version 3.1 SFRs. These minor differences were not modified to ensure the author's original intent was preserved.

The version 3.1 SARs were rewritten by the common criteria international community. The NIAP/CCEVS staff developed an assurance equivalence mapping between the version 2.3 and 3.1 SARs. The assurance equivalent version 3.1 SARs replaced the version 2.3 SARs in the PP.

Any issue that may arise when claiming compliance with this PP can be resolved using the observation report (OR) and observation decision (OD) process.

Further information, including the status and updates of this protection profile can be found on the CCEVS website: http://www.niap-ccevs.org/cc-scheme/pp/. Comments on this document should be directed to ppcomments@missi.ncsc.mil. The email should include the title of the document, the page, the section number, the paragraph number, and the detailed comment and recommendation.

Version 1.1 July 25, 2007

Protection Profile Title:

Authorization Server Protection Profile for Basic Robustness Environments.

Criteria Version:

This Protection Profile (PP) was developed using Version 3.1 of the Common Criteria (CC) [1a] and applying the NIAP interpretations that have been approved by CCEVS Management as of May 1, 2004.

Constraints:

Targets of Evaluation (TOEs) developed to satisfy this Protection Profile shall conform to CC Part 2 and CC Part 3 and applicable NIAP approved interpretations.

Table of Contents

List of Figures

List of Tables

1 INTRODUCTION

This section contains document management and overview information necessary to allow a Protection Profile (PP) to be registered through a Protection Profile Registry. The Identification provides the labeling and descriptive information necessary to identify, catalogue, register, and cross-reference a PP. The Overview summarizes the profile in narrative form and provides sufficient information for a potential user to determine whether the PP is of interest. The Overview can also be used as a stand-alone abstract for PP catalogues and registers. The Conventions section provides an explanation of the Common Criteria (CC) notation. The Terms section gives a basic definition of terms, which are specific to this PP. The Related Profiles section identifies profiles directly related to this profile and may be of interest to those interested in this profile. Finally, the Protection Profile Organization section describes how this document is organized.

Protection Profile Identification

Title: US Government Protection Profile Authorization Server for Basic Robustness Environments (PPASBRE)

Sponsor: National Security Agency (NSA)

CC Version: Common Criteria for Information Technology Security Evaluation Part 2: Security functional requirements, Version 3.1, September 2006, Part 2 extended. Common Criteria for Information Technology Security Evaluation Part 3: Security assurance requirements, Version 3.1, September 2006,. Part 3 conformant Evaluation Assurance Level 2 (EAL 2) augmented with ALC_FLR.2,. STs that claim conformance to this PP shall meet a minimum standard of demonstrable-PP conformance as defined in section D3 of part 1.

Registration: <to be provided upon registration>

Protection Profile Version: Version 1.1, dated July 25, 2007

Keywords: authorization, access control, Enterprise Access Management (EAM), Privilege Management Infrastructure (PMI), Authorization Service

Protection Profile Overview

This PP specifies a set of security functional and assurance requirements for Authorization Server products. The Authorization Server is a family of software products that supports access control of IT resources (e.g., web servers, databases, application servers, individual web pages, and specific data files/objects). Access control, or authorization, is defined as determining whether a *principal* shall be granted permission to perform an *operation* on a *resource*. The term *principal* indicates an authenticated identity, and might be a user at a web browser, web service, or other application. The *operation* would most often be read access (e.g. viewing a web page or querying a web service interface), but might also include other operations such as creation,

modification and deletion. The *resource* could be static content (e.g., web pages, files and images) or dynamic (e.g., web applications and services).

Authorization Server functionality provides a capability to map a principal's identity to a set of privilege attributes. It also provides a mechanism to assign access requirements for IT resources. When acting as an Authorization Server, the TOE executes pre-defined rules or policies which compare a principal's privilege attributes to the requested IT resources access requirements to make an access control decision. The majority of products with PPASBRE compliant STs will support Authorization Server functionality, but it is not mandatory (it is possible to comply with PPASBRE with only Attribute Authority functionality).

Additional functionality may or may not be present in an Authorization Server product and will be specified by the refinement of the security functional requirements (SFRs) of section 0 by the ST author – relevant SFRs and application notes in the relevant SFRs will detail where refinements should be applied. The additional functionality includes:

- Authorization Enforcement – If the TOE enforces the access control decision to grant or deny access to a resource.

- Authentication Server – If the TOE performs authentication of the principals who are attempting to access protected resources.

- Attribute Authority – If the TOE provides an interface for external applications and/or users to obtain principals' privilege attributes.

The deployment of Authorization Servers can also be characterized as a deployment of "Privilege Management Infrastructure" (PMI). The PMI can be defined as the systems, processes and software required to operate an "Authorization Service."

PPASBRE-conformant products provide the ability to protect themselves and their associated data from unauthorized access or modification while ensuring accountability for authorized actions.

The PPASBRE is a "software only" PP dependent on the IT environment (hardware, operating system, and other software products) to meet some of the security functional requirements for a Basic Robustness environment (as defined by the NSA Information Assurance Directorate (IAD) document "Protection Profile (PP) Consistency Guidance for Basic Robustness"). This protection profile provides a level of protection that is appropriate for IT environments that have main Authorization Server components on a private protected network (e.g., behind firewalls) and administered by highly trusted users. The TOE and IT Environment do not fully address threats posed by malicious administrative or system development personnel. PPASBRE-conformant products are suitable for use in both commercial and government environments.

The PPASBRE was constructed to provide a target and metric for the development of Authorization Server software. This PP identifies security functions and assurances representative of the lowest common set of requirements that should be addressed by a useful

Authorization Server. Targets of Evaluation (TOEs) compliant with this PP must meet the assurance requirements of Evaluation Assurance Level (EAL) 2 augmented.

This PP defines the following items:

- Assumptions about security aspects of the environment in which the TOE will be used;
- Threats that are to be addressed by the TOE;
- Organizational security policies pertaining to the TOE;
- Security objectives of the TOE and its environment;
- Functional and assurance requirements to meet those security objectives; and
- Rationale demonstrating how the requirements meet the security objectives, and how the security objectives address the threats.

STs that claim conformance to this PP shall meet a minimum standard of demonstrable-PP conformance as defined in section D3 of part 1.

Conventions

The notation, formatting, and conventions used in this PP are largely consistent with those used in version 3.1 of the CC. Selected presentation choices are discussed here to aid the PP user.

The CC allows several operations to be performed on functional requirements; *refinement*, *selection*, *assignment*, and *iteration* are defined in section C.4 of Part 1 of the CC. Each of these operations is used in this PP.

The **refinement** operation is used to add detail to a requirement, and thus further restricts a requirement. Refinement of security requirements is denoted by **bold text**.

The **selection** operation is used to select one or more options provided by the CC in stating a requirement. Selections are denoted by *italicized text*.

The **assignment** operation is used to assign a specific value to an unspecified parameter, such as the length of a password. Assignment is indicated by showing the value in square brackets, [Assignment value].

The **iteration** operation is used when a component is repeated with varying operations. Iteration is denoted by showing the iteration number in parenthesis following the component identifier, (iteration number).

The **Security Target (ST) author** operation is used to denote points in which the final determination of attributes is left to the ST writer. ST writer operations are indicated by the words “determined by the ST Author.”

The CC paradigm also allows PP and ST authors to create their own requirements. Such requirements are termed 'extended requirements' and are permitted if the CC does not offer suitable requirements to meet the authors' needs. Extended requirements must be identified and are required to use the CC class/family/component model in articulating the requirements. In this PP, extended requirements will be indicated with the "EXT" appended to the family name.

Application Notes are provided to help the developer, to clarify the intent of a requirement, identify implementation choices, or define "pass-fail" criteria for a requirement. For those components where Application Notes are appropriate, the Application Notes will follow the requirement component.

Related Protection Profiles

There are no PPs that directly relate to the Authorization Server software. However, the following PPs provide security requirements to components that make up the IT Environment in which the Authorization Server software is deployed:

- Web: Web Server Protection Profile, Web Browser Protection Profile, Version: 1.0, dated 26 December 2006

  If the TOE supports remote administration via web browser, then the guidance documents shall instruct administrators to use a web browser that has been evaluated to be compliant with the Web Server Protection Profile (if any such web browsers exist at the time of the TOE evaluation).

- Operating Systems: Controlled Access (Basic Robustness/C2) (CAPP) Version. 1.d, dated 8 October 1988

  The TOE shall run on an operating system that has been evaluated to be compliant with the Controlled Access Protection Profile.

Protection Profile Organization

Section 1, PP Introduction, provides the document management and overview information necessary to identify the PP along with references to other related PPs.

Section 2, TOE Description, defines the TOE and establishes the context of the TOE by referencing generalized security requirements.

Section 3, TOE Security Environment (TSE), describes the expected environment in which the TOE is to be used. This section defines the set of threats that are relevant to the secure operation of the TOE, organizational security policies with which the TOE must comply, and secure usage assumptions for the operation of the TOE.

Section 4, Security Objectives, defines the set of security objectives to be satisfied by the TOE and by the TOE operating environment.

Section 5, IT Security Requirements, defines the security functional and assurance requirements derived from the CC, Part 2 and Part 3, respectively that must be satisfied by the TOE, the TOE IT environment, and the Non-IT environment.

Section 6, Rationale, provides rationale to demonstrate that the security objectives satisfy the threats and policies. This section also explains how the set of requirements are complete relative to the security objectives. This section includes a dependency analysis, and rational for the use of extended requirements.

Section 7, References, provides background material for further investigation by users of the PP.

Section 8, Terminology, provides a listing of definitions of terms.

Section 9, Acronyms, provides a listing of acronyms used throughout the document.

2 TOE DESCRIPTION

This PP specifies the minimum security requirements for a TOE composed of several “software only” components, which together, make up an Authorization Server system. The purpose of an Authorization Server is to provide an organization with a web access management solution that helps to enable secure access to web-based resources. These commercial security products enhance website security management by providing a platform for centrally managing access to all web resources and applications. In a large organization, this is cost saving over building proprietary user directories and access control systems in the individual applications. The authorization policy management feature of these products enables central or distributed management of user access privileges. The products also provide for the creation of business or policy rules, often called rulesets, which can incorporate both static (such as a role) or dynamic attributes (such as a principal’s checking account balance) to define the access control requirements to protect web-based resources (e.g.: Universal Resource Locators (URLs), files, and objects).

Authorization Server products often also provide an enforcement functionality in the form of either an “agent” which provides the access control decision enforcement point for application servers or by sitting as a proxy in front of the application server.

In addition to web and application server access management, Authorization Server software products may provide an API to enable applications to make their own access control decisions by obtaining a principal’s privilege attributes. In this mode of operation, the Authorization Server software functions as an “Attribute Authority”.

An Authorization Server requires an authenticated identity as an input to the access control decision. In the core configuration the Authorization Server obtains the authenticated identity as an input, but some products will perform the authentication themselves, in which case the server functions also as an “Authentication Server" providing a Single Sign-on (SSO) capability that allows principals to navigate across web-based resources, both within a single site and across multiple sites, while authenticating only once.

The following subsections define the components (software modules and interfaces) that make up an Authorization Server, and provide a number of operational scenarios that demonstrate the usage of an Authorization Server. Finally, there is a section describing the security features of the TOE.

TOE Definition

The following subsections define the possible components that could make up the TOE. Not all functionality will necessarily be supported by every Authorization Server product.

Table 1 summarizes the TOE components that are described below, and indicates which are mandatory and which are optional.

Table 1 – TOE Components

COMPONENT	REQUIREMENT
Administrative User Interface	Mandatory
Privilege Attribute Data Store	Mandatory
Access Policy Data Store	Mandatory
Authorization Server Policy Decision Engine	Required if Access Control Decision API, Authorization Enforcement Engine, or Authorization Enforcement Agent is present.
Access Control Decision API	At least one of these four components must be present.
Authorization Enforcement Engine	
Authorization Enforcement Agent	
Attribute Authority	
Authentication Server	Optional

2.1.1 Administrative User Interface

The administrative interface capability allows administrators to securely log on and gain access to the TOE's management tools. Administrators may gain access to this component either via a web based interface or a client/server interface, depending on the product's design. If the web interface is used, the administrator's browser should be required to meet the security requirements outlined in the "Web Browser Protection Profile." If a client program is used, the client software is part of the TOE.

2.1.2 Privilege Attribute Data Store

The Privilege Attribute Data Store (PADS) contains data about the principal that make up the authorization domain. This data always includes the privilege attributes that are used by the policy decision engine to make the access control decision. Additionally, if the authentication server functionality is included, the PADS data may include additional information required to authenticate the user, for example password information.

This component also provides the tools to create and modify privilege attributes or entitlements, including creating and managing groups as well as changing values for existing attributes.

2.1.3 Access Policy Data Store

The Policy Data Store contains the data that defines the access control policy. Each policy defines who can access each resource, the conditions under which access will be allowed, and the privilege attribute information needed for a successful authorization.

This software component provides the tools to manage the policy information as well as the storage thereof.

2.1.4 Authorization Server Policy Decision Engine

This software component provides a mapping between the required access criteria for a web based resource and privilege attributes. It performs the required computation to make an access control decision. This component, which would reside in a protected enclave, would require secure interfaces to the agent and to the data stores to obtain the information needed to make the policy decision.

2.1.5 Access Control Decision API

Authorization Server software products generally provide an API that allows authorized applications to obtain access control decisions from the Authorization Server's policy engine. In the cases in which the Authorization Server does not perform Authorization Enforcement, this interface is required for applications to determine whether to grant or deny access to the requested resources.

This API accepts an authenticated principal, the requested resource, and the requested operation as input. The API would then access the privilege attribute and policy data stores as necessary to make the decision, and the Policy Engine would then make the decision and the API would return a "Grant or Deny" response to the requesting software application.

This scenario is detailed in Section 2.1.9 below.

2.1.6 Authorization Enforcement Engine

Some Authorization Servers actually control the resources and enforce the access control decisions. The enforcement engine can be implemented in several ways. Some of these mechanisms are described in the following sections.

2.1.6.1 Authorization Enforcement Agent

This software component, generally provided by the authorization server vendor, is installed on the on the web application server. These agents generally conform to the web servers' native architecture. For example, there is a *module* for Apache®; a *filter* for Microsoft™ Internet Information Server® (IIS); an *extension* for iPlanet®, and so on. These will be referred to simply as Agents throughout this document. NOTE: the web or application server software itself is generally not part of the TOE and neither is part of the evaluation. Essentially, these Agents replace or augment the web server's native security mechanisms. The Agent runs in the same process as the web server itself and is invoked whenever the web server needs to determine

access rights for a particular Uniform Resource Identifier (URI). The Web Server Agent forwards access requests and the principal identity information to the Authorization Server using the Access Control Decision API (section 2.1.5). The Policy Engine in the Authorization Server makes the access control decision and passes the answers back to the Agent. The Agent then enforces the decision by granting or denying the user access to the resource.

This scenario is detailed in Section 2.1.10.1 below.

2.1.6.2 Authorization Enforcement Proxy

This software component resides in the network topology between the principals and the resources being requested. In this case, the request from the principal (e.g. the HTTP request) will be examined to identify the resources and the operations being requested. The proxy will authenticate the principal, and interface to the Authorization Server (using the Access Control Decision API) to obtain a grant or deny decision. Based on that decision, the proxy will then either permit the request by transferring to the HTTP to the appropriate location, or will deny access to the user (displaying a static access denied page, or redirecting to a registration site, etc).

This scenario is detailed in Section 2.1.10.2 below.

2.1.7 Attribute Authority

Authorization Server software products may provide an API that enables designated custom applications or databases to obtain user entitlements from the PADS. This API allows the Authorization Server software to function as an "Attribute Authority" to support various IT resources that need user attributes to make their own access control decisions. When the API receives the request for a user attribute, it must first validate the identity of the requesting software entity and ensure it is authorized to use the API. The API would have an interface to the PADS from which it would obtain the user entitlement. The API would then return the attribute values requested to the application or database making the request.

This scenario is detailed in Section 2.1.11 below.

2.1.8 Authentication Server

Some Authorization Server products include Identification and Authentication (I&A) of principals. When I&A functionality is included, the Authorization Server product generally supports multiple mechanisms. The most common are user name/passwords and X.509 PKI certificates, but others include Windows Domain Authentication, Microsoft Passport, Liberty Alliance, RSA's SecureID, s/key, etc. The component that performs these services for the TOE is called the Authentication Server.

The Authentication Server may rely solely on information in the Privilege Attribute Data Store, for example in the case of password based authentication, when the password or a hash thereof may be validated by comparing the value stored in the PADS.

Usage Scenarios

The following sections outline three basic scenarios in which Authorization Server products operate: (1) access control decision; (2) access control decision and enforcement, and (3) attribute authority.

All of the following scenarios assume the existence of an operational system with principals, resources, and policies defined.

2.1.9 Web Server Access Control Scenario

Figure 1 outlines the scenario for an Authorization Server to provide access control decision services to a web server.

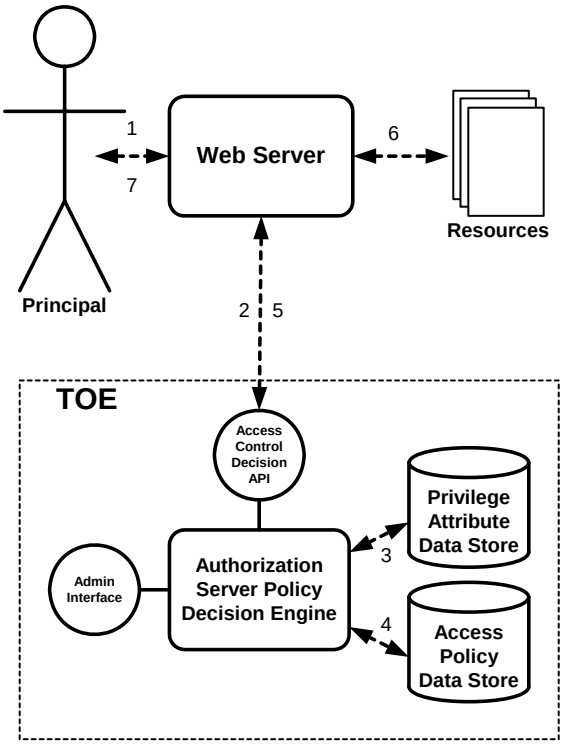

Figure 1 – Web Server Access Control Scenario

The following describes the information between an authorized human user requesting access to a protected web resource and the Authorization Server components depicted above. These steps assume an administrator has previously established access requirements in the Policy Data Store and that the user's entitlements are held in the User Attribute store.

- Step 1: A principal clicks on a link in her web browser, and a HTTP request is sent to the web server. The web server authenticates the principal by whatever means it is configured to use.

- Step 2: The web server invokes the Access Control decision API to query whether access should be granted.
- Step 3: The Policy Engine looks up the principal's privilege attributes from the PADS.
- Step 4: The Policy Engine then obtains the access requirements for the requested resource from the Policy Data Store.
- Step 5: The Policy Engine evaluates the principal's privilege attributes against the access requirements and calculates the access control decision. The decision to grant or deny access is passed over back to the web server.
- Step 6: If access was granted, the web server will obtain the requested resource. If access is denied, the web server may prepare an "access forbidden" page.
- Step 7: The requested resource or the forbidden page is delivered back to the principal by the web server.

2.1.10 Authorization Enforcement

In this section we describe two scenarios: Authorization Enforcement Agent and Proxy. An Authorization Server product compliant with this PP may use other approaches as long as the product meets the security functional and assurance requirements described in Section 5 of this PP.

2.1.10.1 Authorization Enforcement Agent Scenario

Figure 2 illustrates the capability of an Authorization Enforcement Agent.

