US GOVERNMENT PROTECTION PROFILE

ANTI-VIRUS APPLICATIONS FOR WORKSTATIONS IN BASIC ROBUSTNESS ENVIRONMENTS

INFORMATION
ASSURANCE
DIRECTORATE

6 January 2005
Version 1.0

FORWARD

This publication, "*US Government Protection Profile for Anti-Virus Applications for Workstations in Basic Robustness Environments"* is issued by the Information Assurance Directorate of the National Security Agency. This protection profile is based on the *Common Criteria for Information Technology Security Evaluations*, Version 2.2.

Further information, including the status and updates of this protection profile can be found on the internet at: http://www.iatf.net/protection_profiles/index.cfm.

Comments on this document should be directed to: ppcomments@iatf.net. The comments should include the title of the document, the page, the section number, and paragraph number, detailed comment and recommendations.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

{This page intentionally left blank}

1 INTRODUCTION

This Protection Profile (PP) is sponsored by the Defense Information Systems Agency (DISA) to provide secure anti-virus services for workstations, and is intended for the following uses:

1. For vendors and security evaluators, this PP defines the requirements that must be addressed by specific products as documented in vendor Security Targets (STs).
2. For system integrators, this PP is useful in identifying areas that need to be addressed to provide secure system solutions.

1.1 IDENTIFICATION

Title: U.S. Government Protection Profile for Anti-Virus Applications for Workstations in Basic Robustness Environments
Sponsor: Defense Information Systems Agency (DISA)
Developer: ARTEL, Inc. and COACT Inc.
CC Version: Common Criteria (CC) Version 2.2, and applicable international and NIAP interpretations as of 23 November 2004.
Registration: <to be provided upon registration>
Protection Profile Version: Version 1.0, dated 6 January 2005.
Evaluation Assurance Level: Basic Robustness Assurance consisting of: ACM_CAP.2, ADO_DEL.1, ADO_IGS.1, ADV_FSP.1, ADV_HLD.1, ADV_RCR.1, AGD_ADM.1, AGD_USR.1, ALC_FLR.2, ATE_COV.1, ATE_FUN.1, ATE_IND.2, AVA_MSU.1, AVA_SOF.1, AVA_VLA.1
Keywords: Basic Robustness Environments, Anti-Virus.

1.2 PROTECTION PROFILE OVERVIEW

This PP specifies the minimum-security requirements for Anti-Virus Applications (i.e., the Target of Evaluation (TOE)) used on workstations in the US Government in Basic Robustness Environments. The Anti-Virus Application provides protection against viruses coming into the workstation from network connections and/or removable media, and is considered sufficient protection for environments where the likelihood of an attempted compromise is low. The target robustness level of "basic" is discussed in Section 3.0 of this PP. STs claiming compliance may consist of software only.

The PP defines the requirements for a general-purpose Anti-Virus Application that may be used in a variety of systems. Relative to these requirements the PP includes:

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

1.3 THE TOE AS A COMPONENT OF A SYSTEM

The PP includes security requirements associated with an Anti-Virus Application as part of a larger system, (e.g., running on top of an operating system). As a component of these systems the TOE must work in concert with other components to provide system security services. While the PP includes requirements for component security functions to support system security services, it doesn't specify protocols or standards for compliance.

1.4 COMMON CRITERIA CONFORMANCE

1.4.1 Conformance Claim

This Protection Profile is Common Criteria Part 2 Extended and Common Criteria Part 3 conformant, with U.S. DoD Basic Robustness Assurance (as defined in the *Consistency Instruction Manual For development of US Government Protection Profiles (PP) For use in Basic Robustness Environments* [BRCIM]). This PP is also conformant with CEM Supplement: ALC_FLR – Flaw Remediation.

1.4.2 STs Claiming Conformance to this PP

An ST claiming conformance to this PP must define the TOE in the ST to include all SFRs levied against the TOE in the PP (specified in Section 5.1) without reliance to its environment. SFRs levied against the IT Environment in the PP (specified in section 5.2) may be implemented in the ST TOE.

1.5 PROTECTION PROFILE CONTENTS AND ORGANIZATION

Section 1 introduces this PP document through an overview, a statement of Common Criteria Conformance, and a description of this PP organization.

Section 2 describes the TOE and the environment. This section also provides an overview of the security functionality provided upon conformance with this PP.

Section 3 provides informative introductory text to help the reader gain an understanding of the various robustness levels and more importantly how to determine the proper robustness level for a given system. Additionally, Section 3 discusses the characteristics of environments and threat levels appropriate for the TOE and specifies the TOE assumptions, threats, and organizational security policies.

Section 4 identifies the security objectives satisfied by the TOE and the TOE environment.

Section 5 specifies the functional and assurance requirements for the TOE and its IT environment.

Section 6 provides the rationale for the security objectives and the security requirements. The objectives rationale shows that the security objectives address the assumptions, threats and policies. The requirements rationale shows that the requirements meet the objectives and that all dependencies are satisfied. In addition, rationale is provided for the Strength of Function (SOF) and Assurance requirements.

Section 7 contains expansions of acronyms used throughout this PP.

Section 8 contains the references.

Section 9 provides a glossary of terms.

2 TOE DESCRIPTION

2.1 PRODUCT TYPE

The product type of the Target of Evaluation (TOE) described in this Protection Profile (PP) is an Anti-Virus application running on workstations (e.g., desktops and laptops), along with a management component to control and monitor execution of the Anti-Virus application. The TOE may be software only.

In general terminology, "virus" is used generically to refer to an entire suite of exploits of security vulnerabilities, such as worms and Trojan Horses. The same term is used more specifically to refer to exploits that replicate themselves. The term "Anti-Virus" typically refers to measures used to counter the entire suite of exploits, not just the more specific definition of virus. In this PP, virus is used to refer to a suite of exploits.

An Anti-Virus application scans content being introduced onto the workstation for viruses. The content may be introduced via removable media (e.g., CDs) inserted into the workstation or via incoming network traffic (e.g., HTML, e-mail attachments, FTP). Anti-Virus applications provide:

- Real-time scanning (to detect viruses as they are entering the system),
- On-demand scans (especially useful for scanning removable media), and
- Scheduled scans (backup mechanism in case a virus is introduced in a way that escaped detection).

