U.S. Government Protection Profile

Web Server

For

Basic Robustness Environments

Information
Assurance
Directorate

July 25, 2007
Version 1.1

Protection Profile Title:

U.S. Government Protection Profile Web Server for Basic Robustness Environments

Criteria Version:

This Protection Profile "*US Government Protection Profile Web 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.

Table of Contents

1 Introduction to the Protection Profile 6
1.1 PP Identification 6
1.2 Overview of the Protection Profile 6
1.3 Conventions 6
1.4 Glossary of Terms 8
1.5 Document Organization 8
2 TOE Description 9
2.1 Product Type 9
2.2 General TOE Security Functionality 9
2.3 TOE Operational Environment 10
2.3.1 Security Function Policies (SFPs) 13
2.3.2 Basic-Robustness Environment 13
2.3.3 TOE Administration 14
3 Security Environment 15
3.1 Threats 15
3.1.1 Threat Agent Characterization 15
3.2 Organizational Security Policies 18
3.3 Assumptions 19
4 Security Objectives 20
4.1 TOE Security Objectives 20
4.2 Environment Security Objectives 21
5 IT Security Requirements 23
5.1 TOE Security Functional Requirements 23
5.1.1 Security Audit (FAU) 24
5.1.2 Cryptographic Support (FCS) 28
5.1.3 User data protection (FDP) 28
5.1.4 Identification and authentication (FIA) 30
5.1.5 Security management (FMT) 31
5.1.6 Protection of the TOE Security Functions (FPT) 33
5.1.7 Toe Access (FTA) 33
5.2 Security Requirements for the IT Environment 34
5.2.1 IT Environment (FIT) 34
5.3 TOE Security Assurance Requirements 35
5.3.1 Class ADV: Development 36
5.3.2 Class AGD: Guidance documents 38
5.3.3 Class ALC: Life-cycle support 40
5.3.4 Class ATE: Tests 42
5.3.5 Class AVA: Vulnerability assessment 44
6 Rationale 46
6.1 Rationale for TOE Security Objectives 46
6.2 Rationale for the Security Objectives and Security Functional Requirements for the Environment 53
6.3 Rationale for TOE Security Requirements 55
6.4 Rationale for Assurance Requirements 64
6.5 Rationale for Satisfying all Dependencies 64

6.6 Rationale for Extended Requirements 66
7 Appendices 68
A References 69
B Glossary 70
C Acronyms 72
D Robustness Environment Characterization 73
D.1 General Environmental Characterization 73
D.1.1 Value of Resources 73
D.1.2 Authorization of Entities 73
D.1.3 Selection of Appropriate Robustness Levels 74
E Refinements 78

List of Tables

Table 1 Basic Robustness Applicable Threats 17
Table 2 Basic Robustness Applicable Policies 18
Table 3 Basic Robustness Applicable Assumptions 19
Table 4 Basic Robustness Security Objectives 20
Table 5 Basic Robustness Environmental Security Objectives 21
Table 6 Security Functional Requirements 23
Table 7 Auditable Events 25
Table 8 IT Environment Security Functional Requirements 34
Table 9 Assurance Requirements 35
Table 11 Assurance Requirements 35
Table 10 Rationale for TOE Security Objectives 46
Table 11 Rational for IT Environmental Objectives 53
Table 12 Rationale for TOE Security Requirements 55
Table 13 Rationale for IT Environment Requirements 63
Table 14 Functional Requirement Dependencies 64
Table 15 Functional Requirement Dependencies for IT Environment 65
Table 16 Assurance Requirement Dependencies 65
Table 17 Rationale for Extended Requirements 66
Table 18 Rationale for Environmental Requirements 67

1 INTRODUCTION TO THE PROTECTION PROFILE

1.1 PP Identification

Title: U.S. Government Protection Profile Web Server For Basic Robustness Environments

Sponsor: National Security Agency (NSA)

CC Version: Common Criteria (CC) Version 3.1, and applicable interpretations

PP Version: 1.1

Keywords: Web Server, HTTP Server, COTS, commercial security, basic robustness, SSL, TLS, access control, CC EAL2 augmented.

Note: Basic Robustness is EAL2 augmented with ALC_FLR.2: Flaw remediation.

1.2 Overview of the Protection Profile

The "U.S. Government Protection Profile for Web Server in Basic Robustness Environments" specifies security requirements for a commercial-off-the-shelf (COTS) Web Server. A product compliant with this Protection Profile includes, but is not limited to, a Web Server and may be evaluated as a software only application layered on an underlying system (i.e., operating system, hardware, network services and/or custom software) and is usually embedded as a component of a larger system within an operational environment. This profile establishes the requirements necessary to achieve the security objectives of the Target of Evaluation (TOE) and its environment.

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

A conformant product, in conjunction with an IT environment that satisfies all the requirements in this protection profile, provides necessary security services, mechanisms, and assurances to process administrative, private, and sensitive information. The intended environment for conformant products has a relatively low threat for the sensitivity of the data processed. Authorized users, including authorized administrators, of the TOE generally are trusted not to attempt to circumvent access controls implemented by the TOE to gain access to data for which they are not authorized.

1.3 Conventions

Except for replacing United Kingdom spelling with American spelling, the notation, formatting, and conventions used in this PP are consistent with version 3.1 of the CC. Selected presentation choices are discussed here to aid the PP reader.

The CC allows several operations to be performed on functional requirements; *refinement*, *selection*, *assignment*, and *iteration* are defined in paragraph 148 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 that have been made by the PP authors are denoted by *italicized text*, selections to be filled in by the Security Target (ST) author appear in square brackets with an indication that a selection is to be made, [selection:], and are not italicized.

The **assignment** operation is used to assign a specific value to an unspecified parameter, such as the length of a password. Assignments that have been made by the PP authors are denoted by showing the value in square brackets, [Assignment_value], assignments to be filled in by the ST author appear in square brackets with an indication that an assignment is to be made [assignment:].

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). In addition some iterations are given unique identifiers by appending a slash ("/") and an iteration identifier to the element identifiers from the CC. (e.g. FMT_REV.1(1), FMT_REV.1(2))

As this PP was sponsored, in part by National Security Agency (NSA), National Information Assurance Partnership (NIAP) interpretations are used and are presented with the NIAP interpretation number as part of the requirement identifier (e.g., **FAU_GEN.1-NIAP-0410** for Audit data generation).

The CC paradigm also allows protection profile and security target 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)" following the component name.

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

Interp Notes are provided to show the reader where international interpretations have modified a requirement. These modifications will be displayed before or after the affected element.

1.4 Glossary of Terms

See Appendix B for the Glossary.

1.5 Document Organization

Section 1 provides the introductory material for the protection profile.

Section 2 describes the Target of Evaluation in terms of its envisaged usage and connectivity.

Section 3 defines the expected TOE security environment in terms of the threats to its security, the security assumptions made about its use, and the security policies that must be followed.

Section 4 identifies the security objectives derived from these threats and policies.

Section 5 identifies and defines the security functional requirements from the CC that must be met by the TOE and the IT environment in order for the functionality-based objectives to be met. This section also identifies the security assurance requirements for EAL2 augmented.

Section 6 provides a rationale to demonstrate that the Information Technology Security Objectives satisfy the policies and threats. Arguments are provided for the coverage of each policy and threat. The section then explains how the set of requirements are complete relative to the objectives, and that each security objective is addressed by one or more component requirement. Arguments are provided for the coverage of each objective.

Section 7, Appendices, includes the appendices that accompany the PP and provides clarity and/or explanation for the reader.

Appendix A, References, provides background material for further investigation by users of the PP.

Appendix B, Glossary, provides a listing of definitions of terms.

Appendix C, Acronyms, provides a listing of acronyms used throughout the document.

Appendix D, Robustness Environment Characterization, contains a discussion characterizing the level of robustness TOEs compliant with the PP can achieve. The PPRB created a discussion that provides a definition of factors for TOE environments as well as an explanation of how a given level of robustness is categorized.

Appendix E, Refinements, identifies the refinements that were made to CC requirements where text is deleted from a requirement.

2 TOE DESCRIPTION

2.1 Product Type

The product type of the Target of Evaluation (TOE) described in this Protection Profile (PP) is a Web Server or Hypertext Transport Protocol (HTTP) Server designed to receive requests for information (or content) and deliver that information to a web user.

HTTP servers were originally designed to receive anonymous requests from unauthenticated hosts on the Internet. However, HTTP servers have evolved to deliver both **public content** and restricted information (or **controlled access content**) through a common client interface (a "browser") and referenced by a Universal Resource Locator (URL). Public content is information available to any web user that requests it without authentication. Controlled access content is information available only to web users authorized for that content by the content provider. Note that each content provider has control over the sets of web users authorized to access their content.

The content that is made available by the web server represents information provided by a **content provider** to a web user. Static content is provided to the web user 'as is', with no processing performed by the web server (i.e., HTML, Java, JavaScript). Dynamic content is content that is generated on the fly, being assembled either by the server or as the output of executable content.

For the purposes of this protection profile, **Web Servers** are application programs. They execute on a host platform that provides the underlying abstractions used to store content and execute programs. The Web Server controls access to information by means of its own security features in combination with the features provided by the host platform.

2.2 General TOE Security Functionality

A Web Server evaluated against this PP will provide security services either completely by itself or in cooperation with the host operating system.

Security services provided by the TOE are:

- Access Control, which controls access to objects based on the identity of the subjects, and which allows authorized users to specify how the objects that they control are protected.

- Identification and Authentication (I&A) by which users are uniquely identified and authenticated before they are authorized to access information stored on the Web Server.

- Audit Capture is the function that creates information on all auditable events.

- Authorized administration role to allow authorized administrators to configure the policies for access control, identification and authentication, and auditing. The TOE must enforce the authorized administration role.

- Cryptographic Services to establish secure sessions between itself and web clients.

The following security services must be provided by the operating system located in the IT environment:

- Identification and Authentication (I&A) to protect access information stored on the Web Server.

- Discretionary Access Control to allow authorized users to specify which resources may be assessed by which user.