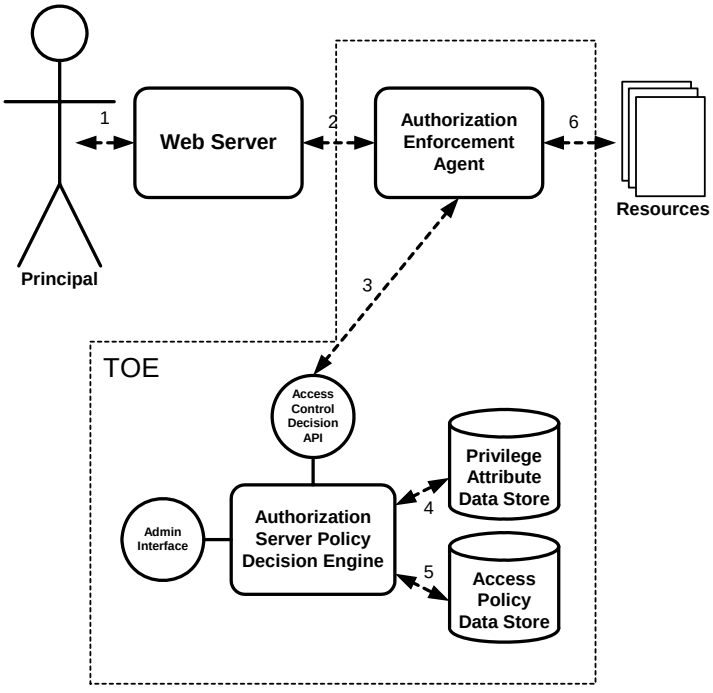

Figure 2 – Authorization Enforcement Agent Scenario

The following steps describe the Authorization Enforcement Agent scenario.

- Step 1: A principal clicks on a link in her web browser, and a HTTP request is sent to the web server.
- Step 2: The web server authenticates the principal and passes the authenticated identity and request information to the Authorization Enforcement Agent.
- Step 3: The Authorization Enforcement Agent queries the Access Control Decision API.
- Step 4: The Policy Engine looks up the principal's privilege attributes from the PADS.
- Step 5: The Policy Engine looks up the access requirements for the requested resource from the Policy Data Store. The Policy Engine calculates the access control decision and passes the result back to the Authorization Enforcement Agent.
- Step 6: If the access decision was to grant access, the Authorization Enforcement Agent obtains the requested resource and passes it back to the web server for delivery to the principal.

2.1.10.2 Authorization Enforcement Proxy Scenario

Figure 3 illustrates the Authorization Enforcement Proxy scenario.

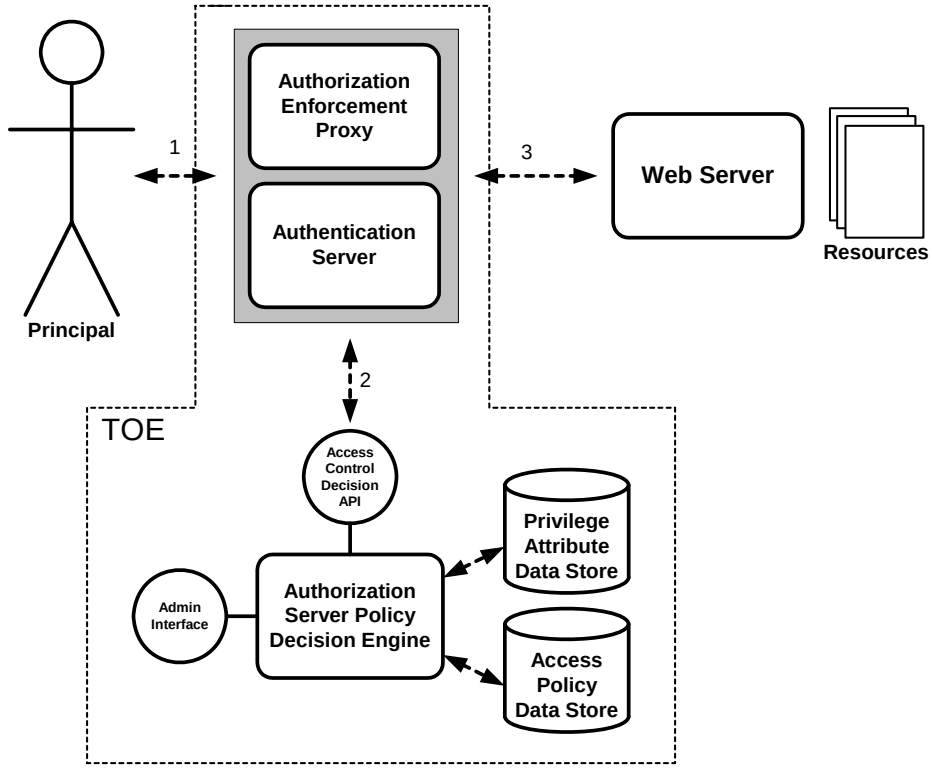

Figure 3 - Authorization Enforcement Proxy Scenario

The following steps describe the Authorization Enforcement Proxy scenario.

- Step 1: A principal clicks on a link in her web browser, and a HTTP request is intercepted by the Authorization Enforcement Proxy. The Authentication Server component performs the steps required to authenticate the principal, which might include communicating with the Privilege Attribute Data Store.

- Step 2: The Authorization Enforcement Proxy passes the authenticated identity and request information to the Authorization Server Policy Decision Engine to determine whether to grant access to the resource.

- Step 3: If the decision was to grant access, the HTTP request will be forwarded to the web server and the response forwarded back to the principal. Otherwise the Authorization Server will present an access denied message to the principal.

2.1.11 Attribute Authority Scenario

Figure 4 illustrates the Attribute Authority scenario.

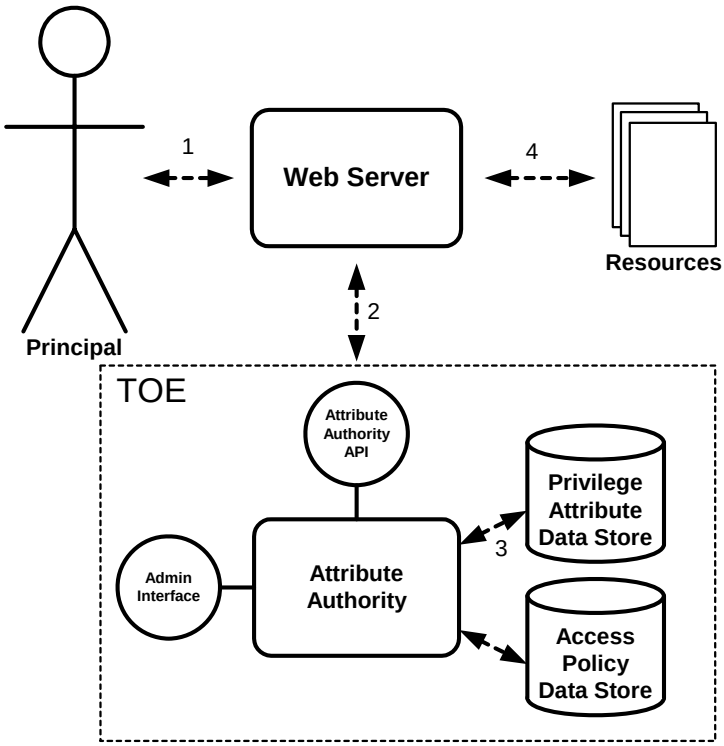

Figure 4 - Attribute Authority Scenario

The following illustrates the capability of an Authorization Server to act as an Attribute Authority, providing privilege attributes to relying applications.

- Step 1: A principal clicks on a link in her web browser, and a HTTP request is sent to the web server.
- Step 2: The web server authenticates the principal, and then queries the Attribute Authority for the privilege attributes required to make an access control decision. The web server may have to authenticate to the Attribute Authority in order to make this request.
- Step 3: If the web server is authorized to access the requested data, the Attribute Authority obtains the requested information from the Privilege Attribute Data Store and returns it to the web server.
- Step 4: The web server can use the principal's attributes to make access control decisions.

TOE Security Features

The TOE security features and functional requirements can be categorized as follows: Identification and Authentication, Administration, Access Control, Encryption, and Audit.

2.1.12 Identification and Authentication

The TOE may perform Identification and Authentication (I&A) for multiple classes of users. There may be different I&A requirements for the different classes of users. At a minimum, administrators must undergo I&A in order to access TOE data, manage privilege attributes, and set up access control policies. Additionally, if the Authorization Server acts as an Attribute Authority, I&A will be performed on the systems or processes that request privilege attributes from the server, to ensure that the requesting party is authorized to obtain the requested data. Finally, the server might act as an Authentication Server on behalf of a web server, in which case the principals requesting access to protected resources will be identified and authenticated by the TOE. The I&A requirements on the principals may vary among classes of users and/or may depend on the sensitivity of the data being requested.

2.1.13 Administration

"Administrators" refers to the roles assigned to the individuals responsible for the installation, configuration, and maintenance of the TOE. The TOE requires two separate administrative roles: Audit Administrator and Security Administrator. The Audit Administrator is responsible for the regular review and management of the TOE's audit data. The Security Administrator is responsible for all other administrative tasks (e.g., managing the access control policies and privilege attributes). The Security Administrator is also responsible for the configuration and maintenance of cryptographic elements related to the establishment of secure connections to and from the TOE. It is important to note that while this PP requires the two administrative roles outlined above, it provides the ST author the option of including additional administrative roles as well. For example, most authorization server products allow for subordinate Security Administrators that can be given limited authority to manage access to specific web resources. This allows for a distributed administration of web resources by local webmasters.

2.1.14 Access Control

Unlike some PPs, where all the access control functionality is designed to provide requirements to protect just the TOE data, this PP specifies the access control functional requirements for the TOE to provide authorization and access control services over protected web based resources that are not actually part of the TOE (unless the TOE is acting solely as an Attribute Authority). This PP includes the introduction of an "Authorization Server Access Control policy." This is not a single "standard" policy, like Discretionary Access Control (DAC), but rather dynamic policy that is based on the Security Administrator's defined rules or operations. In this concept there are "principals" who request access to "resources," where all the operations between principal and resource are covered by the Authorization Server Access Control policy. The "principals" are generally the users of web browsers, or web services. The "resources" are the designated web objects (web server, files, or applications) that the Authorization Server is protecting. The access control is based on the Security Attributes of both the principal and the

resource. These attributes can be identity and group membership(s) associated with a principal, the time of day attribute associated with the operation; and access control attributes associated with a resource**.**

2.1.15 Cryptography

Section 5.1.8 defines the minimum set of cryptographic security functional requirements for the IT Environment that will be made available for the use of the TOE. The cryptographic module must be validated by CMVP for FIPS PUB 140-2 Security Level 1 or higher. The ST author is provided several implementation selections for key generation and may distribute keys manually, electronically, or both. Cryptographic functions can be used to secure communication among the distributed components of the TOE and/or between the TOE and other IT products.

2.1.16 Audit

Section 5.1.1 describes the TOE's generation of auditable events. Since the TOE will be running on top of an operating system that is compliant with the CAPP, the storage, protection, and analysis of the audit records will be a function of the IT environment. The IT environment and the TOE together will cover the alarming aspects as a result of an audit analysis and the overall audit management. Table 9 in the FAU_GEN.1.2 requirement lists the minimum set of auditable events that must be available to the Audit Administrator for configuration on the TOE. Each auditable event must generate an audit record. Table 9 also provides a minimum list of attributes that must be included in each audit record. The ST author may include additional auditable events and audit record attributes. If the ST author includes any additional security functional requirements not specified by this PP, at a minimum they shall include audit events for the basic level of audit, in addition they must consider any security relevant audit events associated with those requirements beyond the basic level and include them in the TOE's list of auditable events and records.

3 TOE SECURITY ENVIRONMENT

In trying to specify the environments in which TOEs with various levels of robustness are appropriate, it is useful to first discuss the two defining factors that characterize that environment: ***value of the resources*** and ***authorization of the entities*** to those resources.

In general terms, the environment for a TOE can be characterized by the authorization (or lack of authorization) the least trustworthy entity has with respect to the highest value of TOE resources (i.e., the TOE itself and all of the data processed by the TOE).

Note that there are an infinite number of combinations of entity authorization and value of resources; this conceptually "makes sense" because there are an infinite number of potential environments, depending on how the resources are valued by the organization, and the variety of authorizations the organization defines for the associated entities. In Section 3.3, these two environmental factors will be related to the robustness required for selection of an appropriate TOE.

Value of Resources

Value of the resources associated with the TOE includes the data being processed or used by the TOE, as well as the TOE itself (for example, a real-time control processor). "Value" is assigned by the organization. For example, in the Department of Defense (DoD) low-value data might be equivalent to data marked "For Official Use Only (FOUO)," while high-value data may be those classified Top Secret. In a commercial enterprise, low-value data might be the internal organizational structure as captured in the corporate on-line phone book, while high-value data might be corporate research results for the next generation product. Note that when considering the value of the data one must also consider the value of data or resources that are accessible through exploitation of the TOE. For example, a firewall may have "low value" data itself, but it might protect an enclave with high value data. If the firewall was being depended upon to protect the high value data, then it must be treated as a high-value-data TOE.

Authorization of Entities

Authorization that entities (users, administrators, other IT systems) have with respect to the TOE (and thus the resources of that TOE, including the TOE itself) is an abstract concept reflecting a combination of the trustworthiness of an entity and the access and privileges granted to that entity with respect to the resources of the TOE. For instance, entities that have total authorization to all data on the TOE are at one end of this spectrum; these entities may have privileges that allow them to read, write, and modify anything on the TOE, including all TSF data. Entities at the other end of the spectrum are those that are authorized to few or no TOE resources. For example, in the case of a router non-administrative entities may have their packets routed by the TOE, but that is the extent of their authorization to the TOE's resources. In the case of an Operating System (OS), an entity may not be allowed to log on to the TOE at all (that is, they are not valid users listed in the OS's user database)**.**

It is important to note that authorization ***does not*** refer to the ***access*** that the entities actually have to the TOE or its data. For example, suppose the owner of the system determines that no one other than employees were authorized to certain data on a TOE, yet they connect the TOE to the Internet. There are millions of entities that are not ***authorized*** to the data (because they are not employees), but they actually have connectivity to the TOE through the Internet and thus can attempt to access the TOE and its associated resources.

Entities are characterized according to the value of resources to which they are authorized; the extent of their authorization is implicitly a measure of how trustworthy the entity is with respect to compromise of the data (that is, compromise of any of the applicable security policies; e.g., confidentiality, integrity, availability). In other words, in this model the greater the extent of an entity's authorization, the more trustworthy (with respect to applicable policies) that entity is.

Selection of Appropriate Robustness level

Robustness is a characteristic of a TOE defining how well it can protect itself and its resources; a more robust TOE is better able to protect itself. This section relates the defining factors of IT environments, authorization, and value of resources to the selection of appropriate robustness levels.

When assessing any environment with respect to Information Assurance (IA) the critical point to consider is the likelihood of an attempted security policy compromise, which was characterized in the previous section in terms of entity authorization and resource value. As previously mentioned, robustness is a characteristic of a TOE that reflects the extent to which a TOE can protect itself and its resources. It follows that as the likelihood of an attempted resource compromise increases, the robustness of an appropriate TOE should also increase.

It is critical to note that several combinations of the environmental factors will result in environments in which the likelihood of an attempted security policy compromise is similar. Consider the following two cases:

- The first case is a TOE that processes only low-value data. Although the organization has stated that only its employees are authorized to log on to the system and access the data, the system is connected to the Internet to allow authorized employees to access the system from home. In this case, the least trusted entities would be unauthorized entities (e.g., non-employees) exposed to the TOE because of the Internet connectivity. However, since only low-value data are being processed, the likelihood that unauthorized entities would find it worth their while to attempt to compromise the data on the system is low and selection of a basic robustness TOE would be appropriate.

- The second case is a TOE that processes high-value (e.g., classified) information. The organization requires that the TOE be stand-alone, and that every user with physical and logical access to the TOE undergo an investigation so that they are authorized to the highest value data on the TOE. Because of the extensive checks done during this investigation, the organization is assured that only highly trusted users are authorized to

use the TOE. In this case, even though high value information is being processed, it is unlikely that a compromise of that data will be attempted because of the authorization and trustworthiness of the users and once again selection of a basic robustness TOE would be appropriate.

The preceding examples demonstrate that it is possible for radically different combinations of entity authorization/resource values to result in a similar likelihood of an attempted compromise. As mentioned earlier, the robustness of a system is an indication of the protection being provided to counter compromise attempts. Therefore, a basic robustness system should be sufficient to counter compromise attempts where the likelihood of an attempted compromise is low. Figure 1 depicts the "universe" of environments characterized by the two factors discussed in the previous section: on one axis is the authorization defined for the least trustworthy entity, and on the other axis is the highest value of resources associated with the TOE.

As depicted in this figure, the robustness of the TOEs required in each environment steadily increases as one goes from the upper left of the chart to the lower right; this corresponds to the need to counter increasingly likely attack attempts by the least trustworthy entities in the environment. Note that the shading of the chart is intended to reflects the notion that different environments engender similar levels of "likelihood of attempted compromise," signified by a similar color. Further, the delineations between such environments are not stark, but rather are finely grained and gradual.

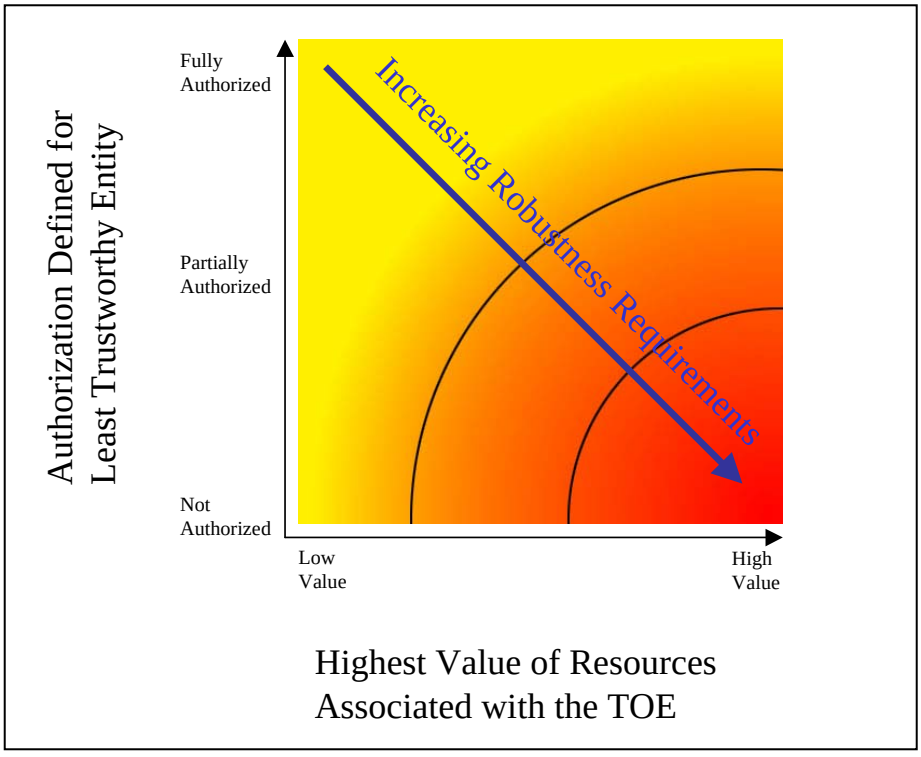

Figure 5 – Environmental Factors for Consideration

While it would be possible to create many different "levels of robustness" at small intervals along the “Increasing Robustness Requirements” line to counter the increasing likelihood of attempted compromise due to those attacks, it would not be practical nor particularly useful. Instead, in order to implement the robustness strategy where there are only three robustness levels: Basic, Medium, and High, the graph is divided into three sections, with each section corresponding to set of environments where the likelihood of attempted compromise is roughly similar. This is graphically depicted in Figure 6 - Sectionalized Environments Figure 4.