Viruses may be file-based or memory-based (i.e., the virus itself does not have to be written to the workstation disk via the file system in order to execute – an example is CodeRed). To detect memory-based viruses, Anti-Virus applications may scan incoming network traffic or scan application memory space (or both). File-based scans must be able to detect viruses contained within compressed files.

Scanning is performed against "signatures" of known viruses. A signature is a known pattern indicative of a virus. To combat new viruses, vendors update and make available a file of signatures (often referred to as DAT files) on a frequent basis. The Anti-Virus application must be able to import updated signatures as necessary. A message digest is used to verify the integrity of the imported signature file on the individual workstations executing the Anti-Virus application.

When a file-based virus is detected, a configured action (or ordered list of actions) is performed to isolate and/or eliminate the virus. The actions available include:

- Clean the virus from the file,
- Quarantine the file,
- Rename the file,
- Delete the file, and
- No action (allow the virus to remain in the file).

When a memory-based virus is detected, the virus is prevented from further execution. The mechanism used to accomplish this is dependent on the type of scanning being performed.

Possible mechanisms include discarding incoming network traffic that contains the virus, or terminating a process that has the virus present in its memory space.

An alert message is generated on the screen of the workstation informing the user of the workstation about the virus and the action performed. This alert remains on the screen until acknowledged by the user (or the user ends the session).

In the past, new viruses have been known to propagate themselves to additional platforms via email. Some instances have used self-contained mail functionality. Conformant TOEs must prevent unauthorized processes (i.e., Trojan) from sending email (via SMTP) from the workstation.

Conformant TOEs will be used in Enterprise environments. To support this usage, centralized control and monitoring is required. A Central Administrator must be able to remotely configure the TOE on all network-attached workstations within the Central Administrator's domain. At a minimum, the configuration options that are only made available to the Central Administrator include:

- Configuration of the actions to be taken when file-based viruses are detected,
- Frequency of scheduled scans,
- Depth of scans (for compressed files), and
- File types to be included and/or excluded from scans.

Copies of all audits (including alert messages) from the network-attached workstations are sent to a central management system, where they can be reviewed by the Central Administrator. Audit buffers are provided on the workstations to account for temporary interruptions in connectivity between the workstation and central collection system.

An alert message is generated to the Central Administrator (if a session is active at the time the audit information is received by the central collection system) informing him/her about detection of a virus and the action performed. This alert remains on the screen until acknowledged by the Central Administrator (or the session is ended).

Workstations may not be network-attached (i.e., stand-alone). In those situations, the local administrator for the workstation assumes the privileges of the Central Administrator for that workstation.

The Central Administrator is able to electronically transfer signature files to the network-attached workstations in the domain. Stand-alone workstations depend on physical transfer of the signature files.

Signature files are expected to be updated frequently. The updates originate with the vendor of the Anti-Virus application, and distribution of the updates occurs in several stages.

In the first stage, the updates are securely transmitted from the vendor to the Enterprise (e.g., DISA). The mechanisms used for this stage are expected to vary depending on the needs of the Enterprise, and are outside the scope of this PP. Enterprises are encouraged to use strong mechanisms to verify both the source and content of the updates. Once received from the vendor, the Enterprise would be expected to validate the updates before making them available for the second stage.

In the second stage, the updates are made available to the central management systems supporting the Anti-Virus applications. The mechanisms for this distribution are defined by the Enterprise, and are outside the scope of this PP. Since this distribution occurs within the Enterprise, the mechanisms do not necessarily need to be as strong as those used for the first stage.

The third stage involves distribution from the central management system to the individual workstations. Secure communication paths are assumed to exist between the distributed components of the TOE. The strength of the mechanisms used for the secure communications paths are determined by the requirements of the environment in which the TOE is used.

2.2 TOE BOUNDARY

The TOE will be used on workstations in a trusted network configuration, as illustrated in Figure 2.1. The Firewall/Guard at the boundary of the trusted network represents one or more systems that perform protection services for the trusted network as a whole. It is assumed that protocols commonly used to transport viruses, such as SMTP, HTTP, and FTP, are screened at the Firewall/Guard function. This provides a "defense in depth" since the TOE (executing on the workstations) performs similar functions.