- Audit Storage is the service that stores records for all security-relevant operations that users perform on user and Web Server data.

- Audit Review service that allows the authorized administrator to review stored audit records in order to detect potential and actual security violations.

- Non-bypassibility of the security functions to prohibit any access to data or the TOE that is not governed by the TOE security policies.

- Domain separation will ensure that other software operating on the same computer as the TOE cannot interfere with or negate the security functions of the TOE. Domain separation also ensures that multiple instances of the TOE concurrently executing cannot interfere with one another.

2.3 TOE Operational Environment

The TOE is expected to be executing on an OS and hardware platform that have been evaluated against a NIAP validated (basic robustness or higher) operating system PP. The evidence of a validated OS can be met by providing the NIAP/CCEVS certificate that the underlying operating system is compliant with the Controlled Access Protection Profile or with a protection profile at the Basic Level of Robustness or greater since all of security services provided by the operating system are included in the validated OS. In addition, if the Web Server implementation relies on support from software, other than an OS, that software should be evaluated as part of the TOE. If the TOE implements a product previously evaluated against a protection profile at the Basic Level of Robustness or greater the NIAP/CCEVS certificate from that evaluation may be used as evidence that the product is compliant with the PP. .

The physical Web Server is physically protected to a level commensurate with the data it processes. Only authorized administrators are allowed physical access to the server. All

users and administrators are cleared to the level of the information being served but some users may not possess a need-to-know to all of the information being served.

The administrator establishes the configuration of the server, and controls the set of authorized content providers. To secure the content provided by the TOE, the administrator and content providers have the capability to control web user's access to the content.

The TOE responds to requests for public information using the Hypertext Transport Protocol (HTTP). The content provided can reside in static files (e.g. HTML files) or dynamic content can be generated "on-the-fly" (e.g. Common Gateway Interface, Active Server Pages, Java Server Pages etc.). Web applications that create content on-the-fly are beyond the scope of this PP.

The TOE is also able to deliver restricted content using HTTP (secure) also referred to as HTTPS using FIPS 140-2 validated SSL v3.0 or TLS v1.0. Identification and authentication of web users is provided through personal digital certificates or through user ID and password schemes[1].

While HTTP is an extensible protocol, the standard (RFC 2616) defines the following eight methods that can performed on the resource identified by the requested Universal Resource Identifier (URI): OPTIONS, GET, HEAD, POST, PUT, DELETE, TRACE and CONNECT. The ST must address any additional methods supported by the TOE in a manner consistent with the objectives defined in this PP.

The installation guidelines used by the content providers that will determine acceptable and unacceptable content that will be placed on the Web Server for use. The content provider will determine if the TOE will implement certain cryptographic protocols (e.g., HTTPS using FIPS 140-2 validated SSL v3.0 or TLS v1.0) so that information is restricted from public access. These cryptographic protocols will allow the client and server resources to exchange information in a secure manner.

[1] The TOE may also provide support for password protection and the serving of password protected content over unencrypted connections, but such support is not a secure usage for protected data, and is assumed not to be used by those who consider their data controlled access.

Figure 2-1 provides the conceptual model of the TOE's placement in an overall network. Alternately, multiple forms of network application services (Web Server, FTP server, terminal server) could be located on the same machine. The key point, applicable to the services, is that the operating system provides low-level mediation of access to files. See figure 2-2 for a conceptual view of the TOE and its execution environment.

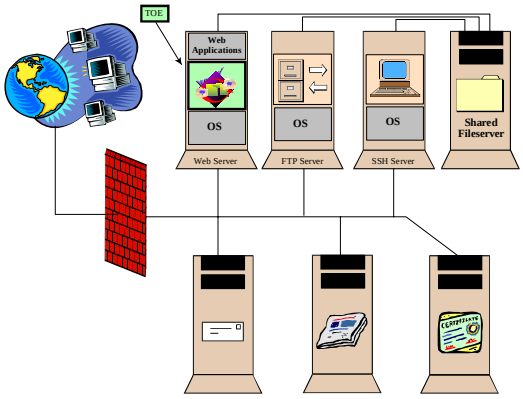


Figure 2-1: Placement of the TOE in an overall System Architecture

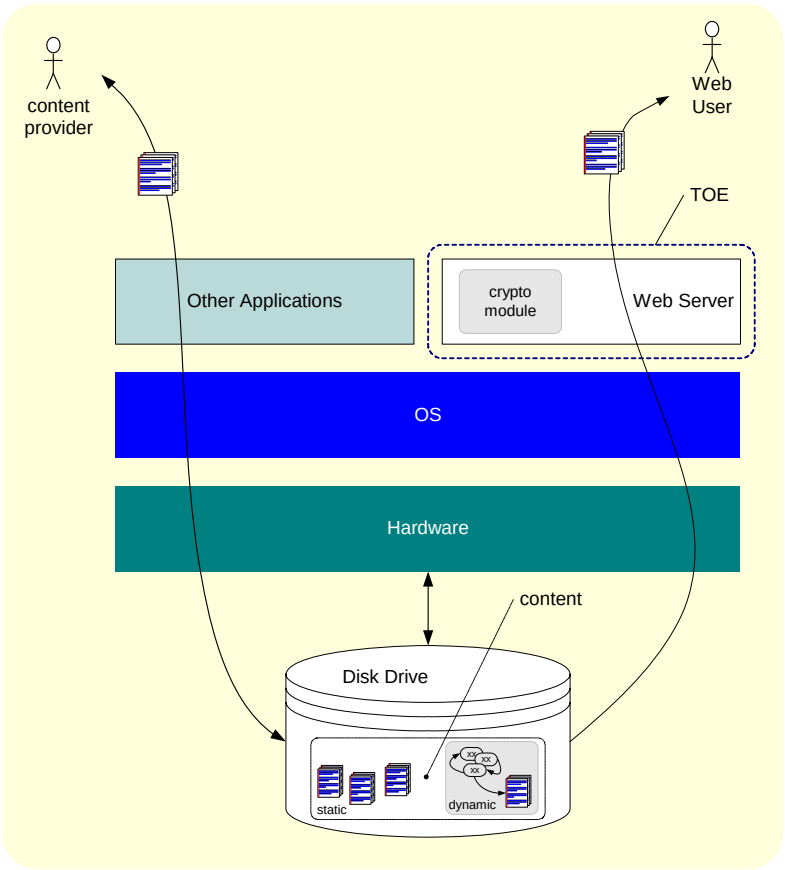


Figure 2-2: The TOE and its execution environment

2.3.1 Security Function Policies (SFPs)

TOE evaluation is concerned primarily with ensuring that a defined TOE Security Policy (TSP) is enforced over the TOE resources. The TSP defines the rules by which the TOE governs access to its resources, and thus all information and services controlled by the TOE.

The TSP is, in turn, made up of multiple Security Function Policies (SFPs). Each SFP has a scope of control, that defines the subjects, objects, and operations controlled under the SFP. The SFP is implemented by a Security Function (SF), whose mechanisms enforce the policy and provide necessary capabilities.

Web User (WU) SFP

The intent of the Web User SFP is to control access by entities accessing the server over the network to obtain content. All other operations between these subjects and objects are expressly denied. The Web User SFP is summarized in the following table:

Subject[2]	Object	Operation[3]	Description
User	Public content	Read	Any user may access any public content provided by the TOE.
User	Controlled-access content	Read	To access controlled content, an authorized user must be identified and authenticated and the access must be explicitly permitted by the content provider.

2.3.2 Basic-Robustness Environment

The TOE described in this PP is intended to operate in environments having a basic level of robustness as defined in the Glossary in Appendix B.

Basic robustness allows processing of data at a single sensitivity level in an environment where users are cooperative and threats are minimal. Authorized users of the TOE are cleared for all information managed by the Web Server, but may not have the need-to-know authorization for all of the data. Hence, the risk that significant damage will be done due to compromise of data is low.

Entities in the IT environment on which the TOE depends for security functions must be of at least the same level of robustness as the TOE. It is necessary for such an environment that the underlying operating system on which the Web Server is installed be evaluated against a basic robustness protection profile for operating systems.

[2] A subject is a process acting on behalf of the entity specified.
[3] The only operation permitted by this SFP is the read operation. Other operations (e.g. delete, modify, rename) that may exist are outside of the scope of the TOE

The TOE in and of itself is not of sufficient robustness to store and protect information of such criticality that the integrity or secrecy is critical to the survival of the enterprise.

2.3.3 TOE Administration

Authorized administrators of the TOE will have capabilities that are commensurate with their assigned administrative roles. There may be one or more administrative roles. The TOE developers will establish some roles for their products. If the security target allows it, the administrators of the system may establish other roles. This PP defines one necessary administrator role (authorized administrator) and allows the Web Server developer or ST writer to define more. When the Web Server is established, the ability to segment roles and assign capabilities with significant freedom regarding the number of roles and their responsibilities must also exist. Of course, the very ability to establish and assign roles will be a privileged function.

3 SECURITY ENVIRONMENT

The security environment for the functions addressed by this specification includes threats, security policies, and usage assumptions, as discussed below.

Basic robustness TOEs fall in the upper left area of the robustness figures shown in Appendix D. 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.

3.1 Threats

3.1.1 Threat Agent Characterization

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 to specify exactly 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 is "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.

The following threats, which were drawn from the *Consistency Instruction Manual* (CIM) *for Development of US Government Protection Profiles for Use in Basic Robustness Environments,* Version 3.0, are addressed by the TOE, and should be read in conjunction with the threat rationale, Section 6.1. There are other threats that the TOE does not address (e.g., malicious developer inserting a backdoor into the TOE) and it is up to a site to determine how these types of threats apply to its environment.

Table 1 Basic Robustness Applicable Threats

Threat	Definition
T. ACCIDENTAL_ADMIN_ERROR	An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.
T.ACCIDENTAL_ CRYPTO_ COMPROMISE	A 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.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.