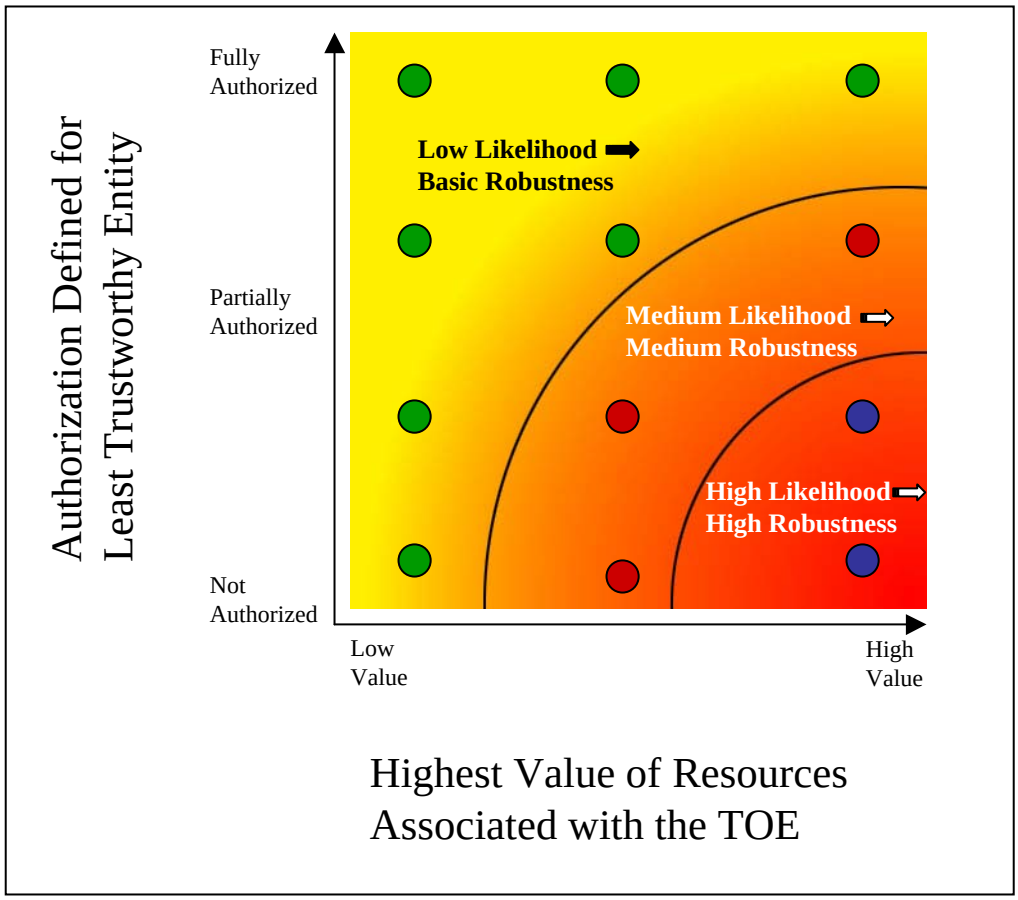

Figure 6 - Sectionalized Environments

In this second representation of environments and the robustness plane, the “dots” represent given instantiations of environments; like-colored dots define environments with a similar likelihood of attempted compromise. Correspondingly, a TOE with a given robustness should provide sufficient protection for environments characterized by like-colored dots. In choosing the appropriateness of a given robustness level TOE PP for an environment, then, the user must first consider the lowest authorization for an entity as well as the highest value of the resources in that environment. This should result in a “point” in the chart above, corresponding to the likelihood that that entity will attempt to compromise the most valuable resource in the environment. The appropriate robustness level for the specified TOE to counter this likelihood can then be chosen.

The difficult part of this activity is differentiating the authorization of various entities, as well as determining the relative values of resources; (e.g., what constitutes "low value" data vs. "medium value" data). Because every organization will be different, a rigorous definition is not possible. In Section 3.5 of this PP, the targeted threat level for a basic robustness Authorization Server TOE is characterized. This information is provided to help organizations insure that the functional requirements specified by this basic robustness PP are appropriate for their intended application of a compliant Authorization Server.

Basic robustness TOEs falls in the upper left area of the previously discussed robustness figures. A Basic Robustness TOE is considered sufficient for low threat environments or where compromise of protected information will not have a significant impact on mission objectives. This implies that the motivation of the threat agents will be low in environments that are suitable for TOEs of this robustness. In general, basic robustness results in "good commercial practices" that counter threats based in casual and accidental disclosure or compromise of data protected by the TOE.

Threat agent motivation can be considered in a variety of ways. One possibility is that the value of the data processed or protected by the TOE will generally be seen as of little value to the adversary (i.e., compromise will have little or no impact on mission objectives). Another possibility, (where higher value data is processed or protected by the TOE) is that procuring organizations will provide other controls or safeguards (i.e., controls that the TOE itself does not enforce) in the fielded system in order to increase the threat agent motivation level for compromise beyond a level of what is considered reasonable or expected to be applied.

The following subsections address:

- Assumptions about the security aspects of a compliant TOE environment;
- Threats to TOE assets or to the TOE environment which must be countered; and
- Organizational security policies that compliant TOEs must enforce.

Assumptions

This section contains assumptions regarding the security environment and the intended usage of the TOE. The specific conditions identified in Table 2 are assumed to exist in a PP-compliant TOE environment.

Table 2 – TOE Assumptions

IDENTIFICATION	DESCRIPTION
A.IT_ACCESS	The TOE has access to all the IT System data it needs to perform its functions.
A.LOWEXP	The threat of malicious attacks aimed at discovering exploitable vulnerabilities is considered low.

IDENTIFICATION	DESCRIPTION
A.MANAGE	There will be one or more competent individuals assigned to manage the TOE and the security of the information it contains.
A.NO_EVIL	Administrators are non-hostile, appropriately trained and follow all administrator guidance.
A.NO_TOE_BYPASS	Principals cannot gain access to resources protected by the TOE without passing through the TOE access control mechanisms.
A.PHYSICAL	The IT environment provides the TOE with appropriate physical security, commensurate with the value of the IT assets protected by the TOE.
A.SCALABLE	The TOE environment is appropriately scalable to provide support to the IT Systems in the organization it is deployed.

Threats to the TOE

In addition to helping define the robustness appropriate for a given environment, the threat agent is a key component of the formal threat statements in the PP. Threat agents are typically characterized by a number of factors such as ***expertise***, ***available resources***, and ***motivation***. Because each robustness level is associated with a variety of environments, there are corresponding varieties of specific threat agents (that is, the threat agents will have different combinations of motivation, expertise, and available resources) that are valid for a given level of robustness. The following discussion explores the impact of each of the threat agent factors on the ability of the TOE to protect itself (that is, the robustness required of the TOE).

The ***motivation*** of the threat agent seems to be the primary factor of the three characteristics of threat agents outlined above. Given the same expertise and set of resources, an attacker with low motivation may not be as likely to attempt to compromise the TOE. For example, an entity with no authorization to low value data none-the-less has low motivation to compromise the data; thus a basic robustness TOE should offer sufficient protection. Likewise, the fully authorized user with access to highly valued data similarly has low motivation to attempt to compromise the data, thus again a basic robustness TOE should be sufficient.

Unlike the motivation factor, however, the same can't be said for ***expertise***. A threat agent with low motivation and low expertise is just as unlikely to attempt to compromise a TOE as an attacker with low motivation and high expertise; this is because the attacker with high expertise does not have the motivation to compromise the TOE even though they may have the expertise to do so. The same argument can be made for ***resources*** as well.

Therefore, when assessing the robustness needed for a TOE, the motivation of threat agents should be considered a "high water mark." ***That is, the robustness of the TOE should increase as the motivation of the threat agents increases.***

Having said that, the relationship between expertise and resources is somewhat more complicated. In general, if resources include factors other than just raw processing power

(money, for example), then expertise should be considered to be at the same "level" (low, medium, high, for example) as the resources because money can be used to purchase expertise. Expertise in some ways is different, because expertise in and of itself does not automatically procure resources. However, it may be plausible that someone with high expertise can procure the requisite amount of resources by virtue of that expertise (for example, hacking into a bank to obtain money in order to obtain other resources).

It may not make sense to distinguish between these two factors; in general, it appears that the only effect these may have is to lower the robustness requirements. For instance, suppose an organization determines that, because of the value of the resources processed by the TOE and the trustworthiness of the entities that can access the TOE, the motivation of those entities would be "medium." This normally indicates that a medium robustness TOE would be required because the likelihood that those entities would attempt to compromise the TOE to get at those resources is in the "medium" range. However, now suppose the organization determines that the entities (threat agents) that are the least trustworthy have no resources and are unsophisticated. In this case, even though those threat agents have medium motivation, the likelihood that they would be able to mount a successful attack on the TOE would be low, and so a basic robustness TOE may be sufficient to counter that threat.

It should be clear from this discussion that there is no "cookbook" or mathematical answer to the question of how exactly to specify the level of motivation, the amount of resources, and the degree of expertise for a threat agent so that the robustness level of TOEs facing those threat agents can be rigorously determined. However, an organization can look at combinations of these factors and obtain a good understanding of the likelihood of a successful attack being attempted against the TOE. Each organization wishing to procure a TOE must look at the threat factors applicable to their environment; discuss the issues raised in the previous paragraph; consult with appropriate accreditation authorities for input; and document their decision regarding likely threat agents in their environment. The important general points we can make are:

- The motivation for the threat agent defines the upper bound with respect to the level of robustness required for the TOE**.**

- A threat agent's expertise and/or resources that are "lower" than the threat agent's motivation (e.g., a threat agent with high motivation but little expertise and few resources) may lessen the robustness requirements for the TOE (see next point, however**).**

- The availability of attacks associated with high expertise and/or high availability of resources (for example, via the Internet or "hacker chat rooms") introduces a problem when trying to define the expertise of, or resources available to, a threat agent.

3.1.1 Threats Addressed by the TOE

Table 3 identifies the threats to the TOE. The assumed level of expertise of the attacker for all the threats is assumed to be unsophisticated.

Table 3 – TOE Threats

IDENTIFICATION	DESCRIPTION
T.ACCIDENTAL_ADMIN_ ERROR	An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.
T.ACCIDENTAL_AUDIT_ COMPROMISE	An administrative user or process may view audit records, cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking a user's action.
T.ACCIDENTAL_CRYPTO_ COMPROMISE	An administrative user or process may cause key, data or executable code associated with the cryptographic functionality to be inappropriately accessed (viewed, modified, or deleted), thus compromising the cryptographic mechanisms and the data protected by those mechanisms.
T.LOW_PRIORITY	A low priority process may exhaust resources required by the TOE.
T.MASQUERADE	A user or process may masquerade as another entity in order to gain unauthorized access to data or TOE resources.
T.POOR_DESIGN	Unintentional errors in requirements specification or design of the TOE may occur, leading to flaws that may be exploited by a casually mischievous user or program.
T.POOR_IMPLEMENTATION	Unintentional errors in implementation of the TOE design may occur, leading to flaws that may be exploited by a casually mischievous user or program.
T.POOR_TEST	Developers or test engineers may implement tests that are insufficient to demonstrate that all TOE security functions operate correctly (including in a fielded TOE) may result in incorrect TOE behavior being discovered thereby causing potential security vulnerabilities.
T.RESIDUAL_DATA	A user or process may gain unauthorized access to data through reallocation of TOE resources from one user or process to another.
T.TSF_COMPROMISE	An attacking user or process may cause, through an unsophisticated attack, TSF data, or executable code to be inappropriately accessed (viewed, modified, or deleted).
T.UNATTENDED_SESSION	A user may gain unauthorized access to an unattended session.
T.UNAUTHORIZED_ACCES S	A user or application may gain access to the data for which they are not authorized according to the TOE security policy.
T.UNIDENTIFIED_ACTIONS	The administrator may not have the ability to notice potential security violations, thus limiting the administrator's ability to identify and take action against a possible security breach.

Organizational Security Policies

An organizational security policy is a set of rules, practices, and procedures imposed by an organization to address its security needs. This section identifies the organizational security policies applicable to the PPASBRE.

All PP-compliant TOEs must address the organizational security policies described in Table 4.

Table 4 – TOE Policies

IDENTIFICATION	DESCRIPTION
P.ACCESS_BANNER	The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the system.
P.ACCOUNTABILITY	The TOE shall log all actions by authorized users such that the authorized users can be held accountable for their actions within the TOE.
P.BASIC_ROBUSTNESS	The TOE must be developed in accordance with the Basic Robustness guidelines.
P.CAPP_OS	The operating system the TOE operates on top of must be evaluated to be compliant with the Controlled Access Protection Profile.
P.COMMS	Communications exist between the TOE components (internally) and between the TOE components and the IT components.
P.CRYPTOGRAPHY	Only NIST FIPS 140-2 validated cryptography (methods and implementations) are acceptable for key management (i.e.; generation, access, distribution, destruction, handling, and storage of keys) and cryptographic services (i.e., encryption, decryption, signature, hashing, key exchange, and random number generation services).
P.HIGH_AVAILABILITY	The TOE shall include providing resource allocations to support priority of service and fault tolerance.
P.NO_GENERAL_PURPOSE	There will be no general-purpose computing or storage repository capabilities (e.g., compilers, editors, or user applications) available on the hardware platforms that the TOE administrative and authorization policy engine software are installed. If Authorization Server “Agent” software is part of the TOE, then the system on which the Agent operates is exempt from this assumption.
P.TOE_ENVIRONMENT_ ACCESS	The TOE environment will provide mechanisms that control a user’s logical access to the TOE environmental components.
P.WEB_BROWSER_PP	If administrators use a web browser to access the TOE for remote administration, they must to use software that has been evaluated to the Web Browser Protection Profile.

4 SECURITY OBJECTIVES

This chapter describes the security objectives for the TOE and the TOE's operating environment. The security objectives are divided between TOE Security Objectives (i.e., security objectives addressed directly by the TOE) and Security Objectives for the Operating Environment (i.e., security objectives addressed by the IT Environment or by non-technical or procedural means).

The security objectives identify the responsibilities of the TOE and its environment in meeting the security needs.

TOE Security Objectives

Table 5 defines the security objectives for the TOE.

Table 5 – TOE Security Objectives

IDENTIFICATION	DESCRIPTION
O.ADMIN_GUIDANCE	The TOE will provide administrators with the necessary information for secure management.
O.AUDIT_GENERATION	The TOE will provide the capability to detect and create records of security-relevant events associated with users.
O.CORRECT_TSF_OPERATION	The TOE will provide the capability to test the TSF to ensure the correct operation of the TSF at a customer's site.
O.DISPLAY_BANNER	The TOE will display an advisory warning regarding use of the TOE to the administrative users.
O.MANAGE	The TOE will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.
O.MEDIATE	The TOE must protect user data in accordance with its security policy.
O.PARTIAL_SELF_PROTECTION	The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure through its own interfaces.
O.RESIDUAL_INFORMATION	The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated.
O.TOE_ACCESS	The TOE will provide mechanisms that control a user's logical access to the TOE.

Security Objectives for the Development Environment

The following

Table 6 defines the security objectives to be met by the TOE's development environment.

Table 6 – TOE Development Environment Security Objectives

IDENTIFICATION	DESCRIPTION
OD.BASIC_ROBUSTNESS	The TOE shall be developed in accordance with the Basic Robustness requirements.
OD.CONFIGURATION_ IDENTIFICATION	The configuration of the TOE is fully identified in a manner that will allow implementation errors to be identified, corrected with the TOE being redistributed promptly.
OD.DOCUMENTED_DESIGN	The design of the TOE is adequately and accurately documented.
OD.PARTIAL_FUNCTIONAL_ TESTING	The TOE will undergo some security functional testing that demonstrates the TSF satisfies its security functional requirements.
OD.VULNERABILITY_ANALYSIS	The TOE will undergo vulnerability analysis demonstrate the design and implementation of the TOE does not contain any obvious flaws.

Security Objectives for the Operating Environment

Since this is a "software only" PP, there are several objectives that must be met by the hardware components and the underlying operating systems to provide a secure TOE Environment, including the existence of a FIPS 140-2 cryptographic module to provide services to the TOE. These include objectives that levy IT requirements on the hardware and operating system and those that can be satisfied by procedural or administrative measures.

Table 7 defines the security objectives that are to be addressed by the IT Environment or by non-technical or procedural means. All of the assumptions stated in Section 0 are considered to be security objectives for the environment. There are additional objectives for the environment, listed in the Table below. The mapping and rationale for the security objectives are provided in Section 6.

Table 7 – TOE Operating Environment Security Objectives

IDENTIFICATION	DESCRIPTION
OE.AUDIT_PROTECTION	The IT Environment will provide the capability to protect audit information.
OE.CAPP_OS	Operating systems the TOE operates on top of must be compliant with the Controlled Access Protection Profile. The operating system will therefore provide all the capabilities outlined in the CAPP security function requirements and will have been evaluated against the CAPP assurance requirements.
OE.COMMS	Sites deploying the TOE will ensure that adequate communications exist between the TOE components (internally) and between the TOE components and the IT components.
OE.CRYPTOGRAPHY	The IT environment components shall use NIST FIPS 140-2 validated cryptographic modules if they provide cryptographic services.
OE.DISPLAY_BANNER	The underlying operating system of the TOE will display an advisory warning regarding use of the TOE to administrative users logging on the platform where the TOE software is installed.
OE.FAULT_TOLERANCE	The IT environment will provided limited capabilities to support degraded fault tolerance and fail over for some TOE components
OE.IT_ACCESS	Sites deploying the TOE will ensure the TOE has access to all the IT System data it needs to perform its functions.
OE.LOWEXP	Site deploying the TOE will establish a protective environment where the threat of malicious attacks aimed at discovering exploitable vulnerabilities is considered low.
OE.MANAGE	The TOE environmental components will provide all the functions, facilities and competent individuals necessary to support the administrators in their management of the security of the environment, and restrict these functions and facilities from unauthorized use.
OE.NO_EVIL	Sites using the TOE shall ensure that administrators are non-hostile, appropriately trained and follow all administrator guidance.
OE.NO_GENERAL_PURPOSE	There will be no general-purpose computing or storage repository capabilities (e.g., compilers, editors, or user applications) available on the hardware platforms that the TOE administrative and authorization policy engine software are installed. This objective does not apply to agent software that might reside on a web server.
OE.NO_TOE_BYPASS	Principals cannot gain access to resources protected by the TOE without passing through the TOE access control mechanisms.

IDENTIFICATION	DESCRIPTION
OE.PHYSICAL	Physical security will be provided within the domain for the value of the IT assets protected by the operating system and the value of the stored, processed, and transmitted information.
OE.PRIORITY	The IT Environment will provide prioritization of resources to support the TOE.
OE.RESIDUAL_ INFORMATION	The IT Environment will ensure that any information contained in a protected resource is not released when the resource is reallocated.
OE.SCALABLE	Sites using the TOE will deploy the appropriate hardware and software environment to ensure the TOE system is scalable to provide support to the IT Systems in the organization it is deployed.
OE.TOE_ENVIRONMENT_ ACCESS	The TOE environment will provide mechanisms that control a user's logical access to the environmental components.
OE.WEB_BROWSER_PP	If administrators use a web browser to access the TOE for remote administration, they must to use software that has been evaluated to the Web Browser Protection Profile.

5 IT SECURITY REQUIREMENTS

This section provides functional and assurance requirements that must be satisfied by a Protection Profile-compliant TOE. It also provides functional requirements the IT environment must meet to deploy an Authorization Server in a secure manner, meeting the policy objectives. These requirements consist of functional components from Part 2 of the CC and assurance components from Part 3 of the CC.

TOE Functional Security Requirements

This section provides security functional requirements that must be satisfied by a PP-compliant TOE. These requirements consist of functional components from Part 2 with NIAP interpretations. Table 8 summarizes the TOE Functional Requirements to meet the stated objectives and identifies the extended requirements that were necessary to express the desired functionality or meet the NIAP Basic Robustness Consistency Guidance.

Table 8 – TOE Security Functional Requirements

TOE Security Functional Requirements	
FAU_GEN.1	Audit Data Generation
FAU_GEN.2	User Identity Association
FDP_ACC.1	Access Control Policy
FDP_ACF_(EXT).1	Access Control Functions
FDP_RIP.2	Full Residual Information Protection
FIA_AFL.1	Authentication Failure Handling
FIA_ATD.1(1)	User Attribute Definition – Administrator
FIA_ATD.1(2)	User Attribute Definition – Principal
FIA_ATD.1(3)	User Attribute Definition – Authorized Application
FIA_SOS.1	Verification of Secrets
FIA_UAU.2	Timing of Authentication
FIA_UID.2	Timing of Identification
FMT_MOF.1(1)	Management of Security Functions Behavior (access policy)
FMT_MOF.1(2)	Management of Security Functions Behavior (authorized applications)
FMT_MOF.1(3)	Management of Security Functions Behavior (audit)

FMT_MSA.1(1)	Management of Security Attributes – Attribute Management
FMT_MSA.1(2)	Management of Security Attributes – Attribute Authority
FMT_MSA.2	Secure Security Attributes
FMT_MSA.3	Static Attribute Initialization
FMT_MTD.1	Management of TSF Data
FMT_SMF.1	Specification of Management Functions
FMT_SMR.1	Security Management Roles
FTA_TAB.1	Default TOE Access Banners
FPT_TST_(EXT)1.1	TSF Testing

5.1.1 Class FAU: Security Audit

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:

- Start-up and shutdown of the audit functions;
- All auditable events for the *basic* level of audit **as identified in Table 9**;
- [selection: [assignment: *events at a basic level of audit introduced by the inclusion of additional SFRs determined by the ST author*], [assignment: *events commensurate with a basic level of audit introduced by the inclusion of extended requirements determined by the ST author*], "*no additional events*"].