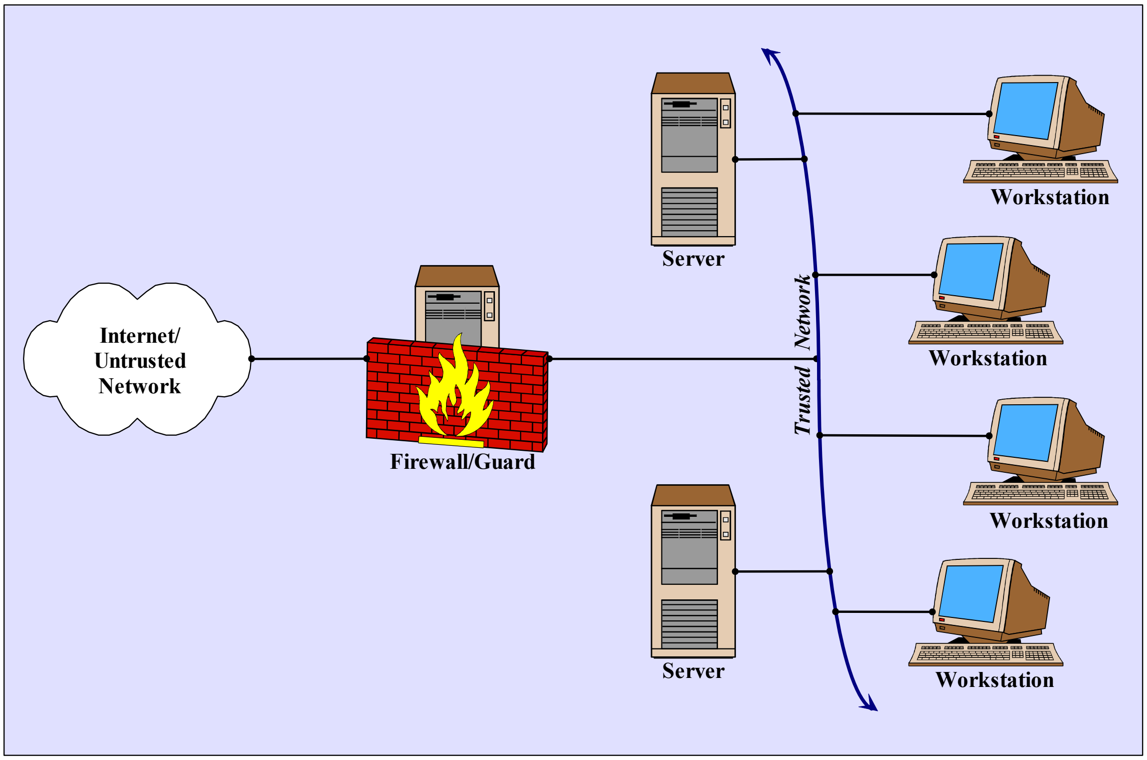

Figure 2.1 – Network Environment of the TOE

It is expected that Anti-Virus applications may be executing on both the servers (e.g., network attached storage, email servers, web servers) and workstations within the trusted network. This

PP does not address the servers; instead, it focuses on workstations. A single product is not required to address both workstations and servers because:

- Servers may utilize different operating systems than workstations,
- Vendors may provide different products for workstations and servers, and
- Servers typically do not have "local users" logged in on a directly-connected screen.

On the workstations (see Figure 2.2), the Anti-Virus application executes on top of the operating system to perform its scanning, reaction, and logging functions. The interfaces between the operating system, applications, and the Anti-Virus application are vendor and operating system specific, but in general terms may include:

- Interception of file system calls to scan files when they are created, modified, and/or opened;
- Interception of incoming network traffic to scan for memory-based virus attacks;
- Access to application process memory to scan for memory-based viruses; and
- Interception of outgoing network traffic to validate the source of SMTP traffic from the workstation.

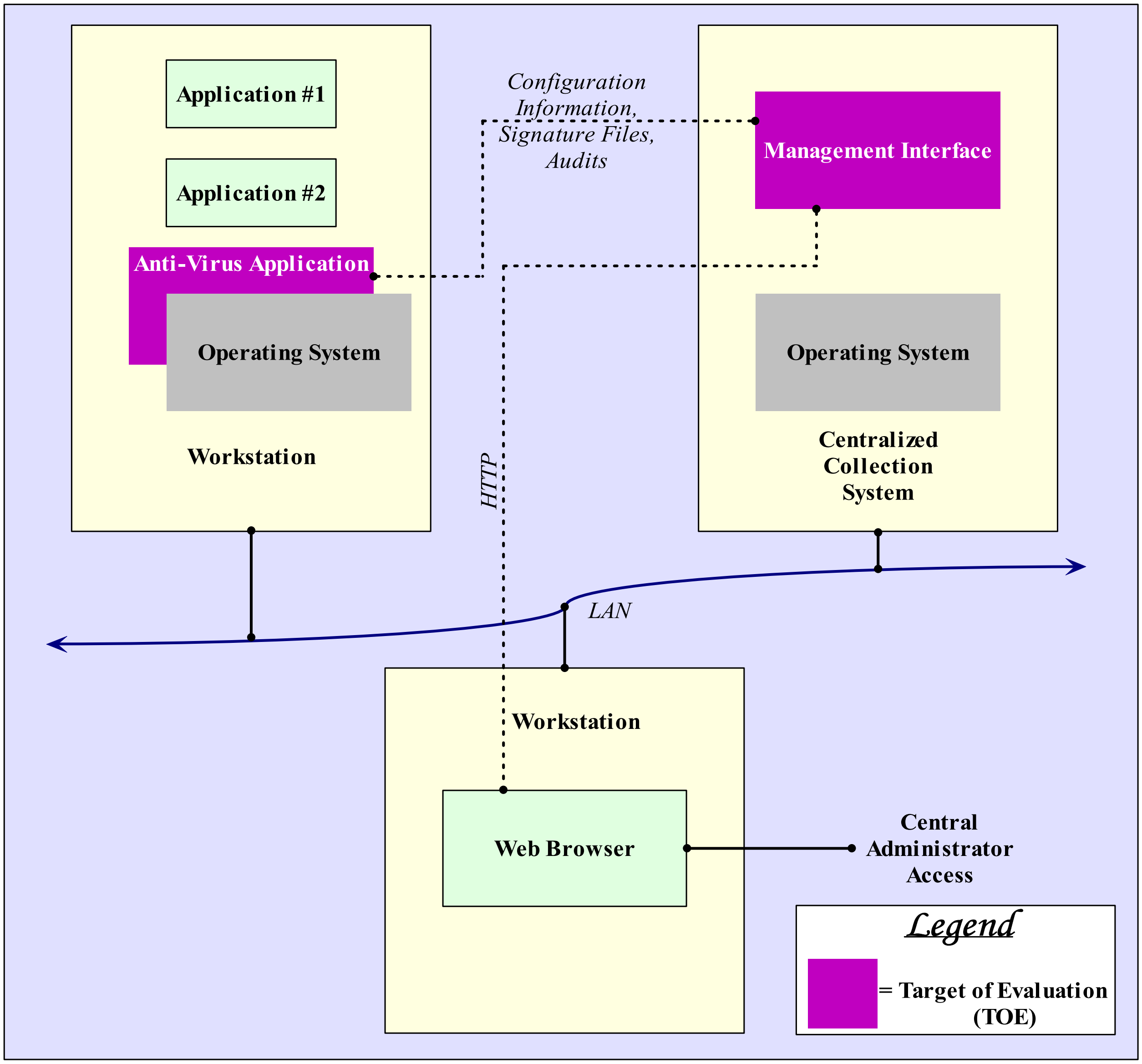

Figure 2.2 – TOE Boundary

The management functions of the Central Administrator for a conformant TOE may execute on a separate system from the portion of the TOE performing virus scanning on workstations. Access to those management functions may be remote via HTTP.