Threat	Definition
T.POOR_TEST	Lack of or insufficient tests 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	A malicious user or process may cause configuration data to be inappropriately accessed (viewed, modified or deleted).
T.UNAUTHORIZED_ACCESS	A user may gain unauthorized access to user data for which they are not authorized according to the TOE security policy.
T.UNIDENTIFIED_ACTIONS	Failure of the authorized administrator to identify and act upon unauthorized actions may occur.

3.2 Organizational Security Policies

An organizational security policy is a set of rules, practices, and procedures imposed by an organization to address its security needs

Table 2 Basic Robustness Applicable Policies

Policy	Definition
P.ACCOUNTABILITY	The authorized users of the TOE shall be held accountable for their actions within the TOE.
P.CRYPTOGRAPHY	All cryptographic-based security components used to protect sensitive information must be FIPS 140-2 validated.
P.ROLES	The TOE shall provide an authorized administrator role for secure administration of the TOE. This role shall be separate and distinct from other authorized users.

3.3 Assumptions

This section contains assumptions regarding the environment in which the TOE will reside.

Table 3 Basic Robustness Applicable Assumptions

Assumption	Definition
A.NO_EVIL	Administrators are non-hostile, appropriately trained, and follow all administrator guidance.
A.NO_GENERAL_PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on Web Server, other than those services necessary for the operation, administration and support of the Web Server.
A.OS_PP_VALIDATED	The underlying OS has been validated against an NSA sponsored OS PP of at least Basic Robustness.
A.PHYSICAL	It is assumed that appropriate physical security is provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.

4 SECURITY OBJECTIVES

This section identifies the security objectives of the TOE and its supporting environment. The security objectives identify the responsibilities of the TOE and its environment in meeting the security needs.

4.1 TOE Security Objectives

Table 4 Basic Robustness Security Objectives

Objective Name	Objective Definition
O.ACCESS_HISTORY	The TOE will store and retrieve information (to authorized users) related to previous attempts to establish a session.
O.ADMIN_GUIDANCE	The TOE will provide administrators with the necessary information for secure management.
O.ADMIN_ROLE	The TOE will provide authorized administrator roles to isolate administrative actions.
O.AUDIT_GENERATION	The TOE will provide the capability to detect and create records of security relevant events associated with users.
O.CONFIGURATION_IDENTIFICATION	The configuration of the TOE is fully identified in a manner that will allow implementation errors to be identified and corrected with the TOE being redistributed promptly.
O.CRYPTOGRAPHY	All cryptographic-based security components used to protect sensitive information must be FIPS 140-2 validated.
O.DOCUMENTED_DESIGN	The design of the TOE is adequately and accurately documented.

Objective Name	Objective Definition
O.MANAGE	The TOE will provide all the functions and facilities necessary to support the authorized 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_FUNCTIONAL_TEST	The TOE will undergo some security functional testing that demonstrates that the TSF satisfies some of its security functional requirements.
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 within its Scope of Control 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.
O.VULNERABILITY_ANALYSIS	The TOE will undergo some vulnerability analysis to demonstrate that the design and implementation of the TOE does not contain any obvious flaws.

4.2 Environment Security Objectives

Table 5 Basic Robustness Environmental Security Objectives

Environmental Objective Name	Environmental Objective Definition
OE.NO_EVIL	Sites using the TOE shall ensure that authorized administrators are non-hostile, appropriately trained and follow all administrator guidance.

Environmental Objective Name	Environmental Objective Definition
OE.NO_GENERAL_ PURPOSE	There will be no general-purpose computing capabilities (e.g., compilers or user applications) available on Web servers, other than those services necessary for the operation, administration and support of the Web Server.
OE.OS_PP_VALIDATED	The underlying OS has been validated against an NSA sponsored OS PP of at least Basic Robustness.
OE.PHYSICAL	Physical security will be provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.

5 IT SECURITY REQUIREMENTS

5.1 TOE Security Functional Requirements

This section defines the functional requirements for the TOE. Functional requirements in this PP were drawn directly from Part 2 of the CC, or were based on Part 2 of the CC, including the use of NIAP and International Interpretations and extended components. These requirements are relevant to supporting the secure operation of the TOE.

Table 6 Security Functional Requirements

Functional Components	
FAU_GEN.1-NIAP-0410	Audit data generation
FAU_GEN_(EXT).2	User identity association
FAU_SEL.1-NIAP-0407	Selective audit
FCS_BCM_(EXT).1	Baseline Cryptographic Module
FDP_ACC.1	Subset access control
FDP_ACF.1-NIAP-0407	Security attribute based access control
FDP_RIP.1	Subset residual information protection
FIA_ATD.1	User attribute definition
FIA_UAU.1	Timing of authentication
FIA_UID.1	Timing of identification
FMT_MOF.1	Management of security functions behavior
FMT_MSA.1	Management of security attributes
FMT_MSA_(EXT).3	Static attribute initialization
FMT_MTD.1	Management of TSF data
FMT_REV.1(1)	Revocation (user attributes)
FMT_REV.1(2)	Revocation (subject, object attributes)

Functional Components	
FMT_SMF.1	Specification of management functions
FMT_SMR.1	Security roles
FTA_MCS.1	Basic limitation on multiple concurrent sessions
FTA_SSL.3	TSF-initiated session termination
FTA_TAH_(EXT).1	TOE access history
FTA_TSE.1	TOE session establishment
FTP_ITC.1	Inter-TSF trusted channel

5.1.1 Security Audit (FAU)

5.1.1.1 Audit data generation (FAU_GEN.1-NIAP-0410)

FAU_GEN.1.1-NIAP-0410 **Refinement**: The TSF shall be able to generate an audit record of the following auditable events:

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

b) All auditable events for the *minimum* level of audit **listed in Table 7**;

c) **[Start-up and shutdown of the Web Server;**

d) **Use of special permissions (e.g., those often used by authorized** administrators **to circumvent access control policies); and**

e) [selection: [assignment: events at a minimal level of audit introduced by the inclusion of additional SFRs determined by the ST author], [assignment: events commensurate with a minimal 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 or both of the assignments (as detailed in the following paragraphs), or select "no additional events".

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

Application Note: Likewise, if the ST author includes extended requirements not contained in this PP, the corresponding audit events must be added in the second assignment. Because "minimal" 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 minimal level for similar requirements.

Application Note: If no additional (CC or extended) SFRs are included, or if additional SFRs are included that do not have "minimal" audit associated with them then it is acceptable to assign "no additional events" in this item.

FAU_GEN.1.2-NIAP-0410 The TSF shall record within each audit record at least the following information:

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

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [information specified in column three of Table 7 below].

Application Note: In column 3 of the table below, "Audit Record Contents" 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 item a) above) for a particular auditable event type, then an assignment of "none" is acceptable.

Table 7 Auditable Events

Security Functional Requirement	Auditable Event(s)	Additional Audit Record Contents
FAU_GEN.1-NIAP-0410	None	
FAU_GEN_(EXT).2	None	
FAU_SEL.1-NIAP-0407	All modifications to the audit configuration that occur while the audit collection functions are operating	The identity of the authorized administrator that made the change to the audit configuration
FCS_BCM_(EXT).1	Establishment of secure sessions between the server and web clients.	Identity of the subject performing the operation, time and date, and the type of operation being performed.
FDP_ACC.1	None	

Security Functional Requirement	Auditable Event(s)	Additional Audit Record Contents
FDP_ACF.1-NIAP-0407	Successful requests to perform an operation on an object covered by the SFP	The identity of the subject performing the operation
FDP_RIP.1	None	
FIA_ATD.1	None	
FIA_UAU.1	Unsuccessful use of the user authentication mechanism	The identity of the subject performing the operation and the type of operation being performed.
FIA_UID.1	Unsuccessful use of the user identification mechanism	The identity of the subject performing the operation and the type of operation being performed.
FMT_MOF.1	None	
FMT_MSA.1	All modifications of the values of security attributes.	Identity of individual attempting to modify security attributes The current values of the attributes and the changed value
FMT_MSA_(EXT).3	Modification of the default setting of the restrictive rules	Identity of individual attempting to modify security attributes The values of the current rules and the attempted change value
FMT_MTD.1	None	
FMT_REV.1(1)	Unsuccessful revocation of security attributes	Identity of individual attempting to revoke security attributes
FMT_REV.1(2)	Unsuccessful revocation of security attributes	Identity of individual attempting to revoke security attributes
FMT_SMF.1	Use of the management functions	Identity of the administrator performing these functions
FMT_SMR.1	Modifications to the group of users that are part of a role	Identity of authorized administrator modifying the role definition

Security Functional Requirement	Auditable Event(s)	Additional Audit Record Contents
FTA_MCS.1	Rejection of a new session based on the limitation of multiple concurrent sessions	Identity of the session being rejected and the number of sessions that is being exceeded.
FTA_SSL.3	Initiation of a session termination	Identity of the session being terminated and the reason for the termination
FTA_TAH_(EXT).1	None	
FTA_TSE.1	Denial of a session establishment due to the session establishment mechanism	Identity of the individual attempting to establish a session
FTP_ITC.1	Establishing a Trusted Path	Identity of the users of the trusted path

5.1.1.2 User identity association (FAU_GEN_(EXT).2)

FAU_GEN_(EXT).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.

5.1.1.3 Selective audit (FAU_SEL.1-NIAP-0407)

FAU_SEL.1.1-NIAP-0407 **Refinement**: The TSF shall **allow only the administrator** to include or exclude auditable events from the set of audited events based on the following attributes:

a) *user identity*,

b) *event type*,

c) *object identity*,

d) [selection: “subject identity”, “host identity”, “none”];

e) [success of auditable security events;

f) failure of auditable security events; and

g) [selection: [assignment: list of additional criteria that audit selectivity is based upon], "no additional criteria"].]

Application Note: "event type" is to be defined by the ST author; the intent is to be able to include or exclude classes of audit events.

*Application Note: The intent of this requirement is to capture enough audit data to allow the administrator to perform their task, not necessarily to capture only the needed audit data. In other words, the **Web Server** does not necessarily need to include or exclude auditable events based on all attributes at any given time.*

5.1.2 Cryptographic Support (FCS)

The cryptographic requirements shall comply with FCS_BCM_(EXT) defined below. The requirements will enable the TOE to deliver restricted content using HTTP (secure) also referred to as HTTPS using FIPS 140-2 validated SSL v3.0 or TLS v1.0. .