Application Note: *For the selection, the ST author should choose one of the assignments (as detailed in the following paragraphs), or select "no additional events".*

For the first assignment, the ST author augments the table (or lists explicitly) the audit events associated with the basic level of audit for any SFRs that the ST author includes that are not included in this PP.

Likewise, for the second assignment the ST author includes audit events that may arise due to the inclusion of any extended requirements not already in the PP. Because "basic" audit is not defined for such requirements, the ST author will need to determine a set of events that are commensurate with the type of information that is captured at the basic level for similar requirements.

If no additional (CC or extended) SFRs are included, or if additional SFRs are

included that do not have "basic" audit associated with them, then it is acceptable to assign "no additional events" in this item.

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

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

b. For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [**selection by ST Author: "**information specified in column three of Table 9 below", none].

Application Note: *In column 3 of the Table 9, "if applicable" is used to designate data that should be included in the audit record if it "makes sense" in the context of the event that generates the record. If no other information is required (other than that listed in "a") for a particular audit event type, then an assignment of "none" is acceptable.*

Application Note: *Section5.1.7 details auditable events generated by the non-OS portion of the IT Environment. Table 1 of the CAPP specifies auditable events generated by the operating system.*

Dependency: *FPT_STM.1 Reliable Time Stamps*

Table 9 – Auditable Events

REQUIREMENT	AUDITABLE EVENTS	ADDITIONAL AUDIT RECORD CONTENTS (AS APPROPRIATE)
FDP_ACF_(EXT).1	All requests to perform an operation on an object covered by the SFP.	The specific security attributes used in making an access check.
FIA_AFL.1.1	The reaching of the threshold for the unsuccessful authentication attempts.	The claimed identity of the user attempting to gain access
FIA_AFL.1.2	The actions (e.g., disabling of a terminal) taken and the subsequent, if appropriate, restoration to the normal state (e.g., re-enabling of a terminal).	The claimed identity of the user attempting to gain access
FIA_SOS.1	Rejection or acceptance by the TSF of any tested secret	Identification of any changes to the defined quality metrics.
FIA_UAU.2	All use of the authentication mechanism;	Claimed identity of user being authenticated, if that user exists in PADS
FIA_UID.2	All use of the user identification mechanism, including the user identity provided.	Claimed identity of the user using the identification mechanism, if that user exists in PADS
FMT_MOF.1(1)	All modifications to the access policy settings.	Identity of administrator making the modifications

REQUIREMENT	AUDITABLE EVENTS	ADDITIONAL AUDIT RECORD CONTENTS (AS APPROPRIATE)
FMT_MOF.1(2)	All modifications to the list of authorized applications.	Identity of administrator making the modifications
FMT_MOF.1(3)	All modifications to the audit behavior.	Identity of administrator making the modifications
FMT_MSA.1(1)	All modifications of the values of security attributes.	Identity of administrator making the modifications
FMT_MSA.1(2)	All queries of the values of security attributes.	Identity of authorized application making the queries
FMT_MSA.2	All offered and rejected values a security attribute	All offered and accepted secure values for a security attribute.
FMT_MSA.3	All modifications of the default settings of permissive or restrictive rules	Identity of administrator making the modifications
FMT_MSA.3	All modifications of the initial values of static security attributes	Identity of administrator making the modifications
FMT_MTD.1	All modifications to the values of TSF data.	Identity of administrator making the modifications
FMT_SMF.1	Use of the management functions.	Identity of administrator using the management functions
FMT_SMR.1	Modifications to the group of users that are part of a role	Identity of administrator making the modifications
FPT_TST_(EXT)1.1	Execution of the TSF self tests and the results of the tests.	
FRU_FLT.1	Any failure detected by the TSF. Plus all TOE capabilities being discontinued due to a failure.	Identity of component that failed

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.

Application Note: *For failed login attempts no user association is required because the user is not under TSF control until after a successful identification/authentication.*

Dependencies: *FAU_GEN.1 – Audit Data Generation*
FIA_UID.1 – Timing of identification

5.1.2 User data protection (FDP)

FDP_ACC.1	**Subset Access Control Policy**
FDP_ACC.1.1	The TSF shall [selection of one by ST Author: *enforce, provide an access control decision based on*] the [Authorization Server Access Control Policy] on [principals as subjects, [assignment by ST author: *list of named objects*]] as objects, and all the operations among subjects and objects covered by the Authorization Server Access Control policy.]
Application Note:	*If the TOE performs enforcement of authorization decisions, then the Authorization Server Access Control Policy covers access to the resources that the TOE protects. It is not a single "standard" policy, like Discretionary Access Control (DAC). The Authorization Server Access Control Policy is based on the Security Administrator's defined rules or operations. The subjects are the principals (the web browser users or web services), and the named objects are the resources (web server, directories, files, or objects) that the Authorization Server is protecting.*
Application Note:	*This requirement (FDP_ACC.1) is applicable only if the TOE enforces or provides an access control decision. If the TOE acts only as attribute authority, then this requirement is not applicable.*
Dependency:	*FDP_ACF.1 – Security attribute based access control*
FDP_ACF_(EXT).1	**Security Attribute Based Access Control**
FDP_ACF_(EXT).1.1	The TSF shall perform an access control decision and [selection of one of more by ST Author: *enforce the decision, provide the decision*] based on the [Authorization Server Access Control Policy] to objects based on the following: [assignment: *list of subjects and objects controlled under the Authorization Server Access Control Policy, and for each, the relevant security attributes*].
FDP_ACF_(EXT).1.2	The TSF shall [selection of one by ST Author: *enforce, provide an access control decision based on"*] the following rules to determine if an operation among controlled subjects and controlled objects is allowed [assignment by ST Author: *rules governing access among controlled subject and controlled objects using controlled operations on controlled objects*].
FDP_ACF_(EXT).1.3	The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [selection: [assignment: *rules, based on security attributes, that explicitly authorize access of subjects to objects], "no additional rules"]*
FDP_ACF_(EXT).1.4	The TSF shall explicitly deny access of subjects to objects based on the [selection: [assignment*: rules, based on security attributes,*

that explicitly deny access of subjects to objects], "*no additional explicit denial rules*"].

Application Note: *The Authorization Server Access Control Policy is the set of rules that mediate access control on the resources protected by a TOE based on security attributes associated with subjects and objects. The parameters of these rules will be highly dependant on the nature of the TOE and the implementation, and the details are specified by the Security Administrator. The ST Author shall specify, for each controlled subject and object, the security attributes and/or named groups of security attributes that the function will use in the specification of the rules.*

Application Note: *This requirement (FDP_ACF_(EXT).1) is applicable only if the TOE enforces or provides an access control decision. If the TOE acts only as attribute authority, then this requirement is not applicable.*

Dependencies: *FDP_ACC.1 – Subset access control*
FMT_MSA.3 – Static attribute initialization

FDP_RIP.2 **Full residual information protection**

FDP_RIP.2.1 The TSF shall ensure that any previous information content of a TSF Resource is made unavailable upon the *allocation of the resource to* all objects.

Application Note: *The term "resource" is used numerous times in this PP to designate the subjects of an authorization decision. In this SFR, the term "TSF Resource" is used to clarify that the term refers to an internal TSF resource.*

5.1.3 Identification and authentication (FIA)

TOE security functions implemented by a probabilistic or permutational mechanism (e.g., password or hash function) are required (at EAL2 and higher).

FIA_AFL.1 **Authentication failure handling**

FIA_AFL.1.1 The TSF shall detect when [a **Security** Administrator configurable positive integer within [assignment: *range of acceptable values*]] **of** unsuccessful authentication attempts occur related to **administrators attempting to authenticate to the TOE, and** [selection of one or more by ST Author: *none, authorized applications authenticating to the TOE, and principals authenticating to the TOE*].

Application Note: *When the TOE either acts as an Attribute Authority or provides an interface for authorized applications to query the authorization decision function, it must authenticate the requesting application, therefore authorized applications shall be included in the selection. When the TOE includes Authentication Server functionality to authenticate principals directly, principals shall be included in the selection.*

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall [prevent [selection of one

or more by ST Author: *the remote administrators, application, principal*] from performing activities that require authentication until an action is taken by the Security Administrator].

Dependencies: *FIA_UAU.1 – Timing of authentication*

FIA_ATD.1(1) **User attribute definition – Administrator**

FIA_ATD.1.1(1) The TSF shall maintain the following list of security attributes belonging to individual **administrative** users:

- [Administrative user identifier,
- Administrator class (i.e. Security Administrator vs. Audit Administrator)],
- Authentication data,
- [assignment: *list of additional security attributes as determined by the ST Author*]].

FIA_ATD.1(2) **User attribute definition - Principal**

FIA_ATD.1.1(2) The TSF shall maintain the following list of security attributes belonging to individual **principal** users:

- [User identifier,
- Group membership,
- [assignment: *list of security attributes as determined by the ST Author*]].

Application Note: *If the TOE is performing principal authentication, assignment shall include "authentication data".*

Application Note: *This requirement (FIA_ATD.1(2)) is applicable only if the TOE enforces or provides an access control decision. If the TOE acts only as attribute authority, then this requirement is not applicable.*

FIA_ATD.1(3) **User attribute definition – Authorized Application**

FIA_ATD.1.1(3) The TSF shall maintain the following list of security attributes belonging to individual **authorized applications**: [selection: *"none"*, [*Application identifier, Authentication data*, [assignment: *list of security attributes as determined by the ST Author*]]].

Application Note: *If the TOE does not provide attribute authority function for applications, "none" shall be selected. If the TOE provides the attribute authority function, the other selection shall be made.*

FIA_SOS.1	**Specification and Verification of Secrets**
FIA_SOS.1.1	The TSF shall provide a mechanism to verify that secrets meet [the condition that passwords must contain a minimum of 8 alpha numeric characters with at least one numeric character, and shall not be reused within a Security Administrator defined window of password changes].
FIA_UAU.2	**Timing of authentication**
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.
Application Note:	*The selection of authenticated entities that are referred to as "user" in this context are selected in FIA_AFL.1.1 above.*
Dependencies:	*FIA_UID.1 – Timing of identification*
FIA_UID.2	**Timing of Identification**
FIA_UID.2.1	The TSF shall require each user to identify itself before allowing any other TSF-mediated actions on behalf of that user.

5.1.4 Security management (FMT)

FMT_MOF.1(1)	**Management of security functions behavior (access policy)**
FMT_MOF.1.1(1)	The TSF shall restrict the ability to *determine and modify the behavior of* the functions [configure the Authorization Server Access Control Policy settings] to [the Security Administrator].
Application Note:	*This requirement (FMT_MOF.1(1)) is applicable only if the TOE enforces or provides an access control decision. If the TOE acts only as attribute authority, then this requirement is not applicable.*
Dependencies:	*FMT_SMF.1 Specification of Management Functions* *FMT_SMR.1 Security Roles*
FMT_MOF.1(2)	**Management of security functions behavior (authorized applications)**
FMT_MOF.1.1(2)	The TSF shall restrict the ability to *determine and modify the behavior of* the functions: [configure the list of Authorized Applications and specify their security attributes] to [the Security Administrator].
Dependencies:	*FMT_SMF.1 Specification of Management Functions* *FMT_SMR.1 Security Roles*

FMT_MOF.1(3)	**Management of security functions behavior (audit)**
FMT_MOF.1.1(3)	The TSF shall restrict the ability to *enable, disable, determine and modify the behavior of* the functions [related to the security audit generation] to the [Audit Administrator].
Application Note:	*For the audit function, enable and disable refer to the ability to enable or disable the audit mechanism as a whole. "Determine the behavior" means the ability to determine specifically what on the system is being audited, while "modify the behavior" means the ability to set or unset specific aspects of the audit mechanism, such as what user behavior is audited, etc.*
Dependencies:	*FMT_SMF.1 Specification of Management Functions* *FMT_SMR.1 Security Roles*
FMT_MSA.1(1)	**Management of security attributes – Attribute Management**
FMT_MSA.1.1(1)	The TSF shall enforce the [Authorization Server Access Control Policy] to restrict the ability to *change_default, query, modify or delete* the security attributes [associated with both principals and protected resources which are used for access control permission rules] to [a designated Security Administrator].
Application Note:	*This requirement restricts management of the privilege attributes and the security attributes that make up a protected resources access control requirements (or rules) to a designated Security Administrator.*
Application Note:	*This requirement (FMT_MSA.1(1)) is applicable only if the TOE enforces or provides an access control decision. If the TOE acts only as attribute authority, then this requirement is not applicable.*
Dependency:	*FDP_ACC.1 Subset access control* *FMT_SMF.1 Specification of Management Functions* *FMT_SMR.1 Security roles*
FMT_MSA.1(2)	**Management of security attributes – Attribute Authority**
FMT_MSA.1.1(2)	The TSF shall enforce the [Authorization Server Access Control Policy] to restrict the ability *query* the security attributes [associated with both principals and protected resources which are used for access control permission rules] to [a designated Authorized Application].
Application Note:	*This requirement defines the authorized applications for which the TOE will act as an Attribute Authority.*
Dependency:	*FDP_ACC.1 Subset access control* *FMT_SMF.1 Specification of Management Functions* *FMT_SMR.1 Security roles*
FMT_MSA.2	**Secure Security Attributes**

FMT_MSA.2.1	The TSF shall ensure that only secure values are accepted for security attributes**, in particular, user authentication passwords shall be considered insecure if they have been previously used within a Security Administrator configurable number of password changes**.
Dependencies:	*ADV_SPM.1 Informal TOE security policy model* *FDP_ACC.1 Subset access control* *FMT_MSA.1 Management of security attributes* *FMT_SMR.1 Security roles*

FMT_MSA.3 Static attribute initialization

FMT_MSA.3.1	The TSF shall enforce the [Authorization Server Access Control Policy] to provide *restrictive* default values for security attributes that are used to enforce the SFP.
Application Note:	*"Restrictive" in this case means that by default access is not authorized to a protected resource unless an explicit rule in the access control policy allows the access. By default, access to protected data is not allowed.*
FMT_MSA.3.2	The TSF shall allow the [Security Administrator] to specify alternative initial values to override the default values when an object or information is created.
Application Note:	*This requirement (FMT_MSA.3) is applicable only if the TOE enforces or provides an access control decision. If the TOE acts only as attribute authority, then this requirement is not applicable.*
Dependencies:	*FMT_MSA.1 Management of security attributes* *FMT_SMR.1 Security roles*

FMT_MTD.1 Management of TSF Data

FMT_MTD.1.1	The TSF shall restrict the ability to *change_default, query, modify, delete, clear* [all TSF data, including system configuration files and the advisory warning message referenced in FTA_TAB.1], to [the Security Administrator role].
Dependencies:	*FMT_SMF.1 Specification of Management Functions* *FMT_SMR.1 Security roles*

FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1	The TSF shall be capable of performing the following security management functions: [assignment by ST Author: *list of security management functions to be provided by the TSF*].

FMT_SMR.1	**Security Management Roles**
FMT_SMR.1.1	The TSF shall maintain the roles: [Security Administrator; Audit Administrator; [selection of one or more by ST Author: *Authorized Application*, [assignment by ST Author: *other roles*], *none*]].

Application Note: If the TOE acts as an Attribute Authority for certain applications, as specified in FMT_MSA.1(2), then the ST Author shall include Authorized Application in the list of security roles.

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

Dependency: FIA_UID.1 Timing of authentication

5.1.5 Protection of the TOE Security Functions (FPT)

FPT_TST_(EXT)1.1	**TSF testing**
FPT_TST_(EXT)1.1.1	The TSF shall provide security administrator with the capability to verify the integrity of the following TSF data: [TOE system configuration files].
FPT_TST_(EXT)1.1.2	The TSF shall provide security administrator with the capability to verify the integrity of stored TSF executable code.

5.1.6 TOE Access (FTA)

FTA_TAB.1	**Default TOE access banners**
FTA_TAB.1.1	Before establishing a**n administrative** user session, the TSF shall display an advisory warning message regarding unauthorized use of the TOE.
Application Note:	*The access banner should appear before or in conjunction with the administrative users of the TOE being prompted for their user identification and authentication. Principals requesting access to protected web resources are not provided the banner. The intent of this requirement is to advise administrative users of warnings regarding the unauthorized use of the TOE and to provide the Security Administrator with control over what is displayed (e.g., if the Security Administrator chooses, they can remove banner information that informs the user of the product and version number).* *Principals are not required to view the access banner.*

Security Requirements for the IT Environment

This Protection Profile provides functional requirements for the IT Environment. Since this is a "software only" PP, to deploy this software in a secure manner a significant number of requirements must be met by the IT Environment. First, the software must be installed on a securely configured operating system that is compliant with the Control Access PP (CAPP). The CAPP OS will provide security functionality to meet a wide range of IT objectives including

Discretionary Access Control (DAC), audit services (including generation, protection, review and analysis), administrator Identification and Authentication, Self-Protection, and reliable time stamping. Complete details of the CAPP OS security requirements are provided in the CAPP PP. The IT environment also includes authorized IT entities, authorized applications, web servers (with files to be protected), cryptographic modules) and any IT entities that are used by administrators to remotely administer the TOE (e.g., a workstation with a browser).

Table 10 summarizes the IT Environment Functional Requirements that are levied on IT Environment in addition to the CAPP compliant operating system requirements. These additional requirements are necessary to meet the stated objectives. Table 11 identifies the extended requirements that were necessary to express the desired functionality or meet the NIAP Basic Robustness Consistency Guidance. The detailed explanation of these requirements is also provided below.

For reference, Table 12 lists the SFRs from CAPP.

Table 10 - IT Environment Security Functional Requirements

IT Environment Functional Components (from CC Part 2 and NIAP Interpretations)	
FAU_GEN.1	Audit Data Generation
FAU_GEN.2	Audit Data Generation
FCS_CKM.1	Cryptographic Key Generation
FCS_CKM.2	Cryptographic Key Distribution
FCS_CKM.4	Cryptographic Key Destruction
FCS_COP.1	Cryptographic operation
FPT_FLS.1	Failure with preservation of secure state
FPT_ITT.1	Internal TOE TSF Data Transfer
FPT_RCV. 1	Recovery to a Known State
FPT_TST_(EXT)2.1	IT Environment Testing
FRU_FLT.1(1)	Degraded Fault Tolerance (Electrical Power)
FRU_FLT.1(2)	Degraded Fault Tolerance (Web Server Failover)
FRU_PRS.2	Full Priority of Service
FRU_RSA.1	Maximum quotas (transport-layer quotas)
FTA_SSL.1	TSF-initiated Session Locking

IT Environment Functional Components (from CC Part 2 and NIAP Interpretations)	
FTA_SSL.2	User-initiated Locking
FTA_SSL.3	TSF-initiated Termination
FTA_TAB.1	Default TOE Access Banners
FTP_ITC.1	Inter-TSF Trusted Channel

Table 11 – IT Environment Extended Security Functional Requirements

IT Environment Extended Functional Components	
FPT_TST_(EXT)2.1	TSF Testing

Table 12 - CAPP Security Functional Requirements

CAPP Security Functional Requirements	
FAU_GEN.1	Audit Data Generation
FAU_GEN.2	User Identity Association
FAU_SAR.1	Audit Review
FAU_SAR.2	Restricted Audit Review
FAU_SAR.3	Selectable Audit Review
FAU_SEL.1	Selective Audit
FAU_STG.1	Guarantees of Audit Data Availability
FAU_STG.3	Action in Case of Possible Audit Data Loss
FAU_STG.4	Prevention of Audit Data Loss
FDP_ACC.1	Discretionary Access Control Policy
FDP_ACF.1	Discretionary Access Control Policy Functions
FDP_RIP.2	Object Residual Information Protection
FIA_ATD.1	User Attribute Definition
FIA_SOS.1	Strength of Authentication Data

FIA_UAU.1	Timing of Authentication
FIA_UAU.7	Protected Authentication Feedback
FIA_UID.1	Timing of Identification
FIA_USB.1	User Subject Binding
FMT_MSA.1	Management of Object Security Attributes
FMT_MSA.3	Static Attribute Initialization
FMT_MTD.1	Management of the Audit Trail
	Management of Audited Events
	Management of User Attributes
	Management of Authentication Data
FMT_REV.1	Revocation of User Attributes
	Revocation of Object Attributes
FMT_SMR.1	Security Management Roles
FPT_AMT.1	Abstract Machine Testing
FPT_STM.1	Reliable Time Stamps

5.1.7 Class FAU: Security Audit

FAU_GEN.1 Audit Data Generation

FAU_GEN.1-1 The **IT Environment** shall be able to generate an audit record of the following auditable events:

- Start-up and shutdown of the audit functions;
- All auditable events for the basic level of audit **as identified in Table 13**;
- [selection: [assignment: *events at a basic level of audit introduced by the inclusion of additional SFRs determined by the ST author*], [assignment: *events commensurate with a basic level of audit introduced by the inclusion of extended requirements determined by the ST author*], "*no additional events"*].