Figure 2.2 illustrates possible communication between the TOE components or between the TOE and the TOE users. It is not intended to be complete. For example, vendors may support browser access to a TOE executing on a workstation for remote configuration of the TOE.

2.3 SECURITY SERVICES

The functional security requirements can be categorized as follows:

1. Anti-Virus
2. Audit
3. Cryptographic Operations
4. Management
5. Protection of the TOE

2.3.1 Anti-Virus

The Anti-Virus services are described in section 2.1.

2.3.2 Audit

The audit services are described at a high level in section 2.1. The audit functionality required is to generate audits when security-relevant events occur, store the audit information on the local system, transmit the audit information to a central management system, generate alarms for designated events, and audit review.

Protection of audit data in the audit trail involves the TOE and the Operating System (OS). The TOE controls the insertion of audit events into the audit log and the deletion of audit events from the audit log. The OS provides basic file protection services for the audit log.

2.3.3 Cryptographic Operations

Integrity of the signature files is verified by a message digest calculated for the file.

2.3.4 Management

The management services are described in sections 2.1 and 2.4.

2.3.5 Protection of the TOE

Protection of the TOE is required to ensure the TOE security services are not bypassed or tampered with. The TOE and the OS cooperatively provide this service.

Between separate portions of the TOE, secure communication is provided by the IT Environment.

2.4 ROLES

This PP defines three roles:

1. Central Administrator – The Central Administrator controls the operation of all instances of the TOE under their authority. This role has the authority to:
   - Remotely manage operation of the TOE on workstations,
   - Schedule scans of existing files,
   - Manually invoke scans,
   - Control the minimum depth of scans,
   - Update virus signature files,
   - Receive alert notifications from the centralized management system,
   - Acknowledge alert notifications from the centralized management system, and
   - Review the TOE audit information in the centralized management system.

Application Note: When the workstation is stand-alone (i.e., not network-attached), the local administrator for the workstation assumes the privileges of the Central Administrator for that workstation. The Central Administrator privileges associated with the centralized management system do not apply to this scenario, and operation of the TOE must be administered locally.

2. Workstation User – The user utilizing the workstation. This role has the authority to:
   - Manually invoke scans,
   - Increase the depth of scans on manually invoked scans,
   - Receive alert notifications for events on the workstation being used,
   - Acknowledge alert notifications for events on the workstation being used, and
   - Review the TOE audit information on the workstation being used.

3. Network User – A remote user or process sending information to the workstation via a network protocol. This role has the authority to:
   - Send information to the workstation

Application Note: Network users may have authority for other functions on the workstation. However, for the purposes of a conforming TOE, the only relevant authority is what is stated.

3 TOE SECURITY ENVIRONMENT

This section discusses the characteristics of environments and threat levels appropriate for basic robustness TOEs, and it describes the specific security aspects of the environment in which the anti-virus application is intended to be used and the manner in which it is expected to be employed. This information is provided to help organizations using this PP insure that the functional requirements specified by this PP are appropriate for their intended application of a compliant TOE.

This section includes the following:

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

3.1 CHARACTERIZING BASIC ROBUSTNESS

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

3.1.1 TOE Environment Defining Factors

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.

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 "FOUO", while high-value data may be those classified Top Secret. In a commercial enterprise, low-value data might be the internal organizational structure as captured in the corporate on-line phone book, while high-value data might be corporate research results for the next generation product. Note that when considering the value of the data one must also consider the value of data or resources that are accessible through exploitation of the TOE. For example, a firewall may have “low value” data itself, but it might protect an enclave with high value data. If the firewall was being depended upon to protect the high value data, then it must be treated as a high-value-data TOE.

Authorization of Entities

Authorization that entities (users, administrators, other IT systems) have with respect to the TOE (and thus the resources of that TOE, including the TOE itself) is an abstract concept reflecting a combination of the trustworthiness of an entity and the access and privileges granted to that entity with respect to the resources of the TOE. For instance, entities that have total authorization to all data on the TOE are at one end of this spectrum; these entities may have privileges that allow them to read, write, and modify anything on the TOE, including all TSF data. Entities at the other end of the spectrum are those that are authorized to few or no TOE resources. For example, in the case of a router, non-administrative entities may have their packets routed by the TOE, but that is the extent of their authorization to the TOE's resources. In the case of an 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.

3.1.2 Selection of Appropriate Robustness Levels

As defined above, robustness describes how well the TOE can protect itself and its resources. The more robust the TOE, the better it is able to protect itself. This section relates the defining factors of the IT environment, authorization, and value of resources to the selection of appropriate robustness levels.

When assessing any environment with regards to Information Assurance (IA), the critical point to consider is the likelihood of a compromise. This likelihood is somewhat dependent on the value of the TOE and resident data as well as logical connectivity and physical location. 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 environmental factors will result in environments in which the likelihood of an attempted compromise is similar. Consider the following two cases:

1. The first case is a TOE that processes low-value data. This TOE is connected to the Internet and is accessible by authorized entities. In this case, the least trusted entities are

unauthorized entities exposed to the TOE as a result of Internet connectivity. Since only low-value data is being processed, the likelihood that unauthorized entities would attempt to gain access to the system is low. In this instance, TOE compliance with a basic robustness PP is sufficient.

2. The second case is a TOE that processes high-value information. In this example, the TOE is a stand-alone system that is both logically isolated from any external connections and is physically protected. Additionally, every entity with physical and logical access to the TOE holds the highest authorizations thereby assuring that only highly trusted users are authorized to access the TOE. In this case, even though high value information is processed, it is unlikely that a compromise of the TOE and resident information will occur simply because of the physical and logical isolation and the trustworthiness of the entities. Once again, selection of a basic robustness TOE is appropriate.

The preceding examples demonstrated that it is possible for 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 Figure 3.1, the robustness of the TOEs required in each environment steadily increases as one goes from the upper left of the chart to the lower right; this corresponds to the need to counter increasingly likely attack attempts by the least trustworthy entities in the environment. Note that the shading of the chart is intended to reflects the notion that different environments engender similar levels of “likelihood of attempted compromise”, signified by a similar color. Further, the delineations between such environments are not stark, but rather are finely grained and gradual.