5.1.2.1 FCS_BCM_(EXT).1: Baseline Cryptographic Module

Hierarchical to: No other components.

FCS_BCM_(EXT).1.1 The cryptomodule module shall be FIPS PUB 140-2 validated against SSL version 3.0 or TLS version 1.0 or later versions.

5.1.3 User data protection (FDP)

5.1.3.1 Subset access control (FDP_ACC.1)

FDP_ACC.1.1 The TSF shall enforce the [Access Control policy] on [all subjects, all **Web Server**-controlled objects and all operations among them].

5.1.3.2 Security attribute based access control (FDP_ACF.1-NIAP-0407)

Interp Note: The following element was modified per CCIMB Interpretation 103.

FDP_ACF.1.1-NIAP-0407 The TSF shall enforce the [Access Control policy] to objects based on the following:

- [the authorized user identity associated with a subject;
- access operations implemented for **Web Server**-controlled objects; and
- object identity].

*Application Note: **Web Server**-controlled objects may be implementation-specific objects that are presented to authorized users at the user interface to the **Web Server**. They may include, but are not limited to tables, records, files, indexes, views, constraints, stored*

*queries, and metadata. Data structures that are not presented to authorized users at the **Web Server** user interface, but are used internally are internal TSF data structures. Internal TSF data structures are not controlled according to the rules specified in FDP_ACF.1-NIAP-0407.*

FDP_ACF.1.2-NIAP-0407 **Refinement**: The TSF shall enforce the following rules to determine if an operation among controlled subjects and **Web Server-**controlled objects is allowed:

The Access Control policy mechanism shall, either by explicit authorized user action or by default, provide that the Web Server controlled objects are protected from unauthorized access according to the following ordered rules:

[selection:

a) If the requested mode of access is denied to that authorized user, deny access;

b) If the requested mode of access is permitted to that authorized user, permit access;

c) Else, deny access,

OR

a) [If the requested mode of access is denied to that authorized user, deny access;

b) If the requested mode of access is permitted to that authorized user, permit access;

c) Else, deny access

].

Application Note: The deny mode of access may be implicit.

FDP_ACF.1.3-NIAP-0407 **Refinement**: The TSF shall explicitly authorize access of subjects to **Web Server-controlled** 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"].

Application Note: This element allows specifications of additional rules for authorized administrators to bypass the Access Control policy for system management or maintenance (e.g., system backup).

FDP_ACF.1.4-NIAP-0407 The TSF shall explicitly deny access of subjects to objects based on the following rules: [selection: [assignment: rules, based on security attributes, that explicitly deny access of subjects to objects], "no additional explicit denial rules"].

5.1.3.3 Subset residual information protection (FDP_RIP.1)

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the *allocation of the resource to* [assignment: list of objects].

5.1.4 Identification and authentication (FIA)

5.1.4.1 User attribute definition (FIA_ATD.1)

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users:

- [User identifier;
- Security-relevant roles; and
- [assignment: list of security attributes]].

Application Note: The intent of this requirement is to specify the TOE security attributes that the TOE utilizes to determine access. These attributes may be controlled by the environment or by the TOE itself.

5.1.4.2 FIA_UAU.1: Timing of Authentication

Hierarchical to: No other components

FIA_UAU.1.1	The TSF shall allow *the GET operation on public content* on behalf of the user to be performed before the user is authenticated.
FIA_UAU.1.2	The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

Dependencies: FIA_UID.1 Timing of identification

5.1.4.3 FIA_UID.1: Timing of Identification

Hierarchical to: No other components

FIA_UID.1.1	The TSF shall allow *only access of content designated as public* on behalf of the user to be performed before the user is identified.
FIA_UID.1.2	The TSF shall require each user to have been successfully identified before allowing other any TSF-mediated actions on behalf of that user.

Application Note: If the underlying IT environment provides identification services for content providers and administrators, it is acceptable for FIA_UID.1.2 to be satisfied by the presentation and verification of those credentials.

5.1.5 Security management (FMT)

5.1.5.1 Management of security functions behavior (FMT_MOF.1)

FMT_MOF.1.1 The TSF shall restrict the ability to *disable and enable* the functions [relating to the specification of events to be audited] to [authorized administrators].

5.1.5.2 Management of security attributes (FMT_MSA.1)

FMT_MSA.1.1 **Refinement**: The TSF shall enforce the [Access Control policy] to restrict the ability to [*manage*] **all** the security attributes to [authorized administrators].

Application Note: The ST author should ensure that all attributes identified in FIA_ATD.1 are adequately managed and protected.

5.1.5.3 FMT_MSA_(EXT).3 Static attribute initialization

Hierarchical to: No other components.

FMT_MSA_(EXT).3.1 The TSF shall enforce the [Access Control policy] to provide *restrictive* default values for security attributes that are used to enforce the SFP.

Application Note: This requirement applies to new container objects at the top-level (e.g., tables). When lower-level objects are create (e.g., rows, cells), these may inherit the permissions of the top-level objects by default. In other words, the permissions of the 'child' objects can take the permissions of the 'parent' objects by default.

5.1.5.4 Management of TSF data (FMT_MTD.1)

FMT_MTD.1.1 The TSF shall restrict the ability to *include or exclude* the [auditable events] to [authorized administrators].

5.1.5.5 Revocation (FMT_REV.1(1))

FMT_REV.1.1(1) The TSF shall restrict the ability to revoke security attributes associated with the *users* within the TSC to [the authorized administrator].

FMT_REV.1.2(1) The TSF shall enforce the rules [assignment: specification of revocation rules].

5.1.5.6 Revocation (FMT_REV.1(2))

FMT_REV.1.1(2) The TSF shall restrict the ability to revoke security attributes associated with the *objects* within the TSC to [the authorized administrator and users as allowed by the Access Control policy].

FMT_REV.1.2(2) The TSF shall enforce the rules [assignment: specification of revocation rules].

5.1.5.7 Specification of Management Functions (FMT_SMF.1)

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

5.1.5.8 Security roles (FMT_SMR.1)

FMT_SMR.1.1 **Refinement**: The TSF shall maintain the roles:

- [authorized administrator]; **and**
- **[assignment: additional authorized identified roles]**.

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

Application Note: This requirement identifies a minimum set of management roles. A ST or operational environment may contain a finer-grain decomposition of roles that correspond to the roles identified here (e.g., administrator and non-administrative user). The ST writer may change the names of the roles identified above but the "new" roles must still perform the functions that the FMT requirements in this PP have defined.

5.1.6 Protection of the TOE Security Functions (FPT)

5.1.7 Toe Access (FTA)

5.1.7.1 FTA_SSL.3: TSF-initiated session termination

Hierarchical to: No other components

FTA_SSL.3.1 The TSF shall terminate an interactive **HTTPS** session after a [*Web Server Administrator-configurable time interval of session inactivity*].

Dependencies: No dependencies

Application Note: HTTP and HTTPS are state-less protocols that were designed to allow an anonymous user to request a document and the server to service that request. Several mechanisms (e.g. session cookies, URL rewriting, SSL ID tracking, etc.) have been devised to bind a set of requests to a user or browser, providing HTTP sessions. To meet FTA_SSA_(EXT).1.1, the TOE must be able to detect a period of inactivity in each HTTP session and terminate any session if that period exceeds a Web Server *Administrator determined period of time.*

5.1.7.2 Basic limitation on multiple concurrent sessions (FTA_MCS.1)

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

FTA_MCS.1.2 **Refinement**: The TSF shall enforce, by default, a limit of **[selection:** [assignment: default number]**, "an admin configurable number of"]** sessions per user.

5.1.7.3 TOE access history (FTA_TAH_(EXT).1)

FTA_TAH_(EXT).1.1 Upon successful session establishment, the TSF shall be able to display the *date and time* of the last successful session establishment to the user.

FTA_TAH_(EXT).1.2 Upon successful session establishment, the TSF shall be able to display the *date and time* of the last unsuccessful attempt to session establishment and the number of unsuccessful attempts since the last successful session establishment.

FTA_TAH_(EXT).1.3 The TSF shall not erase the access history information from the user interface without giving the user an opportunity to review the information.

5.1.7.4 TOE session establishment (FTA_TSE.1)

FTA_TSE.1.1 **Refinement**: The TSF shall be able to deny session establishment based on [attributes that can be set explicitly by authorized administrator(s), including user identity, time of day, day of the week], **and [assignment: list of additional attributes]**.

5.1.7.5 FTP: Trusted Path

FTP_ITC.1: Inter-TSF trusted channel

Hierarchical to: No other Component

FTP_ITC.1.1 The TSF shall provide a communication channel between itself 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.

FTP_ITC.1.2 The TSF shall permit *either the TSF or the remote trusted IT product* to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for *the transmission of controlled-access content*.

Application note: the intention is that HTTPS would be the mechanism to satisfy this FTP_ITC requirement.

5.2 Security Requirements for the IT Environment

This section contains the security functional requirements for the IT environment. With the TOE being a software-only TOE, the IT environment must provide protection of the TOE from tampering and interference. The TOE can also satisfy these requirements since the TOE is part of the IT environment. The functions to be supported using the validated operating system are listed in section 2.2.

Table 8 IT Environment Security Functional Requirements

IT Environment Security Functional Requirements	
FIT_PPC_(EXT).1	IT Environment Protection Profile Compliance

5.2.1 IT Environment (FIT)

5.2.1.1 IT Environment Protection Profile Compliance (FIT_PPC_(EXT).1)

FIT_PPC_(EXT).1.1 The IT environment shall be compliant with the requirements of the Controlled Access Protection Profile or an Operating System Protection Profile at the Basic Level of Robustness or Greater.

Application Note: This requirement can be met by providing evidence (e.g., certificate) that the underlying operating system is compliant with the Controlled Access Protection Profile or with a protection profile at the Basic Level of Robustness or greater.