FAU_GEN.1.2 The **IT Environment** shall record within each audit record at least the following information:

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

b. For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [*information specified in column three of Table 13 below*].

Application Note: *In column 3 of the Table 13, "if applicable" is used to designate data that should be included in the audit record if it "makes sense" in the context of the event that generates the record. If no other information is required (other than that listed in "a") for a particular audit event type, then an assignment of "none" is acceptable.*

Table 1 of the CAPP specifies auditable events generated by the operating system.

Dependency: *FPT_STM.1 Reliable Time Stamps*

Table 13 – Auditable Events

REQUIREMENT	AUDITABLE EVENTS	ADDITIONAL AUDIT RECORD CONTENTS (AS APPROPRIATE)
FCS_CKM.1	Success and failure of the activity	The object attribute(s), and object value(s) excluding any sensitive information (e.g. secret or private keys).
FCS_CKM.2	Success and failure of the activity	The object attribute(s), and object value(s) excluding any sensitive information (e.g. secret or private keys).
FCS_CKM.4	Success and failure of the activity	The object attribute(s), and object value(s) excluding any sensitive information (e.g. secret or private keys).
FCS_COP.1	Success and failure, and the type of cryptographic activity	Any applicable cryptographic mode(s) of operation, subject attributes and object attributes.
FPT_FLS.1	If possible, failure of the IT Environment	
FPT_RCV.1	The fact that a failure or service discontinuity occurred, resumption of regular operation, and type of failure or service discontinuity.	

REQUIREMENT	AUDITABLE EVENTS	ADDITIONAL AUDIT RECORD CONTENTS (AS APPROPRIATE)
FPT_TST_(EXT)2.1	Execution of the self tests and the results of the tests.	
FRU_FLT.1	Any failure detected by the IE Environment, and all IT Environment capabilities being discontinued due to a failure.	
FRU_PRS.2	Rejection of operation based on the use of priority within an allocation, and all attempted uses of the allocation function which involves the priority of the service functions.	
FRU_RSA.1	Rejection of allocation operation due to resource limits, and all attempted uses of the resource allocation functions for resources that are under control of the IT Environment.	
FTA_SSL.1	Locking of an interactive session by the session locking mechanism, successful unlocking of an interactive session, and any attempts at unlocking an interactive session.	
FTA_SSL.2	Locking of an interactive session by the session locking mechanism, successful unlocking of an interactive session, and any attempts at unlocking an interactive session.	
FTA_SSL.3	Termination of an interactive session by the session locking mechanism.	
FTP_ITC.1	All attempted uses of the trusted channel functions.	The initiator and target of the failed trusted channel functions, and the success or failure of the operation.

FAU_GEN.2 **User Identity Association**

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

Dependencies: *FAU_GEN.1 – Audit data generation*
FIA_UID.1 – Timing of identification

5.1.8 Class FCS: Cryptographic Support

The cryptographic requirements are structured to support the use of FIPS 140-2 validated cryptographic modules to protect communications between TOE components and the IT environments. Transport-Layer Security (TLS v1.0) is the most common means of securing connections between the TOE and the other IT products, and shall be configured to rely solely on FIPS approved cryptographic algorithms for security.

FCS_CKM.1 **Cryptographic key generation**

FCS_CKM.1.1 The **IT Environment** shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [assignment by ST Author, in accordance with Table 14 below] and specified cryptographic key sizes [assignment by ST Author, with values greater or equal to those specified in Table 14 below] that meet the following: [[assignment by ST Author, algorithm standard as specified in Table 14 below], using a FIPS 140-2 validated cryptographic module].

Dependency: *FCS_CKM.2 Cryptographic key distribution*
FCS_CKM.4 – Cryptographic key destruction
FMT_MSA.2 – Secure security attributes

Application Note: *Table 14 details the standards, algorithms, and key sizes that shall be used for cryptographic operations. This table reflects best commercial practices. Implementations must use FIPS 140-2 validated cryptographic modules. The ST author shall refine or iterate this requirement by selecting the cryptographic functions provided by the cryptographic modules, by selecting the modes (if applicable), and key sizes greater than or equal to those specified in Table 14.*

Table 14 - Cryptographic Operation, Standards, Algorithms, and Key Sizes

CRYPTOGRAPHIC OPERATION	STANDARD	ALGORITHM	MINIMUM KEY SIZE
Hashing	FIPS 180-2	SHA-1	N/A
		SHA-256	N/A
		SHA-384	N/A
		SHA-512	N/A
Digital Signature	FIPS 186-2	RSA (X9.31)	1024
		RSA (PKCS-1, V2.1 -- V1.5, PSS)	1024
		ECDSA	192

Key Transfer	ANSI X9.44	RSA	1024
	PKCS-1 V2.1	RSA	1024
Key Exchange	ANSI X9.42	DH	1024
	ANSI X9.63	ECDH	192
Encryption	FIPS 197	AES	128
HMAC	FIPS 198	SHA-1	80

FCS_CKM.2 Cryptographic Key Distribution

FCS_CKM.2.1 The **IT Environment** shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method [as specified in Table 14 above] that meets the following: [algorithm standard as specified in Table 14 below and in the module that is FIPS 140-2 validated].

Dependencies: *FCS_CKM.1 – Cryptographic key generation*
FCS_CKM.4 – Cryptographic key destruction
FMT_MSA.2 – Secure security attributes

Application Note: *The ST author shall refine or iterate this requirement by selecting the cryptographic functions provided by the cryptographic modules, by selecting the modes (if applicable), and key sizes greater than or equal to those specified in Table 14. The only appropriate cryptographic functions for key distribution are key transfer, key exchange, and encryption.*

Application Note: *Key size selection shall be commensurate with the security of the key being protected.*

FCS_CKM.4 Cryptographic Key Destruction

FCS_CKM.4.1 The **IT Environment** shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [which zeroizes all plaintext cryptographic keys and other unprotected security parameters within the device] that meets the following: [FIPS PUB 140-2, Security Level 1].

Dependencies: *FCS_CKM.1 – Cryptographic key generation*
FMT_MSA.2 – Secure security attributes

FCS_COP.1 Cryptographic Operation

FCS_COP.1.1 The **IT Environment** shall perform [assignment: *list of cryptographic operations as specified in* Table 14] in accordance with a specified cryptographic algorithm: [as specified in Table 14 above] and cryptographic key sizes [as specified in Table 14

above] that meet the following: [algorithm standard as specified in Table 14 below and in the module that is FIPS 140-2 validated].

Dependencies: *FCS_CKM.1 – Cryptographic key generation*
FCS_CKM.4 – Cryptographic key destruction
FMT_MSA.2 – Secure security attributes

Application Note: *All communications between a remote administrator and the TOE, authorized applications and the TOE, and between TOE components shall be protected via TLS or similar mechanisms, configured to use the algorithms and key sizes specified in Table 14 above. If the authorized administrator is using a COTS browser as part of their IT environment, then that browser should meet these security requirements. The associated cryptographic module(s) must comply at a minimum with FIPS PUB 140-2 Level 1. The intent of this requirement is not for the evaluator to perform a FIPS PUB 140-2 evaluation; rather, the evaluator will check for a certificate, verifying that the module completed a FIPS PUB 140-2 evaluation.*

Application Note: *The ST author shall refine or iterate this requirement by selecting the cryptographic functions provided by the cryptographic modules, by selecting the modes (if applicable), and key sizes greater than or equal to those specified in Table 14*

5.1.9 Class FPT: Protection of the TOE Security Functions

FPT_FLS.1 **Failure with Preservation of Secure State**

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

- [Operating System software failure,
- Other IT Environment software components failures, and
- Software failures on interfaces between IT components and the TOE].

Dependencies: *ADV_SPM.1 Informal TOE security policy model*

FPT_ITT.1 **Internal TOE TSF Data Transfer**

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

Application Note: *This requirement provides integrity and confidentiality service between the Authorization Server components.*

The IT Environment component that provides these services may be software or hardware, and will be configured to use the cryptographic algorithms as specified above. Examples might include a TLS enabled web server with or without hardware TLS acceleration, a software IPSec VPN client, or a hardware IPSec router.

FPT_RCV. 1	**Recovery to Known State**
FPT_RCV. 1.1	After [• Operating System software failure, • Other IT Environment software components failures, and • Software failures on interfaces between IT components and the TOE]. the **IT Environment** shall enter a maintenance mode where the ability to return to a secure state is provided.
Dependencies:	*AGD_ADM.1 – Administrator guidance* *ADV_SPM.1 – Informal TOE security policy model*
FPT_TST_(EXT)2.1	**IT Environment Testing**
FPT_TST_(EXT)2.1.1	The **IT Environment** shall run a suite of self-tests *during initial start-up, periodically during normal operation as specified by the authorized administrator*, and at the [request of an authorized administrator] to demonstrate the correct operation of the **IT Environment security function**.
FPT_TST_(EXT)2.1.2	The **IT Environment** shall provide **the authorized** administrator with the capability to verify the integrity of the following **IT Environment security function** data: [assignment: system configuration files including cryptographically-related data for which integrity validation is required].
FPT_TST_(EXT)2.1.3	The **IT Environment** shall provide **the authorized** administrator with the capability to verify the integrity of stored **IT Environment security function** executable code.
Dependencies:	*FMT_AMT.1 – Abstract machine testing*

5.1.10 Class FRU: Resource Utilization

FRU_FLT.1(1)	**Degraded Fault Tolerance (Electrical Power)**
FRU_FLT.1.1(1)	The **IT Environment** shall ensure the operation of **[**assignment: • Hardware platforms for TOE components (except web agents) • Operating Systems for TOE components (expect web agents)] when the following failures occur: [• Loss of primary power to TOE and IT Environment components].

Application note:	*To prevent a failure of the Authorization Server system in the event of a primary electrical power failure to the IT hardware that host the TOE and the related IT environment components should be on Uninterrupted Power Supplies (UPS) and connected to a secondary back up power source.*
Dependency:	*FPT_FLS.1 – Failure with preservation of secure state*

FRU_FLT.1(2)	**Degraded Fault Tolerance (Web Server Failover)**
FRU_FLT.1.1(2)	The **IT Environment** shall ensure the operation of [authorization decisions] when the following failures occur: [single instance of failure of authorization decision software].
Application note:	*In the event of software failure of an authorization server policy engine, the web agents should have an automated failover capability allow access to an alternate authorization server policy engine, thereby continuing service.*

Dependency: FPT_FLS.1 – Failure with preservation of secure state

FRU_PRS.2	**Full Priority of Service**
FRU_PRS.2.1	The **IT Environment** shall assign a priority to each subject in the **IT Environment** security function.
FRU_PRS.2.2	The **IT Environment** shall ensure that each access to all shareable resources shall be mediated on the basis of the subject's assigned priority.

FRU_RSA.1	**Maximum quotas (transport-layer quotas)**
FRU_RSA.1.1	The **IT Environment** shall enforce maximum quotas of the following resources: [transport-layer representation] that *users or other subjects* can use *over a specified period of time*.
Application Note:	*"transport-layer representation" refers specifically to the TCP SYN attack, where half-open connections are established thus exhausting the connection table resource. If the IT component does not implement the TCP/IP protocol, this requirement would apply to a similar type of transport-layer entity for that IT environment protocol stack.*

5.1.11 Class FTA: TOE Access

FTA_SSL.1	**TSF-initiated session locking**
FTA_SSL.1.1	The **IT Environment** shall lock a **local** interactive session after [a authorized administrator-specified time period of inactivity] by: • clearing or overwriting display devices, making the current contents unreadable. • disabling any activity of the user's data access/display devices other than unlocking the session.

FTA_SSL.1.2	The **IT Environment** shall require the following events to occur prior to unlocking the session: [user re-authentication].
Dependency:	*FIA_UAU.1 – Timing of authentication*

FTA_SSL.2 User-initiated locking

FTA_SSL.2.1	The **IT Environment** shall allow user-initiated locking of the user's own **local** interactive session by: • clearing or overwriting display devices, making the current contents unreadable. • disabling any activity of the user's data access/display devices other than unlocking the session.
FTA_SSL.2.2	The **IT Environment** shall require the following events to occur prior to unlocking the session: [user re-authentication].
Dependency:	*FIA_UAU.1 – Timing of authentication*

FTA_SSL.3 IT Environment-initiated termination

FTA_SSL.3.1	The **IT Environment** shall terminate an interactive session after a [authorized administrator-configurable time interval of session inactivity].
Application Note:	*The interactive sessions in FTA_SSL.1, FTA_SSL.2 and FTA_SSL.3 are those of the administrative users. Principals do not have any interactive sessions with the IT environment components.*

FTA_TAB.1 Default TOE access banners

FTA_TAB.1.1	Before establishing a user session, the **IT Environment** shall display an advisory warning message regarding unauthorized use of the TOE.
Application Note:	*The access banner should appear before or in conjunction with the administrative users of the IT environment component being prompted for their user identification and authentication. The intent of this requirement is to advise administrative users of warnings regarding the unauthorized use of the IT environment component and to provide the Security Administrator with control over what is displayed (e.g., if the Security Administrator chooses, they can remove banner information that informs the user of the product and version number).*

5.1.12 Class FTP: Trusted Path/Channels

FTP_ITC.1 Inter-TSF Trusted Channel

FPT_ITC.1.1	The **IT Environment** shall provide a communication channel between **the TOE** and a remote 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.

FPT_ITC.1.2 The **IT Environment** shall permit [selection of one or more by ST author: *the TSF, the remote trusted IT product*] to initiate communication via the trusted channel.

FPT_ITC.1.3 The **IT Environment** shall initiate communications via the trusted channel for [assignment by ST author: *list of functions for which a trusted channel is required*].

Application Note: *This requirement provides confidentiality and integrity services for communications between the TOE and remote IT systems. This includes authentication of administrators, authorized applications, and principals, as well as the subsequent communications with administrators and applications, including remote administration sessions and the query and delivery of privilege attributes or authorization decision functions. The principal's web browsing after authentication is not subject to this requirement.*

The selection in FPT_ITC.1.2 shall always include the remote trusted IT product (e.g. the TLS enabled web browser) and shall include the TSF if required.

TOE Security Assurance Requirements

The TOE assurance requirements for this PP are EAL2 augmented several requirements bolded in the table below. All assurance requirements are summarized in the table below.

Assurance Class	ASSURANCE COMPONENTS	ASSURANCE COMPONENTS DESCRIPTION
DEVELOPMENT	ADV_ARC.1	Architectural Design with domain separation and non-bypassability
	ADV_FSP.2	Security-enforcing Functional Specification
	ADV_TDS.1	Basic design
GUIDANCE DOCUMENTS	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative User guidance
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.2	**Flaw Reporting Procedures**
TESTS	ATE_COV.1	Evidence of coverage
	ATE_FUN.1	Functional testing

Assurance Class	ASSURANCE COMPONENTS	ASSURANCE COMPONENTS DESCRIPTION
	ATE_IND.2	Independent testing - conformance
VULNERABILITY ASSESSMENT	AVA_VAN.2	Vulnerability analysis

Table 15 – Assurance Requirements: EAL2 Augmented

5.1.13 Class ADV: Development

5.1.13.1 ADV_ARC.1 Security architecture description

Dependencies: ADV_FSP.1 Basic functional specification

ADV_TDS.1 Basic design

Developer action elements:

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.

Content and presentation elements:

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 initialization 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.

Evaluator action elements:

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

5.1.13.2 ADV_FSP.2 Security-enforcing functional specification

Dependencies: ADV_TDS.1 Basic design

Developer action elements:

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.

Content and presentation elements:

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 SFR-enforcing TSFIs, 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.

Evaluator action elements:

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.

5.1.13.3 ADV_TDS.1 Basic design

Dependencies: ADV_FSP.2 Security-enforcing functional specification

Developer action elements:

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.

Content and presentation elements:

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 summarize 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 behavior described in the TOE design is mapped to the TSFIs that invoke it.

Evaluator action elements:

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.

5.1.14 Class AGD: Guidance documents

5.1.14.1 AGD_OPE.1 Operational user guidance

Dependencies: ADV_FSP.1 Basic functional specification

Developer action elements:

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

Content and presentation elements:

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 fulfill 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.

Evaluator action elements:

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

5.1.14.2 AGD_PRE.1 Preparative procedures

Dependencies: No dependencies.

Developer action elements:

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

Content and presentation elements:

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.

Evaluator action elements:

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.

5.1.15 Class ALC: Life-cycle support

5.1.15.1 ALC_CMC.2 Use of a CM system

Dependencies: ALC_CMS.1 TOE CM coverage

Developer action elements:

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.

Content and presentation elements:

ALC_CMC.2.1C The TOE shall be labeled 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.

Evaluator action elements:

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

5.1.15.2 ALC_CMS.2Parts of the TOE CM coverage

Dependencies: No dependencies.

Developer action elements:

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

Content and presentation elements:

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.

Evaluator action elements:

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

5.1.15.3 ALC_DEL.1 Delivery procedures

Dependencies: No dependencies.

Developer action elements:

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

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

Content and presentation elements:

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.

Evaluator action elements:

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

5.1.15.4 ALC_FLR.2 Flaw reporting procedures

Dependencies: No dependencies.

Developer action elements:

ALC_FLR.2.1D The developer shall document flaw remediation procedures addressed to TOE developers.

ALC_FLR.2.2D The developer shall establish a procedure for accepting and acting upon all reports of security flaws and requests for corrections to those flaws.

ALC_FLR.2.3D The developer shall provide flaw remediation guidance addressed to TOE users.

Content and presentation elements:

ALC_FLR.2.1C The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

ALC_FLR.2.2C 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.2.3C The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC_FLR.2.4C 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.2.5C The flaw remediation procedures shall describe a means by which the developer receives from TOE users reports and enquiries of suspected security flaws in the TOE.

ALC_FLR.2.6C The procedures for processing reported security flaws shall ensure that any reported flaws are remediated and the remediation procedures issued to TOE users.

ALC_FLR.2.7C The procedures for processing reported security flaws shall provide safeguards that any corrections to these security flaws do not introduce any new flaws.

ALC_FLR.2.8C The flaw remediation guidance shall describe a means by which TOE users report to the developer any suspected security flaws in the TOE.

Evaluator action elements:

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

5.1.16 Class ATE: Tests

5.1.16.1 ATE_COV.1 Evidence of coverage

Dependencies: ADV_FSP.2 Security-enforcing functional specification

ATE_FUN.1 Functional testing

Developer action elements:

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

Content and presentation elements:

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.

Evaluator action elements:

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

5.1.16.2 ATE_FUN.1 Functional testing

Dependencies: ATE_COV.1 Evidence of coverage

Developer action elements:

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

ATE_FUN.1.2D The developer shall provide test documentation.

Content and presentation elements:

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.

Evaluator action elements:

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

5.1.16.3 ATE_IND.2 Independent testing - sample

Dependencies: ADV_FSP.2 Security-enforcing functional specification

AGD_OPE.1 Operational user guidance

AGD_PRE.1 Preparative procedures

ATE_COV.1 Evidence of coverage

ATE_FUN.1 Functional testing

Developer action elements:

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

Content and presentation elements:

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.

Evaluator action elements:

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.