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 or particularly useful. Instead, in order to implement the robustness strategy where there are only three robustness levels (Basic, Medium, and High), the graph is divided into three sections, with each section corresponding to set of environments where the likelihood of attempted compromise is roughly similar. This is graphically depicted in Figure 3.2.

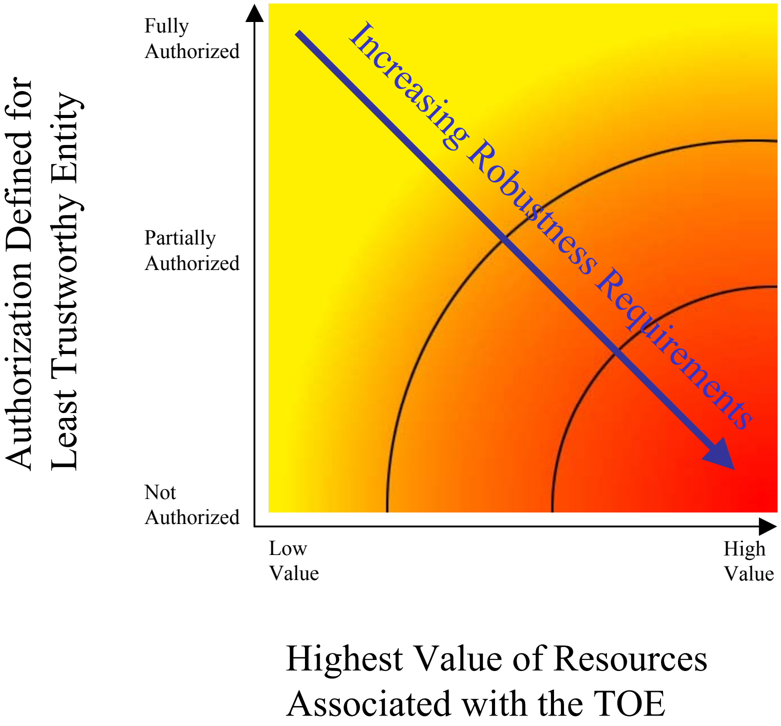
Authorization Defined for Least Trustworthy Entity
Fully Authorized
Partially Authorized
Not Authorized
Increasing Robustness Requirements
Low Value
High Value
Highest Value of Resources Associated with the TOE

Figure 3.1 – Robustness Requirements

In Figure 3.2 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.

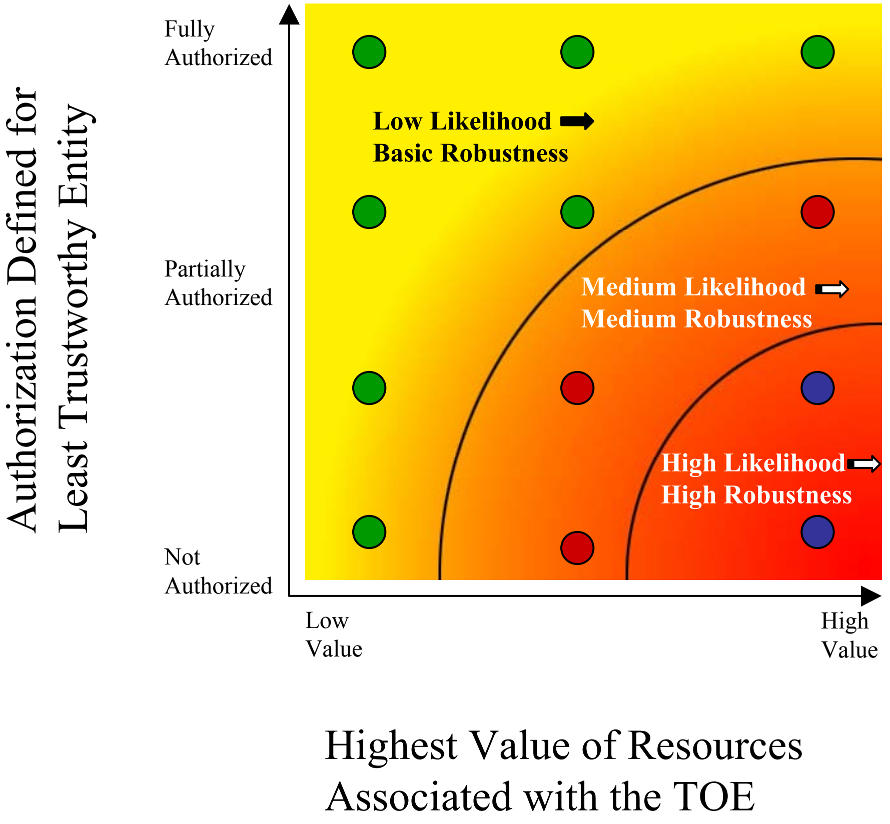

Figure 3.2 – Robustness Levels

3.1.3 Basic Robustness

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

Threat agent motivation can be considered in a variety of ways. One possibility is that the value of the data process 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.

3.2 SECURE USAGE ASSUMPTIONS

Table 3.1 lists the Secure Usage Assumptions.

Table 3.1 – Secure Usage Assumptions

Assumption	Assumption Description
A.AUDIT_BACKUP	Administrators will back up the audit files and monitor disk usage to ensure audit information is not lost.
A.NO_EVIL	Administrators are non-hostile, appropriately trained, and follow all administrator guidance.
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.
A.SECURE_COMMS	It is assumed that the IT environment will provide a secure line of communications between distributed portions of the TOE and between the TOE and remote administrators.
A.SECURE_UPDATES	Administrators will implement secure mechanisms for receiving and validating updated signature files from the Anti-Virus vendors, and for distributing the updates to the central management systems.

3.3 THREATS TO SECURITY

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:

1. The motivation for the threat agent defines the upper bound with respect to the level of robustness required for the TOE.
2. A threat agent's expertise and/or resources that are "lower" than the threat agent's motivation (e.g., a threat agent with high motivation but little expertise and few resources) may lessen the robustness requirements for the TOE (see next point, however).
3. 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.