5.3 TOE Security Assurance Requirements

The agreed upon Security Assurance Requirements drawn from the Common Criteria for Information Technology Security Evaluation, Part 3, dated Aug.99, Version 2.1 of CCIB-99-031 which collectively define "Basic Robustness" include the following:

All of the assurance requirements included in Evaluated Assurance Level (EAL) 2 augmented with the following additions:

- ALC_FLR.2: Flaw remediation

The following is a list of the assurance requirements needed for Basic Robustness.

Table 9 Assurance Requirements

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
	ATE_IND.2	Independent testing - conformance
Vulnerability Assessment	AVA_VAN.2	Vulnerability analysis

Table 10 Assurance Requirements

5.3.1 Class ADV: Development

5.3.1.1 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.

5.3.1.2 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.

5.3.1.3 **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.

5.3.2 Class AGD: Guidance documents

5.3.2.1 **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.

5.3.2.2 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.

5.3.3 Class ALC: Life-cycle support

5.3.3.1 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.

5.3.3.2 ALC_CMS.2 Parts 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.

5.3.3.3 **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.

5.3.3.4 **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.

5.3.4 Class ATE: Tests

5.3.4.1 **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.

5.3.4.2 **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.

5.3.4.3 **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.

5.3.5 Class AVA: Vulnerability assessment

5.3.5.1 **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 provides the rationale for the selection of the IT security requirements, objectives, assumptions, and threats. In particular, it shows that the IT security requirements are suitable to meet the security objectives, which in turn are shown to be suitable to cover all aspects of the TOE security environment.

6.1 Rationale for TOE Security Objectives

Table 11 Rationale for TOE Security Objectives

Threat/Policy	Objectives Addressing the Threat/Policy	Rationale
T. ACCIDENTAL_ADMIN_ERROR An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.	O.ADMIN_GUIDANCE The TOE will provide administrators with the necessary information for secure management.	O.ADMIN_GUIDANCE helps to mitigate this threat by ensuring the TOE administrators have guidance that instructs them how to administer the TOE in a secure manner. Having this guidance helps to reduce the mistakes that an administrator might make that could cause the TOE to be configured in insecurely.
T.ACCIDENTIAL_CRYPTO_COMPROMISE A 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.	O.RESIDUAL_INFORMATION The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated. O.PARTIAL_SELF_PROTECTION The TOE will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure through it own interfaces.	O.RESIDUAL_INFORMATION; contributes to the mitigation of this threat by ensuring that any residual data is removed from network packet objects and ensuring that cryptographic material is not accessible once it is no longer needed. O.PARTIAL_SELF_PROTECTION ensures that the TOE will have adequate protection from external sources and that all TSP functions are invoked.

Threat/Policy	Objectives Addressing the Threat/Policy	Rationale
T.MASQUERADE A user or process may masquerade as another entity in order to gain unauthorized access to data or TOE resources.	O.TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to the TOE.	O.TOE_ACCESS mitigates this threat by controlling the logical access to the TOE and its resources. By constraining how and when authorized users can access the TOE, and by mandating the type and strength of the authentication mechanism this objective helps mitigate the possibility of a user attempting to login and masquerade as an authorized user. 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.
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.	O.CONFIGURATION_IDENTIFICATION The configuration of the TOE is fully identified in a manner that will allow implementation errors to be identified and corrected with the TOE being redistributed promptly.	O.CONFIGURATION_IDENTIFICATION plays a role in countering this threat by requiring the developer to provide control of the changes made to the TOE's design.
	O.DOCUMENTED_DESIGN The design of the TOE is adequately and accurately documented.	O.DOCUMENTED_DESIGN ensures that the design of the TOE is documented, permitting detailed review by evaluators.
	O.VULNERABILITY_ANALYSIS The TOE will undergo some vulnerability analysis to demonstrate the design and implementation of the TOE does not contain any obvious flaws.	O.VULNERABILITY_ANALYSIS ensures that the design of the TOE is analyzed for design flaws.

Threat/Policy	Objectives Addressing the Threat/Policy	Rationale
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.	O.CONFIGURATION_IDENTIFICATION The configuration of the TOE is fully identified in a manner that will allow implementation errors to be identified and corrected with the TOE being redistributed promptly.	O.CONFIGURATION_IDENTIFICATION plays a role in countering this threat by requiring the developer to provide control of the changes made to the TOE's design, although the previous three objectives help minimize the introduction of errors into the implementation.
	O.PARTIAL_FUNCTIONAL_TEST The TOE will undergo some security functional testing that demonstrates the TSF satisfies some of its security functional requirements.	O.PARTIAL_FUNCTIONAL_TEST increases the likelihood that any errors that do exist in the implementation (with respect to the functional specification, high-level, and low-level design) will be discovered through testing.
	O.VULNERABILITY_ANALYSIS The TOE will undergo some vulnerability analysis to demonstrate the design and implementation of the TOE does not contain any obvious flaws.	O.VULNERABILITY_ANALYSIS helps reduce errors in the implementation that may not be discovered during functional testing. Ambiguous design documentation and the fact that exhaustive testing of the external interfaces is not required may leave bugs in the implementation undiscovered in functional testing.
T.POOR_TEST Lack of or insufficient tests 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.	O.DOCUMENTED_DESIGN The design of the TOE is adequately and accurately documented.	O.DOCUMENTED_DESIGN helps to ensure that the TOE's documented design satisfies the security functional requirements. In order to ensure the TOE's design is correctly realized in its implementation, the appropriate level of functional testing of the TOE's security mechanisms must be performed during the evaluation of the TOE.

Threat/Policy	Objectives Addressing the Threat/Policy	Rationale
	O.PARTIAL_FUNCTIONAL_TEST The TOE will undergo some security functional testing that demonstrates the TSF satisfies some of its security functional requirements.	O.PARTIAL_FUNCTIONAL_TEST increases the likelihood that any errors that do exist in the implementation (with respect to the functional specification, high level, and low-level design) will be discovered through testing.
	O.VULNERABILITY_ANALYSIS The TOE will undergo some vulnerability analysis to demonstrate the design and implementation of the TOE does not contain any obvious flaws.	O.VULNERABILITY_ANALYSIS addresses this concern by requiring a vulnerability analysis be performed in conjunction with testing that goes beyond functional testing. This objective provides a measure of confidence that the TOE does not contain security flaws that may not be identified through functional testing. While these testing activities are a necessary activity for successful completion of an evaluation, this testing activity does not address the concern that the TOE continues to operate correctly and enforce its security policies once it has been fielded. Some level of testing must be available to end users to ensure the TOE's security mechanisms continue to operator correctly once the TOE is fielded.
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.	O.RESIDUAL_INFORMATION The TOE will ensure that any information contained in a protected resource within its Scope of Control is not released when the resource is reallocated.	O.RESIDUAL_INFORMATION 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.

Threat/Policy	Objectives Addressing the Threat/Policy	Rationale
T.TSF_COMPROMISE A user or process may cause, through an unsophisticated attack, TSF data, or executable code to be inappropriately accessed (viewed, modified, or deleted).	O.RESIDUAL_INFORMATION The TOE will ensure that any information contained in a protected resource within its Scope of Control is not released when the resource is reallocated.	O.RESIDUAL_INFORMATION is necessary to mitigate this threat, because even if the security mechanisms do not allow a user to view TSF data, if TSF data were to reside inappropriately in a resource that was made available to a user, that user would be able to view the TSF data without authorization.
	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.PARTIAL_SELF_PROTECTION ensures the TOE is capable of protecting itself from attack.
	O.MANAGE The TOE will provide all the functions and facilities necessary to support the authorized administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.	O.MANAGE is necessary because an access control policy is specified to control access to TSF data. This objective is used to dictate who is able to view and modify TSF data, as well as the behavior of TSF functions.

Threat/Policy	Objectives Addressing the Threat/Policy	Rationale
T.UNAUTHORIZED_ACCESS A user may gain unauthorized access to user data for which they are not authorized according to the TOE security policy.	O.MEDIATE The TOE must protect user data in accordance with its security policy.	O.MEDIATE ensures that all accesses to user data are subject to mediation, unless said data has been specifically identifies as public data. The TOE requires successful authentication to the TOE prior to gaining access to any controlled-access content. By implementing strong authentication to gain access to these services, an attacker's opportunity to conduct a man-in-the-middle and/or password guessing attack successfully is greatly reduced. Lastly, the TSF will 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 administrator. This feature ensures that no other user can modify the information flow policy to bypass the intended TOE security policy.
	O.ACCESS_HISTORY The TOE will store and retrieve information (to authorized users) related to previous attempts to establish a session.	O.ACCESS_HISTORY is important to mitigate this threat because it ensures the TOE will be able to store and retrieve the information that will advise the user of the last successful login attempt and performed actions without their knowledge.

Threat/Policy	Objectives Addressing the Threat/Policy	Rationale
T.UNIDENTIFIED_ACTIONS Failure of the authorized administrator to identify and act upon unauthorized actions may occur.	O.ADMIN_GUIDANCE The TOE will provide administrators with the necessary information for secure management.	The threat of an authorized administrator failing to know about malicious audit events produces the objectives of the authorized administrator having the facilities and knowing how to use them (O.ADMIN_GUIDANCE).
	O.MANAGE The TOE will provide all the functions and facilities necessary to support the authorized administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.	The threat of an authorized administrator failing to know about malicious audit events produces the objectives of the authorized administrator having the capability to use the mechanisms (O.MANAGE) to review audit records.
P.ACCOUNTABILITY The authorized users of the TOE shall be held accountable for their actions within the TOE.	O.AUDIT_GENERATION The TOE will provide the capability to detect and create records of security relevant events associated with users.	O.AUDIT_GENERATION addresses this policy by providing the authorized administrator with the capability of configuring the audit mechanism to record the actions of a specific user, or review the audit trail based on the identity of the user. Additionally, the administrator's ID is recorded when any security relevant change is made to the TOE (e.g., access rule modification, start-stop of the audit mechanism, establishment of a trusted channel, etc.).