5.1.17 Class AVA: Vulnerability assessment

5.1.17.1 AVA_VAN.2 Vulnerability analysis

Dependencies: ADV_ARC.1 Security architecture description

ADV_FSP.1 Basic functional specification

ADV_TDS.1 Basic design

AGD_OPE.1 Operational user guidance

AGD_PRE.1 Preparative procedures

Developer action elements:

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

Content and presentation elements:

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

Evaluator action elements:

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.

Application Note: The TOE version used as the basis for testing should include a reference to the specific signature set in place when this activity is conducted.

6 RATIONALE

This section describes the rationale for the Security Objectives and Security Functional Requirements as defined in Section 4 and Section 5, respectively. Table 16 illustrates the mapping from Threats, Policies and Assumptions to Security Objectives; and Table 17 illustrates the mapping from Security Objectives to Threats, Policies and Assumptions.

Rationale for Security Objectives

Table 16 - Security Objectives Mapping to Threats/Policies/Assumptions

Threats/Policies/Assumptions	Security Objectives
A.IT_ACCESS	OE.IT_ACCESS
A.LOWEXP	OE.LOWEXP
A.MANAGE	OE.MANAGE
A.NO_EVIL	OE.NO_EVIL
A.NO_TOE_BYPASS	OE.NO_TOE_BYPASS
A.PHYSICAL	OE.PHYSICAL
A.SCALABLE	OE.SCALABLE
P.ACCESS_BANNER	O.DISPLAY_BANNER
	OE.DISPLAY_BANNER
P.ACCOUNTABILITY	O.AUDIT_GENERATION
	OE.AUDIT_PROTECTION
	OE.CAPP_OS
	O.TOE_ACCESS
P.BASIC_ROBUSTNESS	OD.BASIC_ROBUSTNESS
P.CAPP_OS	OE.CAPP_OS
P.COMMS	OE.COMMS
P.CRYPTOGRAPHY	OE.CRYPTOGRAPHY
P.HIGH_AVAILABILITY	OE.PRIORITY
	OE.FAULT_TOLERANCE
P.NO_GENERAL_PURPOSE	OE.NO_GENERAL_ PURPOSE
P.TOE_ENVIRONMENT_ACCESS	OE.TOE_ENVIRONMENT_ACCESS
P.WEB_BROWSER_PP	OE.WEB_BROWSER_PP
T.ACCIDENTAL_ADMIN_ERROR	O.ADMIN_GUIDANCE
T.ACCIDENTAL_CRYPTO_COMPROMISE	OE.RESIDUAL_ INFORMATION
	OE.CRYPTOGRAPHY
T.ACCIDENTAL_AUDIT_COMPROMISE	O.PARTIAL_SELF_PROTECTION
	O.RESIDUAL_ INFORMATION
	OE.CAPP_OS
T.LOW_PRIORITY	O.ADMIN_GUIDANCE
	OE.PRIORITY

T.MASQUERADE	O.TOE_ACCESS
	O.MEDIATE
T.POOR_DESIGN	OD.CONFIGURATION_IDENTIFICATION
	OD.DOCUMENTED_DESIGN
	OD.VULNERABILITY_ANALYSIS
T.POOR_IMPLEMENTATION	OD.CONFIGURATION_IDENTIFICATION
	OD.PARTIAL_FUNCTIONAL_TESTING
	OD.VULNERABILITY_ANALYSIS
T.POOR_TEST	O.CORRECT_ TSF_OPERATION
	OD.PARTIAL_FUNCTIONAL_TESTING
	OD.VULNERABILITY_ANALYSIS
	OD.DOCUMENTED_DESIGN
T.RESIDUAL_DATA	O.RESIDUAL_INFORMATION
T.TSF_COMPROMISE	O.MANAGE
	O.PARTIAL_SELF_PROTECTION
	O.RESIDUAL_INFORMATION
T.UNATTENDED_SESSION	O.TOE_ACCESS
	OE.TOE_ENVIRONMENT_ACCESS
T.UNAUTHORIZED_ACCESS	O.MEDIATE
T.UNIDENTIFIED_ACTIONS	O.AUDIT_GENERATION
	OE.CAPP_OS

A.IT_ACCESS states that the TOE has access to all the IT System data it needs to perform its functions. This assumption is mapped to:

- **OE.IT_ACCESS**, which states that Sites deploying the TOE will ensure the TOE has access to all the IT System data it needs to perform its functions. **OE.IT_ACCESS** directly upholds **A.IT_ACCESS.**

A.LOWEXP states that the threat of malicious attacks aimed at discovering exploitable vulnerabilities is considered low. This assumption is mapped to:

- **OE.LOWEXP,** which states that Site deploying the TOE will establish a protective environment where the threat of malicious attacks aimed at discovering exploitable vulnerabilities is considered low.

A.MANAGE states that there will be one or more competent individuals assigned to manage the TOE and the security of the information it contains. This assumption is mapped to:

- **OE.MANAGE**, which states that the TOE environmental components will provide all the functions, facilities and competent individuals necessary to support the administrators

in their management of the security of the environment, and restrict these functions and facilities from unauthorized use.

A.NO_EVIL states that Administrators are non-hostile, appropriately trained and follow all administrator guidance. This assumption is mapped to:

- **OE.NO_EVIL**, which states that sites using the TOE shall ensure that administrators are non-hostile, appropriately trained and follow all administrator guidance. **OE.NO_EVIL** directly upholds **A.NO_EVIL.**

A.NO_TOE_BYPASS states that Principals cannot gain access to resources protected by the TOE without passing through the TOE access control mechanisms. This assumption is mapped to:

- **OE.NO_TOE_BYPASS**, which states that Principals cannot gain access to resources protected by the TOE without passing through the TOE access control mechanisms. **OE.NO_EVIL** directly upholds **A.NO_EVIL.**

A.PHYSICAL states that Physical security will be provided within the domain for the value of the IT assets protected by the TOE. This assumption is mapped to:

- **OE.PHYSICAL**, which states that Physical security will be provided within the domain for the value of the IT assets protected by the operating system and the value of the stored, processed, and transmitted information. **OE.PHYSICAL** directly upholds **A.PHYSICAL.**

A.SCALABLE states that the TOE environment is appropriately scalable to provide support to the IT Systems in the organization. This assumption is mapped to:

- **OE.SCALABLE**, which states that Sites using the TOE will deploy the appropriate hardware and software environment to ensure the TOE system is scalable to provide support to the IT Systems in the organization it is deployed. **OE.SCALABLE** directly upholds **A.SCALABLE.**

P.ACCESS_BANNER states that the TOE shall display an initial banner describing restrictions of use. This policy is mapped to:

- **O.DISPLAY_BANNER**, which states that the TOE will display an advisory warning regarding use of the TOE to administrators.
- **OE.DISPLAY_BANNER**, which states that the underlying operating system of the TOE will display an advisory warning regarding use of the TOE to administrative users logging on the platform where the TOE software is installed.

P.ACCOUNTABILITY states that the TOE shall log all actions by authorized users. This policy is mapped to:

- **O.AUDIT_GENERATION**, which states that the TOE will provide the capability to detect and create records of security-relevant events associated with users. This addresses this policy by providing the Security Administrator with the capability of configuring the audit mechanism to record the actions of a specific user.
- **OE.AUDIT_PROTECTION**, which states that the IT Environment will provide the capability to protect audit information.
- **O.TOE_ACCESS**, which states that the TOE will provide mechanisms that control a user's logical access to the TOE. This supports this policy by requiring the TOE to identify and authenticate all authorized users prior to allowing any TOE access or access to any TOE protected resource that the TOE is mediating access on behalf of the users.
- **OE.CAPP_OS**, which states that Operating systems the TOE operates on top of must be compliant with the Controlled Access Protection Profile. This plays a role in supporting this policy by requiring the IT environment to provide a reliable time stamp (configured locally by the Security Administrator or via an external NTP server). The audit mechanism is required to include the current date and time in each audit record. All audit records that include the user ID, will also include the date and time that the event occurred.

P.BASIC_ROBUSTNESS states that the TOE must be developed in accordance with the Basic Robustness guidelines. This policy is mapped to:

- **OD. BASIC_ROBUSTNESS**, which directly enforces **P. BASIC_ROBUSTNESS.**

P.CAPP_OS states that the operating system the TOE operates on top of must be evaluated to be compliant with the Controlled Access Protection Profile. This policy is mapped to:

- **OE.CAPP_OS**, which states that operating systems the TOE operates on top of must be compliant with the Controlled Access Protection Profile. **OE.CAPP_OS** directly enforces **P.CAPP_OS.**

P.COMMS states that Adequate communications exist between the TOE components (internally) and between the TOE components and the IT components. This policy is mapped to:

- **OE.COMMS**, which states that Sites deploying the TOE will provide adequate communications exist between the TOE components (internally) and between the TOE components and the IT components. **OE.COMMS** directly enforces **P.COMMS.**

P.CRYPTOGRAPHY states that Only NIST FIPS 140-2 validated cryptographic methods and implementations are acceptable. This policy is mapped to:

- **OE.CRYPTOGRAPHY**, which states that the IT environment components shall use NIST FIPS 140-2 validated cryptographic modules if they provide cryptographic services. These services will provide confidentiality and integrity protection of TSF data while in transit to between software components of the TOE and for TSF data being transfer to/from trusted IT environment components.

P.HIGH_AVAILABILITY states that the TOE shall include providing resource allocations to support priority of service and fault tolerance. This policy is mapped to:

- **OE.PRIORITY**, which states that the IE Environment will provide prioritization of resources to support the TOE. This will ensure that priority of service is available to the TOE.
- **OE.FAULT_TOLERANCE**, which states that the IT environment will provide limited capabilities to support degraded fault tolerance and fail over for some TOE components. This helps satisfy the policy by ensuring that when a single instance of authorization server policy engine fails, operations are continued by an alternate authorization server policy engine.

P.NO_GENERAL_PURPOSE states that there will be no general-purpose computing or storage repository capabilities available on the hardware platforms on which the TOE software is installed. This policy is mapped to:

- **OE.NO_GENERAL_PURPOSE**, which states that there will be no general-purpose computing or storage repository capabilities available on the hardware platforms on which the TOE software is installed. **OE.NO_GENERAL_PURPOSE** directly enforces **P.NO_GENERAL_PURPOSE.**

P.TOE_ENVIRONMENT_ACCESS states that the TOE environment will provide mechanisms that control a user's logical access to the TOE environmental components. This policy is mapped to:

- **OE.TOE_ENVIRONMENT_ACCESS**, which states that the TOE environment will provide mechanisms that control a user's logical access to the environmental components. **OE.TOE_ENVIRONMENT_ACCESS** directly enforces **P.TOE_ENVIRONMENT_ACCESS.**

P. WEB_BROWSER_PP states that if administrators use a web browser to access the TOE for remote administration, they must to use software that has been evaluated to the Web Browser Protection Profile. This policy is mapped to:

- **OE. WEB_BROWSER_PP**, which directly enforces **P. WEB_BROWSER_PP.**

T.ACCIDENTAL_ADMIN_ERROR states that an administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms. This threat is mapped to:

- **O.ADMIN_GUIDANCE**, which states that the TOE will provide administrators with the necessary information for secure management. This helps to mitigate this threat by ensuring the TOE administrators have guidance that instructs them how to administer the TOE in a secure manner and to provide the administrator with instructions to ensure the TOE was not corrupted during the delivery process. Having this guidance helps to reduce the mistakes that an administrator might make that could cause the TOE to be configured in a way that is insecure.

T.ACCIDENTAL_AUDIT_COMPROMISE states that an administrative user or process may view audit records, cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking a user's action. This threat is mapped to:

- **O.PARTIAL_SELF_PROTECTION**, which states that the TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure through its own interfaces. This contributes to countering this threat by ensuring that the TSF can protect itself from users. If the TSF could not maintain security domains of subjects in the TOE Scope of Control, it could not be trusted to control access to the resources under its control, which includes the audit trail.
- **O.RESIDUAL_INFORMATION**, which states that the TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated. This prevents a user not authorized to read the audit trail from access to audit information that might otherwise be persistent in a TOE resource (e.g., memory). By ensuring the TOE prevents residual information in a resource, audit information will not become available to any user or process except those explicitly authorized for that data.
- **OE.CAPP_OS**, which states that Operating systems the TOE operates on top of must be compliant with the Controlled Access Protection Profile. This contributes to mitigating this threat by controlling access to the audit trail. No one is allowed to modify audit records, and only an authorized administrator is allowed to delete the audit trail. The operating system has the capability to prevent auditable actions from occurring if the audit trail is full.

T.ACCIDENTAL_CRYPTO_COMPROMISE states that an administrative user or process may cause the cryptographic functionality to be inappropriately accessed. This threat is mapped to:

- **OE.RESIDUAL_INFORMATION**, which states that the TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated. This mitigates the possibility of malicious users or processes from gaining inappropriate access to cryptographic data, including keys. This objective ensures that the cryptographic data does not reside in a resource that has been used by the cryptographic module and then reallocated to another process.
- **OE.CRYPTOGRAPHY**, which states that the IT environment components shall use NIST FIPS 140-2 validated cryptographic modules if they provide cryptographic

services. This provides assurance that the cryptographic modules do not permit accidental compromise.

T.LOW_PRIORITY states that a low priority process may exhaust resources required by the TOE. This threat is mapped to:

- **O.ADMIN_GUIDANCE**, which states that the TOE will provide administrators with the necessary information for secure management. This will instruct administrators to configure the IT Environment to support prioritization of the TOE's resources.
- **OE.PRIORITY**, which states that the IT Environment will provide prioritization of resources to support the TOE. This mitigates the threat by ensuring that the TOE can have a higher priority than other processes in the Environment.

T.MASQUERADE states that a user or process may masquerade as another entity. This threat is mapped to:

- **O.TOE_ACCESS**, which states that the TOE will provide mechanisms that control a user's logical access to the TOE. This mitigates this threat by controlling the logical access to the TOE and its resources. By identifying and authenticating all users (and principals if the TOE acts as an authentication server) this objective helps mitigate the possibility of a user attempting to login and masquerade as an authorized user or an unauthorized entity accessing a protected resource. In addition, this objective provides the administrator the means to control the number of failed login attempts a user can generate before an account is locked out, further reducing the possibility of a user gaining unauthorized access to the TOE.
- **O.MEDIATE,** which states that the TOE must protect user data in accordance with its security policy. This works to mitigate this threat by constraining how and when authorized users can access the TOE.

T.POOR_DESIGN states that unintentional errors in requirements specification or design of the TOE may occur. This threat is mapped to:

- **OD.CONFIGURATION_IDENTIFICATION**, which states that the configuration of the TOE is fully identified in a manner that will allow implementation errors to be identified, corrected with the TOE being redistributed promptly. This counters this threat by requiring the developer have a configuration item, a reference for each version of the TOE, and a Configuration Management (CM) system with CM documentation. The developer is also required to establish flaw remediation procedures for accepting and acting upon user reports of security flaws and ensuring that any reported flaws are corrected.
- **OD.DOCUMENTED_DESIGN**, which states that the design of the TOE is adequately and accurately documented. This counters this threat, to a degree, by requiring that the TOE be developed using a documented design engineering approach. By providing at least a high level of informal documenting of the security mechanisms in the TOE, the

design of the TOE can be understood, which increases the chances that design errors will be discovered.

- **OD.VULNERABILITY_ANALYSIS**, which states that the TOE will undergo vulnerability analysis demonstrate the design and implementation of the TOE does not contain any obvious flaws. This ensures that the design of the TOE is analyzed by the developer for obvious design flaws. Having the developer perform a vulnerability assessment and document that known vulnerabilities cannot be exploited may find errors in the design that may have been left undiscovered.

T.POOR_IMPLEMENTATION states that unintentional errors in implementation of the TOE design may occur. This threat is mapped to:

- **OD.CONFIGURATION_IDENTIFICATION**, which states that the configuration of the TOE is fully identified in a manner that will allow implementation errors to be identified, corrected with the TOE being redistributed promptly. This contributes to this objective by requiring the developer have a configuration item, a reference for each version of the TOE, and a Configuration Management (CM) system with CM documentation. The developer is also required to establish flaw remediation procedures for accepting and acting upon user reports of security flaws and ensuring that any reported flaws are corrected. Following a good CM process during development will reduce the risk of a implementation errors
- **O. PARTIAL_FUNCTIONAL_TESTING**, which states that the TOE will undergo security functional testing that demonstrates the TSF satisfies some of its security functional requirements. This increases the likelihood that any errors that do exist in the implementation (with respect to the functional specification and high level design) will be discovered through testing.
- **OD.VULNERABILITY_ANALYSIS**, which states that the TOE will undergo vulnerability analysis demonstrate the design and implementation of the TOE does not contain any obvious flaws. This ensures that the design of the TOE is analyzed for obvious design flaws buy the developer. Having the developer perform a vulnerability assessment and document that known vulnerabilities cannot be exploited may find errors in the design that may have been left undiscovered.

T.POOR_TEST states that lack of or insufficient tests to demonstrate that all TOE security functions operate correctly may result in incorrect TOE behavior being undiscovered. This threat is mapped to:

- **OD.DOCUMENTED_DESIGN**, which states that the TOE's design will be adequately and accurately documented. This ensures the existence of design documentation sufficient to permit adequate testing of the TOE.
- **O.CORRECT_TSF_OPERATION**, which states that the TOE will provide the capability to test the TSF to ensure the correct operation of the TSF at a customer's site. This provides administrators with the capability to verify the integrity TSF data, including stored TSF executable code and configuration files.

- **OD.PARTIAL_FUNCTIONAL_TESTING**, which states that the TOE will undergo security functional testing that demonstrates the TSF satisfies its security functional requirements. This ensures that functional testing is performed to ensure the TSF satisfies the security functional requirements and demonstrates that the TOE's security mechanisms operate as documented. While functional testing serves an important purpose, it does not ensure the TSFI cannot be used in unintended ways to circumvent the TOE's security policies.
- **OD.VULNERABILITY_ANALYSIS**, which states that the TOE will undergo vulnerability analysis demonstrate the design and implementation of the TOE does not contain any obvious flaws. This ensures that the design of the TOE is analyzed by the developer for obvious design flaws. Having the developer perform a vulnerability assessment and document that known vulnerabilities cannot be exploited may find errors in the design that may have been left undiscovered.

T.RESIDUAL_DATA states that a user or process may gain unauthorized access to data through reallocation of TOE resources from one user or process to another. This threat is mapped to:

- **O.RESIDUAL_INFORMATION**, which states that the TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated. This counters this threat by ensuring that TSF data and user data is not persistent when resources are released by one user/process and allocated to another user/process. This means that network packets will not have residual data from another packet due to the padding of a packet. This ensures successful access control decisions make for one user do not carry over to the next user.

T.TSF_COMPROMISE states that an attacking user or process may cause, through an unsophisticated attack, TSF data, or executable code to be inappropriately accessed. This threat is mapped to:

- **O.MANAGE**, which states that the TOE will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use. This defines an access control policy to control access to TSF data or the resources being protected by the TOE. This objective is used to dictate who is able to view and modify TSF data, as well as the behavior of TSF functions.
- **O.PARTIAL_SELF_PROTECTION**, which states that the TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure through its own interfaces. This contributes to countering this threat by ensuring that the TSF can protect itself from users. If the TSF could not maintain security domains of subjects in the TOE Scope of Control, it could not be trusted to control access to the resources under its control. It requires that the TSF be able to protect itself from tampering and that the security mechanisms in the TSF cannot be bypassed.

- **O.RESIDUAL_INFORMATION**, which states that the TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated. This counters this threat by ensuring that TSF data and user data is not persistent when resources are released by one user/process and allocated to another user/process. This means that network packets will not have residual data from another packet due to the padding of a packet. This ensures successful access control decisions make for one user do not carry over to the next user.

T.UNATTENDED_SESSION states that a user may gain unauthorized access to an unattended session. This threat is mapped to:

- **O.TOE_ACCESS**, which states that the TOE will provide mechanisms that control a user's logical access to the TOE, including the locking of sessions.
- **OE.TOE_ENVIRONMENT_ACCESS**, which states that the TOE environment will provide mechanisms that control a user's logical access to the environmental components. This helps to mitigate this threat by including mechanisms that place controls on user's sessions. Local administrator's sessions are locked and remote sessions are dropped after a Security Administrator defined time period of inactivity. Locking the local administrator's session reduces the opportunity of someone gaining unauthorized access the session when the console is unattended. Dropping the connection of a remote session (after the specified time period) reduces the risk of someone accessing the remote machine where the session was established, thus gaining unauthorized access to the session.