Table 3.2 lists the threats to security.

Table 3.2 – Threats to Security

Threat	Description of Threat
T.ACCIDENTAL_ADMIN_ERROR	An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.

Threat	Description of Threat
T.AUDIT_ COMPROMISE	A user or process may gain unauthorized access to the audit trail and cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking a security relevant event.
T.MASQUERADE	A user or process may masquerade as another entity in order to gain unauthorized access to data or TOE resources.
T.POOR_DESIGN	Unintentional errors in requirements specification or design of the TOE may occur, leading to flaws that may be exploited by a casually mischievous user or program.
T.POOR_IMPLEMENTATION	Unintentional errors in implementation of the TOE design may occur, leading to flaws that may be exploited by a casually mischievous user or program.
T.POOR_TEST	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 memory used by the TOE to scan files or process administrator requests.
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).
T.UNATTENDED_SESSION	A user may gain unauthorized access to an unattended session.
T.UNIDENTIFIED_ACTIONS	Failure of the authorized administrator to identify and act upon unauthorized actions may occur.
T.VIRUS	A malicious agent may attempt to introduce a virus onto a workstation via network traffic or removable media to compromise data on that workstation, or use that workstation to attack additional systems.

3.4 ORGANIZATIONAL SECURITY POLICIES

Table 3.3 lists the organizational security policies.

Table 3.3 – Organizational Security Policies

Policy	Policy Description
P.ACCESS_BANNER	The system shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the system.
P.ACCOUNTABILITY	The authorized users of the TOE shall be held accountable for their actions within the TOE.
P.CRYPTOGRAPHY	Only NIST FIPS validated cryptography (methods and implementations) are acceptable for key management (i.e.; generation, access, distribution, destruction, handling, and storage of keys) and cryptographic services (i.e.; encryption, decryption, signature, hashing, key exchange, and random number generation services)
P.MANUAL_SCAN	The authorized users of the workstations shall initiate manual anti-virus scans of removable media (e.g., floppy disks, CDs) introduced into the workstation before accessing any data on that removable

	media.
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.

4 SECURITY OBJECTIVES

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

4.1 SECURITY OBJECTIVES FOR THE TOE

Table 4.1 contains the Security Objectives for the TOE.

Table 4.1 – Security Objectives for the TOE

Objective	Objective Description
O.ADMIN_GUIDANCE	The TOE will provide administrators with the necessary information for secure management.
O.ADMIN_ROLE	The TOE will provide an authorized administrator role to isolate administrative actions.
O.AUDIT_GENERATION	The TOE will provide the capability to detect and create records of security relevant events.
O.AUDIT_PROTECT	The TOE will provide the capability to protect audit information.
O.AUDIT_REVIEW	The TOE will provide the capability to selectively view audit information,
O.CONFIGURATION_IDENTIFICATION	The configuration of the TOE is fully identified in a manner that will allow implementation errors to be identified.
O.CORRECT_TSF_OPERATION	The TOE will provide the capability to test the TSF to ensure the correct operation of the TSF at a customer's site.
O.CRYPTOGRAPHY	The TOE shall use NIST FIPS 140-2 validated cryptographic services.
O.DOCUMENTED_DESIGN	The design of the TOE is adequately and accurately documented.
O.MANAGE	The TOE will provide all the functions and facilities necessary to support the authorized users in their management of the TOE.
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_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.VIRUS	The TOE will detect and take action against known viruses introduced to the workstation via network traffic or removable media.
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.

4.2 SECURITY OBJECTIVES FOR THE ENVIRONMENT

Table 4.2 contains security objectives for the environment.

Table 4.2 – Security Objectives for the IT Environment

Objective	Objective Description
OE.AUDIT_BACKUP	Audit log files are backed up and can be restored, and audit log files will not run out of disk space.
OE.AUDIT_STORAGE	The IT environment will provide a means for secure storage of the TOE audit log files.
OE.DISPLAY_BANNER	The IT environment will display an advisory warning regarding use of the system.
OE.DOMAIN_SEPARATION	The IT environment will provide an isolated domain for the execution of the TOE.
OE.NO_BYPASS	The IT environment shall ensure the TOE security mechanisms cannot be bypassed in order to gain access to the TOE resources.
OE.NO_EVIL	Sites using the TOE shall ensure that authorized administrators are non-hostile, appropriately trained and follow all administrator guidance.
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.
OE.RESIDUAL_INFORMATION	The IT environment will ensure that any information contained in a protected resource within the TOE Scope of Control is not released when the resource is reallocated.
OE.SECURE_COMMS	The IT environment will provide a secure line of communications between distributed portions of the TOE and between the TOE and remote administrators.
OE.SECURE_UPDATES	Enterprises using the TOE shall ensure that signature file updates are received from the vendor via secure mechanisms, the updates are validated before being used, and the updates are distributed to central management systems within the Enterprise via secure mechanisms.
OE.TIME_STAMPS	The IT environment will provide reliable time stamps.
OE.TOE_ACCESS	The IT Environment will provide mechanisms that control a user's logical access to the TOE.

5 IT SECURITY REQUIREMENTS

This section provides the TOE security functional and assurance requirements that must be satisfied by a Protection Profile-compliant TOE, and the IT environment security functional requirements on which the TOE relies. These requirements consist of functional components from Part 2 of the CC, assurance components from Part 3 of the CC, Common Criteria interpretations, NIAP interpretations, and explicit functional components derived from the CC components.

Formatting Conventions

The following formatting conventions apply to the TOE Security Functional Requirements and the Requirements for the IT Environment.

The CC allows several operations to be performed on functional requirements; *refinement, selection, assignment*, and *iteration* are defined in paragraph 2.1.4 of Part 2 of the CC. Each of these operations is used in this PP.

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

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

The **assignment** operation is used to assign a specific value to an unspecified parameter, such as the length of a password. An assignment is indicated by showing the value underlined, assignment_value.

Application notes provide additional information for the reader, but do not specify requirements. Application notes are denoted by *italicized text*.

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). (*) refers to all iterations of a component.

This PP contains several assignment and selection operations left to the ST writer to perform. The notation convention used for these is identical to that used in the Common Criteria.