Threat/Policy	Objectives Addressing the Threat/Policy	Rationale
	O.TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to the TOE.	O.TOE_ACCESS supports this policy by requiring the TOE to identify and authenticate all authorized users prior to allowing any TOE access or any TOE mediated access on behalf of those users.
P.CRYPTOGRAPHY The TOE will use NIST FIPS 140-1/2 validated crypto modules for cryptographic services implementing NIST-approved security functions and random number generation services used by cryptographic functions.	O.CRYPTOGRAPHY All cryptographic-based security components used to protect sensitive information must be FIPS 140-2 validated. These services will provide confidentiality and integrity protection of TSF data while in transit. O.RESIDUAL_ INFORMATION The TOE will ensure that any information contained in a protected resource within its Scope of Control is not released when the resource is reallocated.	O.CRYPTOGRAPHY satisfies this policy by requiring the TOE to implement NIST FIPS validated cryptographic services. These services will provide confidentiality and integrity protection of TSF data while in transit to remote parts of the TOE. O.RESIDUAL_INFORMATIO N satisfies this policy by ensuring that cryptographic data are cleared according to FIPS 140-1/2.
P.ROLES The TOE shall provide an authorized administrator role for secure administration of the TOE. This role shall be separate and distinct from other authorized users.	O.ADMIN_ROLE The TOE will provide authorized administrator roles to isolate administrative actions.	The TOE has the objective of providing an authorized administrator role for secure administration. The TOE may provide other roles as well, but only the role of authorized administrator is required (O.ADMIN_ROLE).

6.2 Rationale for the Security Objectives and Security Functional Requirements for the Environment

Table 12 Rational for IT Environmental Objectives

Assumption	Environmental Objective Addressing the Assumption	Rationale
A.NO_EVIL Administrators are non-hostile, appropriately trained, and follow all administrator guidance.	OE.NO_EVIL Sites using the TOE shall ensure that authorized administrators are non-hostile, are appropriately trained and follow all administrator guidance.	All authorized administrators are trustworthy individuals, having background investigations commensurate with the level of data being protected, have undergone appropriate admin training, and follow all admin guidance.
A.NO_GENERAL_PURPOSE There are no general-purpose computing or storage repository capabilities (e.g., compilers or user applications) available on *Web Server*, other than those services necessary for the operation, administration and support of the *Web Server*.	OE.NO_GENERAL_PURPOSE There will be no general-purpose computing capabilities (e.g., compilers or user applications) available on DMBS servers, other than those services necessary for the operation, administration and support of the *Web Server*.	The *Web Server* must not include any general-purpose commuting or storage capabilities. This will protect the TSF data from malicious processes.
A.OS_PP_VALIDATED It is assumed that the underlying OS has been validated against an NSA sponsored OS PP of at least Basic Robustness.	OE.OS_PP_VALIDATED The underlying OS has been validated against an NSA sponsored OS PP of at least Basic Robustness.	The underlying OS must be validated to at least basic robustness to ensure it provides an appropriate level of protection for the *Web Server*.
A.PHYSICAL Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the IT environment.	OE.PHYSICAL Physical security will be provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.	The TOE, the TSF data, and protected user data is assumed to be protected from physical attack (e.g., theft, modification, destruction, or eavesdropping). Physical attack could include unauthorized intruders into the TOE environment, but it does not include physical destructive actions that might be taken by an individual that is authorized to access the TOE environment.

6.3 Rationale for TOE Security Requirements

Table 13 Rationale for TOE Security Requirements

Objective	Requirements Addressing the Objective	Rationale
O.ACCESS_HISTORY The TOE will store and retrieve information (to authorized users) related to previous attempts to establish a session.	FTA_TAH_(EXT).1	The TOE must be able to display information about previous unauthorized login attempts and the number times the login was attempted every time the user logs into their account. The TOE must also be able to store the last successful authorized login. This information will include the date and time of the attempts. When appropriately displayed, this will allow the user to detect if another user is attempting to access their account. These records should not be deleted until after the user has been notified of their access history. (FTA_TAH_(EXT).1)
O.ADMIN_GUIDANCE The TOE will provide administrators with the necessary information for secure management.	ALC_DEL.1	ALC_DEL.1 ensures that the administrator is provided documentation that instructs them how to ensure the delivery of the TOE, in whole or in parts, has not been tampered with or corrupted during delivery. This requirement ensures the administrator has the ability to begin their TOE installation with a clean (e.g., malicious code has not been inserted once it has left the developer's control) version of the TOE, which is necessary for secure management of the TOE.

Objective	Requirements Addressing the Objective	Rationale
	AGD_PRE.1	AGD_PRE.1ensures 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 Preparative procedures (PRE) documentation ensures that once the administrator has followed the Preparative procedures the result is a TOE in a secure configuration.
	AGD_OPE.1	AGD_OPE.1 mandates the developer provide 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, how to configure the TOE's rule set and the implications of any dependencies of individual rules. The documentation also provides a description of how to setup and review the auditing features of the TOE. The guidance must show the administrator how to use the functionality available, review the results of any tests and/or alerts, and act accordingly. AGD_OPE.1 will also provide guidance on security that is common to both administrators and non-administrators (e.g., password management guidelines).

Objective	Requirements Addressing the Objective	Rationale
O.ADMIN_ROLE The TOE will provide authorized administrator roles to isolate administrative actions.	FMT_SMR.1	The TOE will establish, at least, an authorized administrator role. The ST writer may choose to specify more roles. The authorized administrator will be given privileges to perform certain tasks that other users will not be able to perform. These privileges include, but are not limited to, access to audit information and security functions. (FMT_SMR.1)
O.AUDIT_GENERATION The TOE will provide the capability to detect and create records of security relevant events associated with users.	FAU_GEN.1-NIAP-0410	FAU_GEN.1-NIAP-0410 defines the set of events that the TOE must be capable of recording. This requirement ensures that the administrator has the ability to audit any security relevant events that takes place in the TOE. This requirement also defines the information that must be contained in the audit record 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_(EXT).2	FAU_GEN_(EXT).2 ensures that the audit records associate a user with the auditable event. In the case of authorized users, the association is accomplished with the user ID.
	FAU_SEL.1-NIAP-0407	FAU_SEL.1-NIAP-0407 allows the administrator to configure which auditable events will be recorded in the audit trail. This provides the administrator with the flexibility in recording only those events that are deemed necessary by site policy, thus reducing the amount of resources consumed by the audit mechanism.

Objective	Requirements Addressing the Objective	Rationale
O.CONFIGURATION_IDENTIFICATION The configuration of the TOE is fully identified in a manner that will allow implementation errors to be identified and corrected with the TOE being redistributed promptly.	ALC_CMC.2 ALC_CMS.2	ALC_CMC.2 and ALC_CMS.2 addresses this objective by requiring that there be a unique reference for the TOE, and that the TOE is labeled with that reference. It also requires that there be a CM system in place, and that the configuration items that comprise the TOE are uniquely identified. This provides a clear identification of the composition of the TOE.
	ALC_FLR.2	ALC_FLR.2 addresses this objective by requiring that there be a mechanism in place for identifying flaws subsequent to fielding, and for distributing those flaws to entities operating the system.
O.CRYPTOGRAPHY All cryptographic-based security components used to protect sensitive information must be FIPS 140-2 validated. These services will provide confidentiality and integrity protection of TSF data while in transit.	FCS_BCM_(EXT).1 ADV_ARC.1 FTA_SSL.3 FTP_ITC.1	FCS_BCM_(EXT).1 This requirement specify the use of NIST FIPS validated SSL and TLS. ADV_ARC.1 defines the architecture that limits the non-bypassability to the cryptographic modules. It also captures the application aspects of domain separation. FTA_SSL.3 has automatic session locking and termination, either initiated by the TSF, or a web user. FTP_ITC.1 requires the TOE to initiate the trusted channel, which ensures that the TOE has established a communication path with an authorized IT entity and not some other entity pretending to be an authorized IT entity

Objective	Requirements Addressing the Objective	Rationale
O.DOCUMENTED_DESIGN The design of the TOE is adequately and accurately documented.	ADV_FSP.2	ADV_FSP.2 requires that the interfaces to the TOE be documented and specified.
	ADV_TDS.1	ADV_TDS.1 requires the design of the TOE be documented and specified and that said design be shown to correspond to the interfaces.
O.MANAGE The TOE will provide all the functions and facilities necessary to support the authorized administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.	FMT_MOF.1	FMT_MOF.1 requires that the ability to use particular TOE capabilities be restricted to the administrator.
	FMT_MSA.1	FMT_MSA.1 requires that the ability to perform operations on security attributes be restricted to particular roles.
	FMT_MSA_(EXT).3	FMT_MSA_(EXT).3 requires that default values used for security attributes are restrictive.
	FMT_MTD.1	FMT_MTD.1 requires that the ability to manipulate TOE content is restricted to administrators.
	FMT_REV.1(1) FMT_REV.1(2)	FMT_REV.1 restricts the ability to revoke attributes to the administrator.
	FMT_SMF.1	FMT_SMF.1 identifies the management functions that are available to the authorized administrator.
	FMT_SMR.1	FMT_SMR.1 defines the specific security roles to be supported.

Objective	Requirements Addressing the Objective	Rationale
O.MEDIATE The TOE must protect user data in accordance with its security policy.	FDP_ACC.1	The FDP requirements were chosen to define the policies, the subjects, objects, and operations for how and when mediation takes place in the TOE. FDP_ACC.1 defines the Access Control policy that will be enforced on a list of subjects acting on the behalf of users attempting to gain access to a list of named objects. All the operation between subject and object covered are defined by the TOE's policy.
	FDP_ACF.1-NIAP-0407	FDP_ACF.1-NIAP-0407 defines the security attribute used to provide access control to objects based on the TOE's access control policy.
	FIA_UAU.1	FIA_UAU.1 ensures that the user is authenticated before any action will be allowed on behalf of that user.
	FIA_UID.1	FIA_UID.1 ensures that the user is identified before any action will be allowed on behalf of that user.
O.PARTIAL_FUNCTIONAL_TEST The TOE will undergo some security functional testing that demonstrates the TSF satisfies	ATE_COV.1	ATE_COV.1 requires that there be a correspondence between the tests in the test documentation and the TSF as described in the functional specification.