T.UNAUTHORIZED_ACCESS states that a user may gain access to the data for which they are not authorized according to the TOE security policy. This threat is mapped to:

- **O.MEDIATE**, which states that the TOE must protect user data in accordance with its security policy. This works to mitigate this threat by ensuring that all requests to access user data, or data being protected by the TOE, are subject to an Authorization Server access control policy. A TOE policy engine enforces rules to determine if an operation among controlled subjects and controlled objects is allowed based on the security attributes of the user and the object. The TOE requires successful authentication to the TOE prior to gaining access to administrative services on or mediated by the TOE to protected resources. Communications between the TOE components must be protected from unauthorized disclosure to ensure integrity and confidentiality of the user data. Lastly, the TSF must ensure that all configured enforcement functions (authentication, access control rules, etc.) must be invoked prior to allowing a user to gain access to TOE or TOE mediated services. The TOE restricts the ability to modify the security attributes associated with access control rules, access to authenticated and unauthenticated services, etc to the Security Administrator. This feature ensures that no other user can modify the access control policy to bypass the intended TOE security policy.

T.UNIDENTIFIED_ACTIONS states that the administrator may not have the ability to notice potential security violations. This threat is mapped to:

- **O.AUDIT_GENERATION**, which states that the TOE will provide the capability to detect and create records of security-relevant events associated with users. This means that actions that might result from security violations will be audited, and thus may be detected by administrators.
- **OE.CAPP_OS**, which states that operating systems in which the TOE operates must be compliant with the Controlled Access Protection Profile. This helps to mitigate this threat by providing the Security Administrator with a set of rules for monitoring the audited events and based upon these rules can indicate a potential violation of the TSP. A required minimum set of configurable audit events that could indicate a potential security violation. By configuring these auditable events, when the Security or Audit Administrator reviews the audit records, they can determine the occurrences of these events (e.g. set number of authentication failures, etc.). A search and sort capability provides an efficient mechanism for the Audit Administrator to view pertinent audit information.

Table 17 Security Objectives Mapping to Threats/Policies/Assumptions

Security Objectives	Threats/Policies/Assumptions
O.ADMIN_GUIDANCE	T.ACCIDENTAL_ADMIN_ERROR
	T.LOW_PRIORITY
O.AUDIT_GENERATION	P.ACCOUNTABILITY
	T.UNIDENTIFIED_ACTIONS
OD.CONFIGURATION_IDENTIFICATION	T.POOR_DESIGN
	T.POOR_IMPLEMENTATION
O.CORRECT_TSF_OPERATION	T.POOR_TEST
O.DISPLAY_BANNER	P.ACCESS_BANNER
OD.DOCUMENTED_DESIGN	T.POOR_DESIGN
	T.POOR_TEST
O.MANAGE	T.TSF_COMPROMISE
O.MEDIATE	T.UNAUTHORIZED_ACCESS
	T.MASQUERADE
OD.PARTIAL_FUNCTIONAL_TESTING	T.POOR_IMPLEMENTATION
	T.POOR_TEST
O.PARTIAL_SELF_PROTECTION	T.ACCIDENTAL_AUDIT_COMPROMISE
	T.TSF_COMPROMISE
O.RESIDUAL_INFORMATION	T.RESIDUAL_DATA
	T.ACCIDENTAL_ CRYPTO_COMPROMISE
	T.ACCIDENTAL_AUDIT_COMPROMISE
	T.TSF_COMPROMISE
O.TOE_ACCESS	P.ACCOUNTABILITY

	T.MASQUERADE
	T.UNATTENDED_SESSION
OD.VULNERABILITY_ANALYSIS	T.POOR_DESIGN
	T.POOR_IMPLEMENTATION
	T.POOR_TEST
OE.AUDIT_PROTECTION	P.CAPP_OS
	P.ACCOUNTABILITY
OE.CAPP_OS	P.CAPP_OS
	P.ACCOUNTABILITY
	T.ACCIDENTAL_AUDIT_COMPROMISE
	T.UNIDENTIFIED_ACTIONS
OE.COMMS	P.COMMS
OE.CRYPTOGRAPHY	P.CRYPTOGRAPHY
	T.ACCIDENTAL_CRYPTO_COMPROMISE
OE.DISPLAY_BANNER	P.ACCESS_BANNER
OE.FAULT_TOLERANCE	P.HIGH_AVAILABILITY
OE.IT_ACCESS	A.IT_ACCESS
OE.LOWEXP	A.LOWEXP
OE.MANAGE	A.MANAGE
OE.NO_EVIL	A.NO_EVIL
OE.NO_GENERAL_PURPOSE	P.NO_GENERAL_ PURPOSE
OE.NO_TOE_BYPASS	A.NO_TOE_BYPASS
OE.PHYSICAL	A.PHYSICAL
OE.PRIORITY	T.LOW_PRIORITY
	P.HIGH_AVAILABILITY
OE.RESIDUAL_INFORMATION	T.ACCIDENTAL_CRYPTO_COMPROMISE
OE.SCALABLE	A.SCALABLE
OE.TOE_ENVIRONMENT_ACCESS	P.TOE_ENVIRONMENT_ACCESS
	T.UNATTENDED_ SESSION

Rationale for TOE Security Requirements

Table 18 – Security Objectives Mapping to Security Requirements

Security Objectives	Security Requirements
O.ADMIN_GUIDANCE	ALC_DEL.1
	AGD_PRE.1
	AGD_OPE.1

O.AUDIT_GENERATION	FAU_GEN.1
	FAU_GEN.2
OD.CONFIGURATION_IDENTIFICATION	ALC_CMC.2
	ALC_CMS.2
	ALC_FLR.2
O.CORRECT_TSF_OPERATION	FPT_TST_(EXT)1.1, FPT_TST_(EXT)2.1,
O.DISPLAY_BANNER	FTA_TAB.1
OD.DOCUMENTED_DESIGN	ADV_FSP.2
	ADV_TDS.1
	ADV_ARC.1
O.MANAGE	FMT_MTD.1
	FMT_MSA.1(1)
	FMT_MSA.2
	FMT_MSA.3
	FMT_MOF.1
	FMT_SMF.1
	FMT_SMR.1
O.MEDIATE	FDP_ACC.1
	FDP_ACF_(EXT).1
	FIA_ATD.1(2)
	FIA_ATD.1(3)
	FMT_MSA.1(2)
OD.PARTIAL_FUNCTIONAL_TESTING	ATE_COV.1
	ATE_FUN.1

	ATE_IND.2
O.PARTIAL_SELF_PROTECTION	ADV_ARC.1
O.RESIDUAL_INFORMATION	FDP_RIP.2
	FCS_CKM.4
O.TOE_ACCESS	FIA_AFL.1
	FIA_ATD.1(1)
	FIA_SOS.1
	FIA_UID.2
	FIA_UAU.2
OD.VULNERABILITY_ANALYSIS	AVA_VAN.2
OE.CRYPTOGRAPHY	FCS_CKM.1, FCS_CKM.2, FCS_CKM.3, FCS_COP.1
OE.DISPLAY_BANNER	FTA_TAB.1
OE.FAULT_TOLERANCE	FRU_FLT.1, FRU_RSA.1
OE.PRIORITY	FRU_PRS.2
OE.TOE_ENVIRONMENT_ACCESS	FPT_ITT.1, FTA_SSL.1, FTA_SSL.2, FTA_SSL.3, FTP_ITC.1

O.ADMIN_GUIDANCE states that the TOE will provide administrators with the necessary information for secure management. This security objective is met by:

- ADO_DEL.1, which ensures that the administrator is provided documentation that describe all procedures that are necessary to maintain security when receiving and distributing versions of TOE software at a user's site.
- AGD_PRE.1which ensures the administrator has the information necessary to install the TOE in the evaluated configuration. Often times a vendor's product contains software that is not part of the TOE and has not been evaluated. The Installation, Generation and Startup (IGS) documentation ensures that once the administrator has followed the installation and configuration guidance the result is a TOE in a secure configuration.
- AGD_OPE.1which mandates the developer provides the administrator with guidance on how to operate the TOE in a secure manner. This includes describing the interfaces the administrator uses in managing the TOE, security parameters that are configurable by the

administrator, and how to configure the TOE's access control rulesets for protecting web servers. The documentation also provides a description of how to setup and review the auditing features of the TOE. AGD_OPE.1 would also be used to provide guidance on security that is common to both administrators and non-administrators (e.g., password management guidelines). Since the non-administrative users of this TOE only interact with the TOE via web agent, it is expected that the guidance would discuss the secure access to protected web servers and how the authentication mechanism on the web server is used to pass the user's request for access to web resources to the TOE.

O.AUDIT_GENERATION states that the TOE will provide the capability to detect and create records of security-relevant events associated with users. This security objective is met by:

- FAU_GEN.1, which defines the set of events that the TOE must be capable of recording. This requirement ensures that the Security Administrator has the ability to audit any security relevant event that takes place in the TOE. This requirement also defines the information that must be contained in the audit record for each auditable event. There is a minimum of information that must be present in every audit record and this requirement defines that, as well as the additional information that must be recorded for each auditable event. This requirement also places a requirement on the level of detail that is recorded on any additional security functional requirements an ST author adds to this PP.
- FAU_GEN.2, which ensures that the audit records associate a user identity with the auditable event. In the case of authorized users, the association is accomplished with the "userid". When TOE components imitate actions that need to be audited, the TOE will ensure a mechanism is in place to identity the component as the entity conducting the action.

OD.CONFIGURATION_IDENTIFICATION states that the configuration of the TOE is fully identified in a manner that will allow implementation errors to be identified, corrected with the TOE being redistributed promptly. This security objective is met by:

- ALC_CMC.2 and ALC_CMS.2 contribute to this objective by requiring the developer provide a reference for the TOE and use a configuration management system CM The developer shall also documentation including a configuration list that describes the configuration items that comprise the TOE. This documentation will describe the method used to uniquely identify the configuration items.
- ALC_FLR.2, which plays a role in satisfying the "analyzed" portion of this objective by requiring the developer to have procedures that address flaws that have been discovered in the product, either through developer actions (e.g., developer testing) or those discovered by others. The flaw remediation process used by the developer corrects any discovered flaws and performs an analysis to ensure new flaws are not created while fixing the discovered flaws.

O.CORRECT_TSF_OPERATION states that the TOE will provide the capability to test the TSF to ensure the correct operation of the TSF at a customer's site. This security objective is met by:

- FPT_TST_(EXT)1.1 and FPT_TST_(EXT)2.1, which ensure the correctness of the TSF configuration files, data and executable code. If TSF software is corrupted it is possible that the TSF would no longer be able to enforce the security policies. This also holds true for TSF data, if TSF data is corrupt the TOE may not correctly enforce its security policies. The FPT_TST_(EXT)1 functional requirement includes the critical nature and specific handling of the cryptographic related TSF data. Since the cryptographic TSF data has specific FIPS PUB requirements associated with them it is important to ensure that any fielded testing on the integrity of these data maintains the same level of scrutiny as specified in the FCS functional requirements.

O.DISPLAY_BANNER states that the TOE will display an advisory warning regarding use of the TOE to administrators. This security objective is met by:

- FTA_TAB.1, which meets this objective by requiring the TOE display a Security Administrator defined banner before an administrator can establish an authenticated remote session. This banner is under complete control of the Security Administrator in which they specify any warnings regarding unauthorized use of the TOE and remove any product or version information if they desire.

OD.DOCUMENTED_DESIGN states that the design of the TOE is adequately and accurately documented. This security objective is met by:

- ADV_FSP.2, which ensures the developer documents the TOE with a functional specification that clearly describes the TSF, including the purpose and method of use of all external TSF interfaces.
- ADV_TDS.1, which requires the developer to provide the design of the TSF. Although the presentation of the design can be informal, it must be internally consistent. The design will also include a description of the security functionality provided by each subsystem of the TSF. Having accurate design documentation is imperative for evaluator's to gain an appropriate level of understanding of the TOE's security operations in a reasonable amount of time.
- ADV_ARC.1which provides the architecture of the TSF and an easy to analyze model for security policy.

O.MANAGE states that the TOE will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use. This security objective is met by:

- FMT_MOF.1(1) and FMT_MOF.1(2), which provide Security Administrators the ability to manage the TOE's access policy settings and the list of applications authorized to query the TOE.
- FMT_MOF.1(3), which provides the Audit Administrator the ability to manage the audit settings.
- FMT_MSA.1(1), which provides the Security Administrator with the capability to manage the security attributes of both principals and protected resources.
- FMT_MSA.2 ensures that only specific secure values are accepted for security attributes. This requirement is designed meet the DCID requirement to prevent user authentication password reuse. A history of static authenticator changes will be maintained with assurance of non-replication of individual authenticators. When a user changing their password submits a previously used password, the system will consider that an "insecure" value for that security attribute and reject it.
- FMT_MSA.3 requires that by default, the TOE does not allow an access to a protected resource until an access policy rule allows it.
- FMT_MTD.1 is used by the Security Administrator to manage TSF data and configuration.
- FMT_SMF.1 requires that the TSF shall be capable of performing specified security management functions.
- FMT_SMR.1 requires that roles exist for administrative actions: the Security Administrator, who is responsible for configuring the TOE's security policies, including the management of the security data that is critical to the cryptographic operations; the Audit Administrator, who is restricted to reading and deleting the audit trail; and Authorized Applications which are permitted to query the TOE. The TSF is able to associate a human user with one or more roles.

O.MEDIATE states that the TOE must protect user data in accordance with its security policy. This security objective is met by:

- FDP_ACC.1 defines that an Authorization Server Access Control policy will be enforced on principals attempting to gain access to a list of named objects. All the operations among subject and object covered are by the Authorization Server policy. The "subjects" are generally the Authorization Server "Agents." The "named objects" are the designated web based resources (web server, directories, files, or objects) that the Authorization Server is protecting.
- FDP_ACF_(EXT).1 defines the Security Attribute used to provide Access Control to objects based on the following Authorization Server Access Control policy.
- FIA_ATD.1(2) and FIA_ATD.1(3) define the Security Attributes associated with the principals and authorized applications.
- FMT_MSA.1(2) restricts disclose of user security attributes to authorized applications.

OD.PARTIAL_FUNCTIONAL_TESTING states that the TOE will undergo security functional testing that demonstrates the TSF satisfies its security functional requirements. This security objective is met by:

- ATE_FUN.1 requires the developer to provide the necessary test documentation to allow for an independent analysis of the developer's security functional test coverage
- ATE_COV.2 requires the developer to provide a test coverage analysis that demonstrates the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification. This component also requires an independent confirmation of the completeness of the test suite, which aids in ensuring that correct security relevant functionality of a TSFI is demonstrated through the testing effort.
- ATE_IND.2 requires an independent confirmation of the developer's test results, by mandating a subset of the test suite be run by an independent party. Upon successful adherence to these requirements, the TOE's conformance to the specified security functional requirements will have been demonstrated.

O.PARTIAL_SELF_PROTECTION states that the TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure through its own interfaces. This security objective is met by:

- ADV_ARC.1 is used to define the architecture that will enable the domain separation between the security domains of subjects in the TOE Scope of Control. In addition, the architecture ADV_ARC.1 will also ensure that the IT environment provides domain separation in order to protect the TOE. The architecture will also ensures that the TSF makes policy decisions on all interfaces that perform operations on subjects and objects that are scoped by the policies. Without this non-bypassability requirement, the TSF could not be relied upon to completely enforce the security policies, since an interface(s) may otherwise exist that would provide a user with access to TOE resources (including TSF data and executable code) regardless of the defined policies. This includes controlling the accessibility to interfaces, as well as what access control is provided within the interfaces.
- FPT_FLS.1 will ensure that when the TOE fails, the security is not compromised.
- FPT_RCV. 1 will ensure that the TOE recovers to a correct state after failure
- FPT_ITC.1 is used to protect the TOE data during communication among different parts of the TOE.

O.RESIDUAL_INFORMATION states that the TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated. This security objective is met by:

- FDP_RIP.2 is used to ensure the contents of resources are not available to subjects other than those explicitly granted access to the data. For this TOE it is critical that the memory used to make authorization decisions is either cleared or that some buffer management scheme be employed to prevent the authorization decision of one user's request to be used in a subsequent authorization decision.
- FCS_CKM.4 ensures that the sensitive keys are zeroized when no longer needed so that they can not be misused (e.g., for compromise back traffic, masquerading, etc.).

O.TOE_ACCESS states that the TOE will provide mechanisms that control a user's logical access to the TOE. This security objective is met by:

- FIA_AFL.1 provides a detection mechanism for unsuccessful authentication attempts by remote administrators. The requirement enables a Security Administrator settable threshold that prevents unauthorized users from gaining access to authorized administrator's account by guessing authentication data by locking the targeted account until the Security Administrator takes some action (e.g., re-enables the account) or for some Security Administrator defined time period. Thus, limiting an unauthorized user's ability to gain unauthorized access to the TOE.
- FIA_ATD.1 defines the attributes for administrators, principals, and authorized applications that shall be used to determine identity and enforce what type of access each entity has to the TOE or to another protected resource based on the access control policy.
- FIA_SOS.1.1 ensures that a mechanism is in place to verify that user's passwords must contain a minimum of 8 alphanumeric charters with at least one numeric charter. This type of password cannot be easily be broken with a dictionary search or elementary password cracking software.
- FIA_UAU.2 contributes to this objective by preventing services from being provided by the TOE to unauthenticated users.
- FIA_UID.2 contributes to this objective by preventing services from being provided by the TOE to unidentified users.

OD.VULNERABILITY_ANALYSIS states that the TOE will undergo vulnerability analysis demonstrate the design and implementation of the TOE does not contain any obvious flaws. This security objective is met by:

- AVA_VAN.2 component to provide the basic level of confidence that vulnerabilities do not exist in the TOE that could cause the security policies to be violated. AVA_VAN.2 requires the developer to perform a search for obvious ways in which a user can violate the TSP. The developer will document the disposition of obvious vulnerabilities. For those vulnerabilities that are not eliminated the developer will show that the vulnerability cannot be exploited in the intended environment for the TOE. This component provides the confidence that security flaws do not exist in the TOE that could be exploited by a threat agent of low attack potential to violate the TOE's security policies.

OE.CRYPTOGRAPHY states that the IT environment components shall use NIST FIPS 140-2 validated cryptographic modules if they provide cryptographic services. This security objective is met by:

- FCS_CKM.1, FCS_CKM.2, FCS_CKM.4, and FCS_COP.1, which specify the acceptable cryptographic algorithms and their modes and key sizes as appropriate, and indicates that FIPS 140-2 validated cryptographic modules shall be used.

OE.DISPLAY_BANNER states that the underlying operating system of the TOE will display an advisory warning regarding use of the TOE to administrative users logging on the platform where the TOE software is installed. This security objective is met by:

- FTA_TAB.1, which meets this objective by requiring the underlying operating system display a Security Administrator defined banner before an administrator can establish an authenticated session. This banner is under complete control of the Security Administrator in which they specify any warnings regarding unauthorized use of the TOE and remove any product or version information if they desire.

OE.FAULT_TOLERANCE states that the IT Environment will provide limited capabilities to support degraded fault tolerance and fail over for some TOE components. This security objective is met by:

- FRU_FLT.1, which ensures that authorization decision operations can continue to be provided when a single instance of authorization server policy engine fails and mitigates electrical power disruptions. An automated fail over mechanism should be provided within the TOE to allow for an alternate authorization server policy engine to make decisions.
- FRU_RSA.1, which protects against a class of resource exhaustion network attacks.

OE.PRIORITY states that the IE Environment will provide prioritization of resources to support the TOE. This security objective is met by:

- FRU_PRS.2 which states that the IT environment shall assign a priority to each subject in the IT environment security function, and ensures that each access to all shareable resources shall be mediated on the basis of the subject's assigned priority.

OE.TOE_ENVIRONMENT_ACCESS states that the IE Environment will provide mechanisms that control a user's logical access to the environmental components. This security objective is met by:

- **FTA_SSL.1**, **FTA_SSL.2**, and **FTA_SSL.3**, which together state that either the IT Environment or a user can lock a session, and after a sufficient duration the session will be terminated. These ensure that the TOE can not be accessed when the user is not absent for periods of time.
- **FPT_ITT.1**, which states that the IT Environment shall protect TSF data from disclosure and modification when it is transmitted between separate parts of the TOE, and **FTP_ITC.1**, which states that the IT Environment shall provide a communications channel between the TOE and remote trusted IT products. Together these components protect the authentication data in transit and from masquerading threats.