5.1 TOE SECURITY FUNCTIONAL REQUIREMENTS

The functional security requirements for the TOE consist of the following components derived from Part 2 of the CC, CC interpretations, NIAP interpretations, and explicit components, summarized in Table 5.1 below.

Table 5.1 – TOE Security Functional Components

Component	Name
FAU_GEN.1-NIAP-0347	Audit Data Generation
FAU_GEN.2-NIAP-0410	User Identity Association
FAU_SAR.1	Audit Review
FAU_SAR.2	Restricted Audit Review
FAU_SAR.3	Selectable Audit Review
FAU_STG.1-NIAP-0429	Protected Audit Trail Storage
FAU_STG.NIAP-0414-NIAP-0429	Site-Configurable Prevention of Audit Loss
FAV_ACT_EXP.1	Anti-Virus Actions
FAV_ALR_EXP.1	Anti-Virus Alerts
FAV_SCN_EXP.1	Anti-Virus Scanning
FCS_COP1	Cryptographic Operation
FMT_MOF.1	Management of Security Functions Behaviour
FMT_MTD.1	Management of TSF Data
FMT_SMF.1	Specification of Management Functions
FMT_SMR.1	Security Roles
FPT_SEP_EXP.1	Partial TSF Domain Separation

5.1.1 Class FAU: Security audit

5.1.1.1 FAU_GEN.1-NIAP-0347 Audit Data Generation

FAU_GEN.1.1-NIAP-0347 The TSF shall be able to generate an audit record of the following auditable events:

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

b) All auditable events for the *minimum* level of audit; and

c) *The events identified in Table 5.2*.

FAU_GEN.1.2-NIAP-0347 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, *the additional information identified in Table 5.2*.

Table 5.2 – FAU_GEN.1 Events and Additional Information

SFR	Auditable Events	Additional Information
FAU_GEN.1-NIAP-0347	None	Not Applicable
FAU_GEN.2-NIAP-0410	None	Not Applicable
FAU_SAR.1	None	Not Applicable
FAU_SAR.2	None	Not Applicable

SFR	Auditable Events	Additional Information
FAU_SAR.3	None	Not Applicable
FAU_STG.1-NIAP-0429	None	Not Applicable
FAU_STG.NIAP-0414-NIAP-0429	Selection of an action	Action selected
FAV_ACT_EXP.1	Action taken in response to detection of a virus	Virus detected Action taken File or process identifier where the virus was detected
FAV_ALR_EXP.1	None	Not Applicable
FAV_SCN_EXP.1	None	Not Applicable
FCS_COP.1	None	Not Applicable
FMT_MOF.1	None	Not Applicable
FMT_MTD.1	None	Not Applicable
FMT_SMF.1	None	Not Applicable
FMT_SMR.1	None	Not Applicable
FPT_SEP_EXP.1	None	Not Applicable

5.1.1.2 FAU_GEN.2-NIAP-0410 User Identity Association

FAU_GEN.2.1-NIAP-0410 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 FAU_SAR.1 Audit Review

FAU_SAR.1.1(1) Refinement: The TSF shall provide the Central Administrator with the capability to read all audit information from the audit records **on the central management system**.

FAU_SAR.1.2(1) The TSF shall provide the audit records in a manner suitable for the user to interpret the information.

FAU_SAR.1.1(2) Refinement: The TSF shall provide the Central Administrator and Workstation Users with the capability to read all audit information from the audit records **on the workstation being used**.

FAU_SAR.1.2(2) The TSF shall provide the audit records in a manner suitable for the user to interpret the information.

Application Note: The Workstation User is permitted to review all audit records saved on the workstation being used by that user. The Central Administrator is permitted to review all logs on a specific workstation (which will only apply to that workstation) or on the central management system (which will apply to all workstations within that domain).

5.1.1.4 FAU_SAR.2 Restricted Audit Review

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

Application Note: This SFR applies to read access to the audit records through the TSFIs. The IT Environment (OS) is responsible for prohibiting read access to the audit file via OS interfaces.

5.1.1.5 FAU_SAR.3 Selectable Audit Review

Hierarchical to: No other components.

FAU_SAR.3.1 The TSF shall provide the ability to perform *searches and sorting* of audit data based on

a) Date and time of the event,

b) Type of event, and

c) Subject identity.

5.1.1.6 FAU_STG.1-NIAP-0429 Protected Audit Trail Storage

FAU_STG.1.1-NIAP-0429 Refinement: The TSF shall protect the stored audit records in the audit trail from unauthorised deletion **via the TSFI**.

FAU_STG.1.2-NIAP-0429 Refinement: The TSF shall be able to *prevent* unauthorised modifications to the audit records in the audit trail **via the TSFI**.

Application Note: FAU_STG.1-NIAP-0429 applies to both the central management system and the individual workstations.

Application Note: This instance of FAU_STG.1-NIAP-0429 applies to protection of the audit records via the TSFI. The IT Environment (OS) is responsible for preventing deletion of the audit file via OS interfaces.

5.1.1.7 FAU_STG.NIAP-0414-NIAP-0429 Site-Configurable Prevention of Audit Loss

FAU_STG.NIAP-0414-1-NIAP-0429 The TSF shall provide the administrator the capability to select one or more of the following actions [selection: *'ignore auditable events', 'prevent auditable events, except those taken by the authorised user with special rights', 'overwrite the oldest stored audit records'*] and [assignment: *other actions to be taken in case of audit storage failure]* to be taken if the audit trail is full.

FAU_STG.NIAP-0414-2-NIAP-0429 The TSF shall [selection: *'ignore auditable events', 'prevent auditable events, except those taken by the authorised user with special rights', 'overwrite the oldest stored audit records',* [assignment: *other actions to be taken in case of audit storage failure*]] if the audit trail is full and no other action has been selected.

5.1.2 Class FAV: Anti-Virus (Explicitly Stated)

5.1.2.1 FAV_ACT_EXP.1 Anti-Virus Actions

FAV_ACT_EXP.1.1 Upon detection of a memory-based virus, the TSF shall prevent the virus from further execution.