Objective	Requirements Addressing the Objective	Rationale
some of its security functional requirements.	ATE_FUN.1	ATE_FUN.1 requires that the developer provide test documentation for the TOE, including test plans, test procedure descriptions, expected test results, and actual test results. These need to identify the functions tested, the tests performed, and test scenarios. There require that the developer run those tests, and show that the expected results were achieved.
	ATE_IND.2	ATE_IND.2 requires that the evaluators test a subset of the TSF to confirm correct operation, on an equivalent set of resources to those used by the developer for testing. These sets should include a subset of the developer run tests.
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.	ADV_ARC.1	The architecture description as required by ADV_ARC.1 will ensure the TSF provides a domain that protects itself from untrusted users. If the TSF cannot protect itself it cannot be relied upon to enforce its security policies.
O.RESIDUAL_INFORMATION The TOE will ensure that any information contained in a protected resource within its Scope of Control is not released when the resource is reallocated.	FDP_RIP.1	FDP_RIP.1 is used to ensure the contents of resources are not available to subjects other than those explicitly granted access to the data.
O.TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to the TOE.	FIA_ATD.1	FIA_ATD.1 defines the attributes of users, including a user ID that is used by the TOE to determine a user's identity and enforce what type of access the user has to the TOE.

Objective	Requirements Addressing the Objective	Rationale
	FTA_MCS.1	FTA_MCS.1 ensures that users may only have a maximum of a specified number of active sessions open at any given time.
	FTA_TSE.1	FTA_TSE.1 allows the TOE to restrict access to the TOE based on certain criteria.
O.VULNERABILITY_ANALYSIS The TOE will undergo some vulnerability analysis to demonstrate the design and implementation of the TOE does not contain any obvious flaws.	AVA_VAN.2	The AVA_VAN.2 component provides the necessary 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 systematic search for potential vulnerabilities in all the TOE deliverables. For those vulnerabilities that are not eliminated, a rationale must be provided that describes why these vulnerabilities cannot be exploited by a threat agent with a low attack potential, which is in keeping with the desired assurance level of this TOE. As with the functional testing, a key element is this component is that an independent assessment of the completeness of the developer's analysis is made, and more importantly, an independent vulnerability analysis coupled with testing of the TOE is performed. This component provides the confidence that security flaws do not exist in the TOE that could be exploited by a threat agent or moderate (or lower) attack potential to violate the TOE's security policies.

The following table includes the rationale for the IT Environment Requirements. These Environmental Objectives (OE) are non-IT objectives except for OE.OS_PP_VALIDATED which is required to meet TOE requirements.

Table 14 Rationale for IT Environment Requirements

Environmental Objective	Requirements Addressing the Objective	Rationale
OE.NO_EVIL Sites using the TOE shall ensure that authorized administrators are non-hostile, are appropriately trained and follow all administrator guidance.	N/A	This objective does not contain any IT security requirements because it is a non-IT related objective. Thus, the CC does not mandate it map to any requirements.
OE.NO_GENERAL_ PURPOSE There will be no general-purpose computing capabilities (e.g., compilers or user applications) available on Web servers, other than those services necessary for the operation, administration and support of the *Web Server*.	N/A	This objective does not contain any IT security requirements because it is a non-IT related objective. Thus, the CC does not mandate it map to any requirements.
OE.OS_PP_VALIDATED The underlying OS has been validated against an NSA sponsored OS PP of at least Basic Robustness.	FIT_PPC_(EXT).1	FIT_PPC_(EXT).1 states the underlying OS must have been validated against an OS PP of at least basic robustness. This will ensure that all the SFR that are required by the TSF have been evaluated. The functions to be supported using the validated operating system are listed in section 2.2.
OE.PHYSICAL Physical security will be provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.	N/A	This objective does not contain any IT security requirements because it is a non-IT related objective. Thus, the CC does not mandate it map to any requirements.

6.4 Rationale for Assurance Requirements

This protection profile is developed at the basic robustness level. The assurance requirements are those recommended in instruction 4 from the *Consistency Instruction Manual for Development of US Government Protection Profiles for Use in Basic Robustness Environments*, Version 3.0, dated 1 February 2005.

Flaw Remediation is the only requirement not included in any EAL level because it does not add any assurance to the current system, but to subsequent releases. Therefore, the PPRB decided to augment EAL2 with ALC_FLR.2 to instruct the vendors on proper flaw remediation techniques.

6.5 Rationale for Satisfying all Dependencies

Table 15 Functional Requirement Dependencies

Requirement	Dependency	Satisfied
FAU_GEN.1-NIAP-0410	FPT_STM.1	The OS in the IT environment will satisfied this requirement. .
FAU_GEN_(EXT).2	FAU_GEN.1-NIAP-0410 FIA_UID.1	Satisfied
FAU_SEL.1-NIAP-0407	FAU_GEN.1-NIAP-0410 FMT_MTD.1	Satisfied
FCS_BCM_(EXT).1	No Dependencies Specified	
FDP_ACC.1	FDP_ACF.1-NIAP-0407	Satisfied
FDP_ACF.1-NIAP-0407	FDP_ACC.1 FMT_MSA.3	The dependency on FMT_MSA.3 is satisfied by FMT_MSA_(EXT).3.
FDP_RIP.1	None	N/A
FIA_ATD.1	None	N/A
FIA_UAU.1	FIA_UID.1	Satisfied
FIA_UID.1	None	N/A
FMT_MOF.1	FMT_SMF.1[4] FMT_SMR.1	Satisfied

[4] This list of dependency has been modified per CCIMB Interpretation 065.

Requirement	Dependency	Satisfied
FMT_MSA.1	[FDP_ACC.1 or FDP_IFC.1] FMT_SMF.1 [4] FMT_SMR.1	Dependency satisfied by FDP_ACC.1.
FMT_MSA_(EXT).3	FMT_MSA.1 FMT_SMR.1	Satisfied
FMT_MTD.1	FMT_SMF.1 FMT_SMR.1	Satisfied
FMT_REV.1(1)	FMT_SMR.1	Satisfied
FMT_REV.1(2)	FMT_SMR.1	Satisfied
FMT_SMF.1	None	N/A
FMT_SMR.1	FIA_UID.1	Satisfied
FTA_MCS.1	FIA_UID.1	Satisfied
FTA_SSL.3	None	N/A
FTA_TAH_(EXT).1	None	N/A
FTA_TSE.1	None	N/A
FTP_ITC.1	None	N/A

Table 16 Functional Requirement Dependencies for IT Environment

Requirement	Dependency	Satisfied
FIT_PPC_(EXT).1	None	N/A

Table 17 Assurance Requirement Dependencies

Requirement	Dependency	Satisfied
ADV_ARC.1	**ADV_FSP.1** **ADV_TDS.1**	**Yes**

Requirement	Dependency	Satisfied
ADV_FSP.2	**ADV_TDS.1**	**Yes**
ADV_TDS.1	**ADV_FSP.2**	**Yes**
AGD_OPE.1	**ADV_FSP.1**	**Yes**
AGD_PRE.1	**None**	**N/A**
ALC_CMC.2	**ALC_CMS.1**	**Yes**
ALC_CMS.2	**None**	**N/A**
ALC_DEL.1	**None**	**N/A**
ALC_FLR.2	**None**	**N/A**
ATE_COV.1	**ADV_FSP.2** ATE_FUN.1	**Yes**
ATE_FUN.1	**ATE_COV.1**	**Yes**
ATE_IND.2	**ADV_FSP.2** **AGD_OPE.1** **AGD_PRE.1** **ATE_COV.1** **ATE_FUN.1**	**Yes**
AVA_VAN.2	**ADV_ARC.1** **ADV_FSP.1** **ADV_TDS.1** **AGD_OPE.1** **AGD_PRE.1**	**Yes**

6.6 Rationale for Extended Requirements

Table 18 presents the rationale for the inclusion of the extended functional and assurance requirements found in this PP. The extended requirements that are included as NIAP interpretations do not require a rationale for their inclusion per CCEVS management.

Table 18 Rationale for Extended Requirements

Extended Requirement	Identifier	Rationale

Extended Requirement	Identifier	Rationale
FCS_BCM_(EXT).1	Baseline Cryptographic Module	The CC does not provide a means to specify the use of NIST FIPS validated cryptography as a baseline for the cryptographic module. FCS_BCM_(EXT).1 is an extended component that specifies FIPS PUB 140-2 validated against SSL version 3.0 or TLS version 1.0 or later versions.
FAU_GEN_(EXT).2	User identity association	This requirement was needed to replace FAU_GEN.2.1-NIAP-0410 because this PP does not require the TOE to implement a user identity. It does require the TOE to implement a user identity to satisfy the access control policy. Therefore, this extended requirement was created to allow the audit function to use the user identity.
FTA_TAH_(EXT).1	TOE Access History	This requirement has been modified so that the TOE has the capability to be able to store and retrieve the access history and display it.
FMT_MSA_(EXT).3	Static attribute initialization	The CC does not allow the PP author to specify restrictive values that are not modifiable. This extended requirement eliminates the element FMT_MSA.3.2 from the component FMT_MSA.3 and makes the component more secure by requiring the security attributes of the objects on creation to be restrictive and not allowing any user to be able of override the restrictive default values.

Table 19 Rationale for Environmental Requirements

Environmental Requirement	Identifier	Rationale
FIT_PPC_(EXT).1	IT Environment Protection Profile Compliance	This requirement is necessary to ensure the TOE will be running on an OS that is at least as robust as the TOE itself. The functions to be supported using the validated operating system are listed in section 2.2.

7 APPENDICES

The following sections are the appendices for this Protection Profile.

A REFERENCES

1) *Common Criteria for Information Technology Security Evaluation,* CCIB-98-031 Version 2.1, August 1999.

2) Department of Defense Directive 8500.1, "Information Assurance," October 24, 2002.