Table 19 - Security Requirements Mapped to Security Objectives

Security Functional Requirement	Security Objectives
FAU_GEN.1	O.AUDIT_GENERATION
FAU_GEN.2	O.AUDIT_GENERATION
FCS_CKM.1	OE.CRYPTOGRAPHY
FCS_CKM.2	OE.CRYPTOGRAPHY
FCS_CKM.4	OE.RESIDUAL_INFORMATION, OE.CRYPTOGRAPHY
FCS_COP.1	OE.CRYPTOGRAPHY
FDP_ACC.1	O.MEDIATE
FDP_ACF_(EXT).1	O.MEDIATE
FDP_RIP.2	O.RESIDUAL_INFORMATION
FIA_AFL.1	O.TOE_ACCESS
FIA_ATD.1	O.TOE_ACCESS
FIA_SOS.1	O.TOE_ACCESS
FIA_UAU.2	O.TOE_ACCESS
FIA_UID.2	O.TOE_ACCESS
FMT_MOF.1	O.MANAGE
FMT_MSA.1	O.MANAGE
FMT_MSA.2	O.MANAGE
FMT_MSA.3	O.MANAGE
FMT_MTD.1	O.MANAGE
FMT_SMF.1	O.MANAGE
FMT_SMR.1	O.MANAGE
FPT_FLS.1	O.PARTIAL_SELF_PROTECTION
FPT_ITC.1	O.PARTIAL_SELF_PROTECTION
FPT_RCV. 1	O.PARTIAL_SELF_PROTECTION
ADV_ARC.1	O.PARTIAL_SELF_PROTECTION
FPT_TST_(EXT)1.1	O.CORRECT_TSF_OPERATION
FPT_TST_(EXT)2.1	O.CORRECT_TSF_OPERATION
FRU_FLS.2	OE.PRIORITY
FRU_FLT.1	OE.FAULT_TOLERANCE
FRU_RSA.1	OE.FAULT_TOLERANCE
FTA_SSL.1	O.TOE_ACCESS
FTA_SSL.2	O.TOE_ACCESS
FTA_SSL.3	O.TOE_ACCESS
FTA_TAB.1	O.DISPLAY_BANNER
FTA_TAB.1	OE.DISPLAY_BANNER
FTP_ITC.1	O.TOE_ACCESS

Rationale for Assurance Requirements

EAL2 augmented was chosen to ensure an adequate level of confidence in security services used to protect information in Basic Robustness Environments. The assurance selection was based on the postulated low threat environment.

The EAL definitions in Part 3 of the CC were reviewed and the Basic Robustness Assurance Package (EAL2 augmented with assurance requirement ALC_FLR.2) was believed to best achieve this goal. The sponsor concluded that EAL2 augmented is applicable since this PP addresses circumstances where developers and users require a basic to moderate level of independently assured security in commercial products. The addition of assurance requirement AVA_VLA.1 ensures that the developer vulnerability analysis is done to demonstrate the resistance to penetration attackers with low attack potential and that a systematic approach is used to search for obvious vulnerabilities. This collection of assurance requirements require TOE developers to gain assurance from good software engineering development practices which, though rigorous, do not require substantial specialist knowledge, skills, and other resources.

The postulated threat environment specified in Section 3 of this PP was used in conjunction with the Information Assurance Technical Framework (IATF) Robustness Strategy guidance to derive the chosen assurance level.

These factors were taken into consideration and the conclusion was that the basic robustness assurance package was the appropriate level of assurance.

Rational for Satisfying all Dependencies

The Authorization Server Protection Profile satisfies all the requirement dependencies of the Common Criteria. Table 20 below lists each requirement from the Authorization Server Protection Profile with a dependency and indicates whether the dependent requirement was included. As the table indicates, all dependencies have been met.

Table 20 – Requirement Dependencies

Functional Component	Dependencies	Included
FAU_GEN.1	FPT_STM.1	Yes, in CAPP
FAU_GEN.2	FAU_GEN.1	Yes by FAU_GEN.1
	FIA_UID.1	Yes (by FIA_UID.2)
FCS_CKM.1	FCS_CKM.2 , FCS_CKM.4, FMT_MSA.2	Yes
FCS_CKM.2	FCS_CKM.1, FCS_CKM.4, FMT_MSA.2	Yes
FCS_CKM.4	FCS_CKM.1, FMT_MSA.2	Yes
FCS_COP.1	FCS_CKM.1,FCS_CKM.4, FMT_MSA.2	Yes

FDP_ACC.1	FDP_ACF.1	Yes (by FDP_ACF_(EXT).1)
FDP_ACF_(EXT).1	FDP_ACC.1, FMT_MSA.3	Yes
FDP_RIP.2	None	
FIA_AFL.1	FIA_UAU.1	Yes (by FIA_UAU.2)
FIA_ATD.1	None	
FIA_SOS.1	None	
FIA_UAU.2	FIA_UID.1	Yes (by FIA_UID.2)
FIA_UID.2	None	
FMT_MOF.1	FMT_SMF.1, FMT_SMR.1	Yes
FMT_MSA.1	FDP_ACC.1, FMT_SMF.1, FMT_SMR.1	Yes
FMT_MSA.2	ADV_SPM.1, FDP_ACC.1 , FMT_MSA.1, FMT_SMR.1	Yes
FMT_MSA.3	FMT_MSA.1, FMT_SMR.1	Yes
FMT_MTD.1	FMT_SMF.1, FMT_SMR.1	Yes
FMT_SMR.1	FIA_UID.1	Yes
FPT_FLS.1	ADV_SPM.1	Yes
FTP_ITT.1	None	
FPT_RCV.1	AGD_ADM.1, ADV_SPM.1	Yes
FPT_TST.EXP1.1	None	
FPT_TST.EXP2.1	FPT_AMT.1	Yes by CAPP
FRU_FLT.1	FPT_FLS.1	Yes
FRU_PRS.2	None	
FRU_RSA.1	None	
FTA_SSL.1	FIA_UAU.1	Yes (by FIA_UAU.2)
FTA_SSL.2	FIA_UAU.1	Yes (by FIA_UAU.2)
FTA_SSL.3	None	
FTA_TAB.1	None	
FTP_ITC.1	None	

Rationale for Basic Robustness Requirements

Table 21 shows that the requirements of the Basic Robustness guidance have been met.

Table 21 - Basic Robustness Rationale

Requirement	Present?	Required?	Rationale?
FAU_GEN.1-NIAP-0407	Y	should	CC 2.2 incorporates NIAP-0407.
FAU_SEL.1-NIAP-0407	Y (in CAPP)	Y	FAU_SEL.1 is in CAPP, although the NIAP-0407 interpretation and the BRPPG refinement are missing.
FAU_STG.1-NIAP-0429	Y (in CAPP)	should	FAU_STG.1 is in CAPP, with selections that match NIAP-0249, and that exceed BRPPG.
FAU_STG.3	Y (in CAPP)	should	FAU_STG.3 is in CAPP, "generate an alarm to the authorized administrator", vs BRPPG "immediately alert the administrators by displaying a message at the local console". Also, CAPP does not require the alarm threshold to be administrator configurable (but does permit it).
FAU_STG.NIAP-0414	N	should	BRPPG recommends administrator be able to

			specify audit trail loss functionality, but CAPP does not mandate that it is configurable.
FCS_CKM	Y	may	
FCP_COP	Y	may	
FDP_ACF.1-NIAP-0407	Y	if used	
FDP_IFF.1-NIAP-0407	N	if used	These requirements are not mandatory, they are present in the BRPPG only to provide the interpreted requirements text.
FDP_IFF.2-NIAP-0407	N	if used	These requirements are not mandatory, they are present in the BRPPG only to provide the interpreted requirements text.
FIA_AFL.1-NIAP-0425	Y	should	CC 2.2 incorporates NIAP-0425
FIA_USB.1-NIAP-0415	N	suggests	The concept of subjects is not applicable to the operations of the TOE itself. The underlying CAPP OS meets FIA_USB.1.
FPT_TST_(EXT)1.1	Y	should	

A few Basic Robustness requirements in the area of audit (FAU_SEL and FAU_STG) are met by CAPP, but CAPP has slightly different language than the Basic Robustness guidelines. The differences are slight, however, and it is deemed that CAPP language satisfies the spirit of Basic Robustness.

Rationale for Extended requirements

Table 22 presents the rationale for the inclusion of the extended requirements found in this PP.

Table 22 – Rational for Extended Requirements

Extended Requirement	Identifier	Rationale
FDP_ACF_(EXT).1	Security Attribute Based Access Control	This requirement was made extended in order to give the ST Author the flexibility to handle cases in which the access control decision is enforced by the TOE, or where it is simple provided by the TOE through an API.
FPT_TST.EXP1.1	TSF Testing	This extended requirement is necessary since the CC does not specifically provide for “software only” PP to test its features.
FPT_TST.EXP2.1	TSF Testing	This extended requirement is necessary to permit the IT environment to test features in its scope of control.

7 REFERENCES

1) Common Criteria for Information Technology Security Evaluation, *CCIB-98-031 Version 2.2, January 2004.*

[1a] Common Criteria Implementation Board, Common Criteria for Information Technology Security Evaluation, CCMB-2006-09, Version 3.1, September 2006

2) Information Assurance Technical Framework, *Version 3.1, September 2002.*

3) Federal Information Processing Standard Publication (FIPS-PUB) 46-3, Data Encryption Standard (DES), October 1999.

4) Federal Information Processing Standard Publication (FIPS-PUB) 140-2, Security Requirements for Cryptographic Modules, May 25, 2001.

5) Internet Engineering Task Force, IP Encapsulating Security Payload (ESP), RFC 2406, November 1998.

6) Internet Engineering Task Force, Internet Key Exchange (IKE), RFC 2409, November 1998.

7) Internet Engineering Task Force, ESP CBC-Mode Cipher Algorithms, RFC 2451, November 1998.

8) Internet Engineering Task Force, Use of HMAC-SHA-1-96 within ESP and AH, RFC 2404, November 1998.

9) Department of Defense Instruction 8500.2, "Information Assurance," February 6, 2003

10) The AES Cipher Algorithm and Its Use with IPsec (*draft RFC 3602)* http://www.tools.ietf.org/html/rfc3602 ,.

11) Federal Information Processing Standard Publication (FIPS-PUB) 197, Specification for the Advanced Encryption Standard (AES), November 26, 2001

8 TERMINOLOGY

In the Common Criteria, many terms are defined in Section 2.3 of Part 1. The following are a definitions of terms used in this PP and common to other DoD PPs.

Access -- Interaction between an entity and an object that results in the flow or modification of data.

Access Control -- Security service that controls the use of resources[1] and the disclosure and modification of data.[2]

Accountability -- Property that allows activities in an IT system to be traced to the entity responsible for the activity.

Administrator -- A user who has been specifically granted the authority to manage some portion or all of the TOE and whose actions may affect the TSP. Administrators may possess special privileges that provide capabilities to override portions of the TSP.

Assurance -- A measure of confidence that the security features of an IT system are sufficient to enforce its' security policy.

Asymmetric Cryptographic System -- A system involving two related transformations; one determined by a public key (the public transformation), and another determined by a private key (the private transformation) with the property that it is computationally infeasible to determine the private transformation (or the private key) from knowledge of the public transformation (and the public key).

Asymmetric Key -- The corresponding public/private key pair needed to determine the behavior of the public/private transformations that comprise an asymmetric cryptographic system.

Attack -- An intentional act attempting to violate the security policy of an IT system.

Authentication -- Security measure that verifies a claimed identity.

Authentication data -- Information used to verify a claimed identity.

---

[1] Hardware and software.

[2] Stored or communicated.

Authorization -- Permission, granted by an entity authorized to do so, to perform functions and access data.

Authorized user -- An authenticated user who may, in accordance with the TSP, perform an operation.

Availability -- Timely[3], reliable access to IT resources.

Compromise -- Violation of a security policy.

Confidentiality -- A security policy pertaining to disclosure of data.

Critical Security Parameters (CSP) -- Security-related information (e.g., cryptographic keys, authentication data such as passwords and pins, and cryptographic seeds) appearing in plaintext or otherwise unprotected form and whose disclosure or modification can compromise the security of a cryptographic module or the security of the information protected by the module.

Cryptographic Administrator -- An authorized user who has been granted the authority to perform cryptographic initialization and management functions. These users are expected to use this authority only in the manner prescribed by the guidance given to them.

Cryptographic boundary -- An explicitly defined contiguous perimeter that establishes the physical bounds (for hardware) or logical bounds (for software) of a cryptographic module.

Cryptographic key (key) -- A parameter used in conjunction with a cryptographic algorithm that determines [7]:

- the transformation of plaintext data into ciphertext data,
- the transformation of cipher text data into plaintext data,
- a digital signature computed from data,
- the verification of a digital signature computed from data, or
- a data authentication code computed from data.

Cryptographic Module -- The set of hardware, software, firmware, or some combination thereof that implements cryptographic logic or processes, including cryptographic algorithms, and is contained within the cryptographic boundary of the module.

Cryptographic Module Security Policy -- A precise specification of the security rules under which a cryptographic module must operate, including the rules derived from the requirements of this PP and additional rules imposed by the vendor.

---

[3] According to a defined metric.

Defense-in-Depth (DID) -- A security design strategy whereby layers of protection are utilized to establish an adequate security posture for an IT system.

Discretionary Access Control (DAC) -- A means of restricting access to objects based on the identity of subjects and/or groups to which they belong. These controls are discretionary in the sense that a subject with certain access permission is capable of passing that permission (perhaps indirectly) on to any other subject.

DMZ -- A Demilitarized Zone (DMZ) is a network that is mediated by the TOE but, as a result of less stringent access controls, provides access to publicly available services, such as web servers.

Embedded Cryptographic Module -- One that is built as an integral part of a larger and more general surrounding system (i.e., one that is not easily removable from the surrounding system).

Enclave -- A collection of entities under the control of a single authority and having a homogeneous security policy. They may be logical, or may be based on physical location and proximity.

Entity -- A subject, object, user or another IT device, which interacts with TOE objects, data, or resources.

External IT entity -- Any trusted Information Technology (IT) product or system, outside of the TOE, which may, in accordance with the TSP, perform an operation.

Identity -- A representation (e.g., a string) uniquely identifying an authorized user, which can either be the full or abbreviated name of that user or a pseudonym.

Integrity -- A security policy pertaining to the corruption of data and TSF mechanisms.

Integrity label -- A security attribute that represents the integrity level of a subject or an object. Integrity labels are used by the TOE as the basis for mandatory integrity control decisions.

Integrity level -- The combination of a hierarchical level and an optional set of non-hierarchical categories that represent the integrity of data.

Mandatory Access Control (MAC) -- A means of restricting access to objects based on subject and object sensitivity labels.[4]

Mandatory Integrity Control (MIC) -- A means of restricting access to objects based on subject and object integrity labels.

---

[4] The Bell LaPadula model is an example of Mandatory Access Control

Multilevel -- The ability to simultaneously handle (e.g., share, process) multiple levels of data, while allowing users at different sensitivity levels to access the system concurrently. The system permits each user to access only the data to which they are authorized access.

Named Object[5] -- An object that exhibits all of the following characteristics:

- The object may be used to transfer information between subjects of differing user identities within the TSF.
- Subjects in the TOE must be able to request a specific instance of the object.
- The name used to refer to a specific instance of the object must exist in a context that potentially allows subjects with different user identities to request the same instance of the object.

(Note: Due to the deletion of the last sentence in the OS PP (pertaining to intended use of the object being for sharing user data), something may need to be done to the requirements section of the PP (i.e., FDP_ACF_(EXT)) to ensure that some objects, which may satisfy the above but which are not intended for sharing user data do not need a full DAC implementation but rather it is acceptable if they are "owner only" or some other appropriate mechanism.)

Non-Repudiation -- A security policy pertaining to providing one or more of the following:

- To the sender of data, proof of delivery to the intended recipient,
- To the recipient of data, proof of the identity of the user who sent the data.

Object -- An entity within the TSC that contains or receives information and upon which subjects perform operations.

Operating Environment -- The total environment in which a TOE operates. It includes the physical facility and any physical, procedural, administrative and personnel controls.

Operating System (OS) -- An entity within the TSC that causes operations to be performed. Subjects can come in two forms: trusted and untrusted. Trusted subjects are exempt from part or all of the TOE security policies. Untrusted subjects are bound by all TOE security policies.

Operational key -- Key intended for protection of operational information or for the production or secure electrical transmissions of key streams.

Peer TOEs -- Mutually authenticated TOEs that interact to enforce a common security policy.

---

[5]The only named objects in this PP, are operating system controlled files.

Public Object -- An object for which the TSF unconditionally permits all entities "read" access. Only the TSF or authorized administrators may create, delete, or modify the public objects.

Robustness -- A characterization of the strength of a security function, mechanism, service or solution, and the assurance (or confidence) that it is implemented and functioning correctly. DoD has three levels of robustness:

- **Basic:** Security services and mechanisms that equate to good commercial practices.
- **Medium:** Security services and mechanisms that provide for layering of additional safeguards above good commercial practices.
- **High:** Security services and mechanisms that provide the most stringent protection and rigorous security countermeasures.

Secure State -- Condition in which all TOE security policies are enforced.

Security attributes -- TSF data associated with subjects, objects, and users that is used for the enforcement of the TSP.

Security level -- The combination of a hierarchical classification and a set of non-hierarchical categories that represent the sensitivity on the information [10].

Sensitivity label -- A security attribute that represents the security level of an object and that describes the sensitivity (e.g., Classification) of the data in the object. Sensitivity labels are used by the TOE as the basis for mandatory access control decisions [10].

Split key -- A variable that consists of two or more components that must be combined to form the operational key variable. The combining process excludes concatenation or interleaving of component variables.

Subject -- An entity within the TSC that causes operations to be performed.

Symmetric key -- A single, secret key used for both encryption and decryption in symmetric cryptographic algorithms.

Threat -- Capabilities, intentions and attack methods of adversaries, or any circumstance or event, with the potential to violate the TOE security policy.

Threat Agent - Any human user or Information Technology (IT) product or system which may attempt to violate the TSP and perform an unauthorized operation with the TOE.

User -- Any entity (human user or external IT entity) outside the TOE that interacts with the TOE. In the case of the Authorization Server protecting web resources, an "agent" acts on behalf of a user. Therefore, the "user" never truly interacts with the TOE. The authorization server software must have access to the "user's" privilege attributes which are generally maintained in a separate data storage (not part of the TOE).

Vulnerability -- A weakness that can be exploited to violate the TOE security policy.

9 ABBREVIATIONS

The following abbreviations are used in this Protection Profile:

AES Advanced Encryption Standard

ATM Asynchronous Transfer Method

CC Common Criteria for Information Technology Security Evaluation

CAPP Controlled Access Operating System Protection Profile

DES Data Encryption Standard

DoD Department of Defense

DMZ Demilitarized zone

EAL Evaluation Assurance Level

ESP Encapsulating Security Payload

FIPS PUB Federal Information Processing Standard Publication

FTP File Transfer Protocol

GIG Global Information Grid

HTTP Hypertext Transfer Protocol

I&A Identification and Authentication

IATF Information Assurance Technical Framework

ICMP Internet Control Message Protocol

IETF Internet Engineering Task Force

IKE Internet Key Exchange

IPSEC ESP Internet Protocol Security Encapsulating Security Payload

IP Internet Protocol

IT Information Technology

MRE Medium Robustness Environment

NBIAT&S Network Boundary Information Assurance Technologies and Solutions Support

NIAP National Information Assurance Partnership

NIST National Institute of Standards and Technology

NSA National Security Agency

NTP Network Time Protocol

PKI Public Key Infrastructure

PP Protection Profile

RNG Random Number Generator

SFP Security Function Policy

SMTP Simple Mail Transfer Protocol

ST Security Target

TCP Transmission Control Protocol

TFTP Trivial File Transfer Protocol

TOE Target of Evaluation

TSE TOE Security Environment

TSF TOE Security Function

TSP TOE Security Policy

UDP User Datagram Protocol

URL Uniform Research Locator

VPN Virtual Private Network