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

4) *Information Assurance Technical Framework*, Version 3.0, September 2000.

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

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

7) *Internet Engineering Task Force The TLS Protocol Version 1.0,* RFC 2246, January 1999

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

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

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

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

12) *Internet Engineering Task Force, Hypertext Transfer Protocol -- HTTP/1.1,* RFC 2616, June 1999.

13) *Internet Engineering Task Force, HTTP Authentication: Basic and Digest Access Authentication*, RFC 2617, June 1999.

14) *Internet Engineering Task Force, Upgrading to TLS Within HTTP/1.1*, RFC 2817, May 2000.

15) *Internet Engineering Task Force, HTTP Over TLS,* RFC 2818, May 2000.

16) *Department of Defense Instruction, Information Assurance Implementation Draft No. 8500.bb*, September 2001.

17) *The AES Cipher Algorithm and Its Use with IPSec* <draft-ietf-ipsec-ciph-aes-cbc.03.txt>, Internet draft, November 2001.

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

B GLOSSARY

This profile uses a number of terms in specific senses. The following sections provide definitions of the terms that are used in this PP.

Access Control— A means of restricting access to objects based on the identity of subjects.

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

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

Authentication — Security measure that verifies a claimed identity.

Authentication data — Information used to verify a claimed identity.

Authorization — Permission, granted by an entity authorized to do so, to perform functions and access 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 Security Policy — A precise specification of the security rules under which a crypto must operate, including the rules derived from the requirements of this PP and additional rules imposed by the vendor.

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

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. A common representation is the full or abbreviated name of that user or a pseudonym.

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

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.

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

Realm – A realm refers a domain that the content provided or users are interested in or are communicating about.

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. There are three levels of robustness:

Basic: Security services and mechanisms that equate to good commercial practices. Basic robustness equates to EAL-2 plus; ALC_FLR.2 (Flaw Remediation), and AVA_MSU.1 (Misuse-Examination Guidance) as defined in CCIB-98-028, Part 3, Version 2.0

Medium: Security services and mechanisms that provide for layering of additional safeguards above good commercial practices. Medium robustness equates to EAL-4 plus; ALC_FLR.2 (Flaw Remediation); ADV_IMP.2; ADV_INT.1; ATE_DPT.2; and AVA_VLA.3 (Moderately Resistant Vulnerability Analysis) as defined in CCIB-98-028, Part 3, Version 2.0. If cryptographic functions are included in the TOE, then the PP should be augmented with AVA_CCA_(EXT).2 as documented in the Protection Profile Medium Robustness Consistency Guidance.

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 attribute — TSF data associated with subjects, objects, and users that is used for the enforcement of the TSP.

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

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.

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

C ACRONYMS

ACL	Access Control List
CAPP	Controlled Access Protection Profile
CC	Common Criteria for Information Technology Security Evaluation
CCIMB	Common Criteria Interpretation Management Board
DoD	Department of Defense
EAL	Evaluation Assurance Level
ESP	Encapsulating Security Payload
FIPS PUB	Federal Information Processing Standard Publication
FTP	File Transfer Protocol
GIG	Global Information Grid
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP with a Secure Socket Layer (SSL)
I&A	Identification and Authentication
IP	Internet Protocol
IT	Information Technology
N/A	Not Applicable
NIAP	National Information Assurance Partnership
NIST	National Institute of Standards and Technology
NSA	National Security Agency
PP	Protection Profile
SF	Security Function
SFP	Security Function Policy
SSL	Secure Socket Layer
ST	Security Target
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TOE	Target of Evaluation
TSC	TSF Scope of Control
TSE	TOE Security Environment
TSF	TOE Security Function
TSP	TOE Security Policy
URI	Universal Resource Identifier
URL	Uniform Resource Locator
WWW	World Wide Web

D ROBUSTNESS ENVIRONMENT CHARACTERIZATION

D.1 General Environmental Characterization

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 the next section, these two environmental factors will be related to the robustness required for selection of an appropriate TOE.

D.1.1 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 using organization. For example, in the DoD low-value data might be equivalent to data marked "For Official Use Only", 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.

D.1.2 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 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 was 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.

D.1.3 Selection of Appropriate Robustness Levels

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 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.

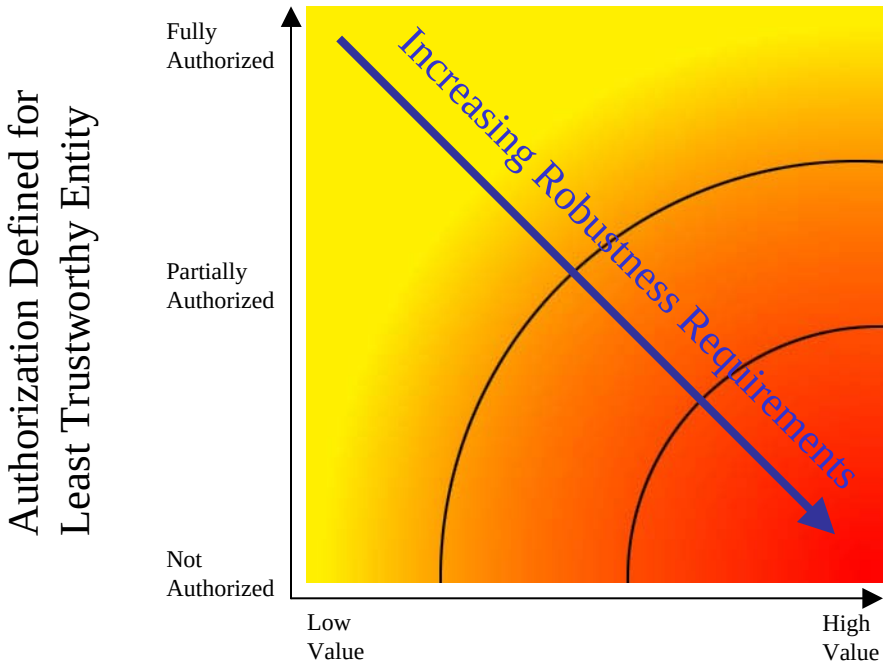
The second case is a TOE that processes high-value (e.g., classified) information. The organization requires that the TOE be stand-alone, and that every user with physical and logical access to the TOE undergo an investigation so that they are authorized to the highest value data on the TOE. Because 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 demonstrated 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. The following chart 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.

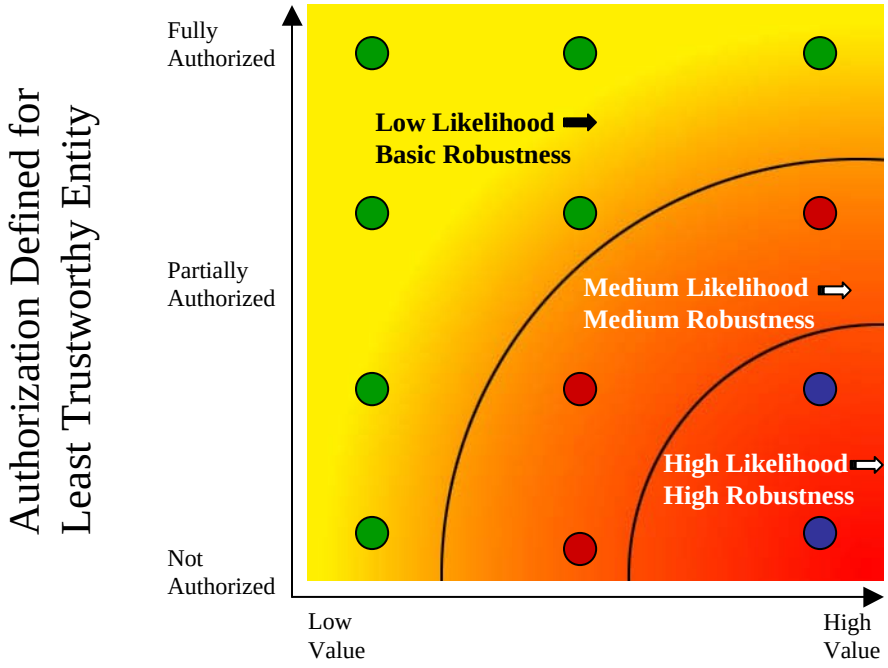
As depicted in the following figure, the robustness of the TOEs required in each environment steadily increases as one goes from the upper left of the chart to the lower right; this corresponds to the need to counter increasingly likely attack attempts by the least trustworthy entities in the environment. Note that the shading of the chart is intended to reflect- 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.

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 a set of environments where the likelihood of attempted compromise is roughly similar. This is graphically depicted in the following chart.



In this second representation of environments and the robustness plane below, 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 of this PP, the targeted threat level for a Basic robustness TOE is characterized. This information is provided to help organizations using this PP -ensure that the functional requirements specified by this Basic robustness PP are appropriate for their intended application of a compliant TOE.


Authorization Defined for Least Trustworthy Entity
Fully Authorized
Partially Authorized
Not Authorized
Low Likelihood
Basic Robustness
Medium Likelihood
Medium Robustness
High Likelihood
High Robustness
Low Value
High Value
Highest Value of Resources Associated with the TOE

E REFINEMENTS

This section contains refinements where text was omitted. Omitted text is shown as bold text within parenthesis. The actual text of the functional requirements as presented in Section 5 has been retained.

FAU_SEL.1.1-NIAP-0407 **Refinement**: The TSF shall **(be able to) allow only the administrator** to include or exclude auditable events from the set of audited events based on the following attributes:

a) *user identity*,

b) *event type*,

c) *object identity*,

d) [selection: "subject identity", "host identity", "none"];

e) [success of auditable security events;

f) failure of auditable security events; and

g) [selection: [assignment: list of additional criteria that audit selectivity is based upon], "no additional criteria"].]

FMT_MSA.1.1 **Refinement**: The TSF shall enforce the [Access Control policy] to restrict the ability to [*manage*] **all** the security attributes to [authorized administrators].

Application Note: The ST author should ensure that all attributes identified in FIA_ATD.1 are adequately managed and protected.