FAV_ACT_EXP.1.2 Upon detection of a file-based virus, the TSF shall perform the action(s) specified by the Central Administrator. Actions are administratively configurable on a per-workstation basis and consist of:

a) Clean the virus from the file,

b) Quarantine the file,

c) Delete the file,

d) [selection: [assignment: *list of other actions*], *no other actions*].

FAV_ACT_EXP.1.3 The TSF shall actively monitor processes attempting to access a remote system using TCP or UDP remote port 25 (SMTP) and block traffic from unauthorized processes defined by [assignment: *ST author to complete*] and simultaneously permit traffic from authorized processes defined by [assignment: *ST author to complete*].

5.1.2.2 FAV_ALR_EXP.1 Anti-Virus Alerts

FAV_ALR_EXP.1.1 Upon detection of a virus, the TSF shall display an alert on the screen of the workstation on which the virus is detected. The alert shall identify the virus that was detected and the action taken by the TOE.

FAV_ALR_EXP.1.2 The TSF shall continue to display the alerts on the screen of the workstation until they are acknowledged by the user of the workstation, or the user session ends.

FAV_ALR_EXP.1.3 Upon receipt of an audit event from a workstation indicating detection of a virus, the TSF shall display an alert on the screen of the Central Administrator if a session is active. The alert shall identify the workstation originating the audit event, the virus that was detected and the action taken by the TOE.

FAV_ALR_EXP.1.4 The TSF shall continue to display the alerts on the screen of the Central Administrator until they are acknowledged by the Central Administrator, or the Central Administrator session ends.

5.1.2.3 FAV_SCN_EXP.1 Anti-Virus Scanning

FAV_SCN_EXP.1.1 The TSF shall perform real-time scans for memory-based viruses based upon known signatures.

FAV_SCN_EXP.1.2 The TSF shall perform real-time, scheduled, and on-demand scans for file-based viruses based upon known signatures.

FAV_SCN_EXP.1.3 The TSF shall perform scheduled scans at the time and frequency configured by the Central Administrator.

FAV_SCN_EXP.1.4 The TSF shall perform manually invoked scans when directed by the Workstation User.

5.1.3 Class FCS: Cryptographic Support

5.1.3.1 FCS_COP.1 Cryptographic Operation

FCS_COP.1.1 Refinement: The TSF shall perform calculate a message digest to verify the integrity of the signature files in accordance with a specified cryptographic algorithm [assignment: ***NIST FIPS 140-2 Approved*** *cryptographic algorithm*] and cryptographic key sizes (not applicable) that meet the following: [assignment: *list of standards*].

Application Note: Conforming STs should specify the Cryptographic Module Validation Program (CMVP) validated algorithm certificate number.

Application Note: Message digests use hash functions, which do not have keys. Therefore, the assignment related to the cryptographic key size has been set to "not applicable".

5.1.4 Class FMT: Security management

5.1.4.1 FMT_MOF.1 Management of Security Functions Behaviour

FMT_MOF.1.1(1) The TSF shall restrict the ability to *determine the behaviour of, disable, enable* the functions

a) Auditing,

b) Real-time virus scanning, and

c) Scheduled virus scanning

to the Central Administrator.

FMT_MOF.1.1(2) The TSF shall restrict the ability to *modify the behaviour of* the functions manually invoked virus scanning to Workstation Users.

5.1.4.2 FMT_MTD.1 Management of TSF Data

FMT_MTD.1.1(1) The TSF shall restrict the ability to *query, modify, delete* the

a) Actions to be taken on workstations when a virus is detected,

b) Files to be scanned automatically on workstations,

c) Minimum depth of file scans on workstations,

d) Scheduled scan frequency on workstations,

e) Processes authorized to transmit data to a remote system using TCP or UDP remote port 25 (SMTP).

f) Virus scan signatures, and

g) Audit logs on the central management system

to the Central Administrator.

FMT_MTD.1.1(2) The TSF shall restrict the ability to *modify* the

a) Depth of file scans on manually invoked scans on workstations, and

b) Files to be scanned manually on workstations

to the Central Administrator and Workstation Users.

FMT_MTD.1.1(3) The TSF shall restrict the ability to *query, delete* the audit logs on the workstation being used to the Central Administrator and Workstation Users.

5.1.4.3 FMT_SMF.1 Specification of Management Functions

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

a) Enable and disable operation of the TOE on workstations,

b) Configure operation of the TOE on workstations,

c) Update virus scan signatures,

d) Acknowledge alert notifications from the central management system,

e) Review audit logs on the central management system,

f) Increase the depth of file scans on manually invoked scans,

g) Acknowledge alert notifications on the workstation being used, and

h) Review audit logs on the workstation being used.

5.1.4.4 FMT_SMR.1 Security Roles

FMT_SMR.1.1 The TSF shall maintain the roles Central Administrator, Workstation User, Network User.

5.1.5 Class FPT: Protection of the TOE Security Functions

5.1.5.1 FPT_SEP_EXP.1 Partial TSF Domain Separation

FPT_SEP_EXP.1.1 The TSF shall maintain a security domain that protects it from interference and tampering by untrusted subjects initiating actions through its own TSFI.

FPT_SEP_EXP.1.2 The TSF shall enforce separation between the security domains of subjects in the TOE Scope of Control.

5.2 SECURITY REQUIREMENTS FOR THE IT ENVIRONMENT

This Protection Profile provides functional requirements for the IT Environment. These requirements consist of functional components derived from Part 2 of the CC, CC interpretations, and NIAP interpretations, summarized in the following table.

Table 5.3 – IT Environment Security Functional Components

Component	Name
FAU_STG.1-NIAP-0429	Protected Audit Trail Storage
FDP_RIP.1	Subset Residual Information Protection
FIA_AFL.1	Authentication Failure Handling

Component	Name
FIA_SOS.1	Verification of Secrets
FIA_UAU.2	User Authentication Before any Action
FIA_UAU.6	Re-Authenticating
FIA_UID.2	User Identification Before any Action
FPT_ITT.1	Basic Internal TSF Data Transfer Protection
FPT_RVM.1	Non-Bypassability of the TSP
FPT_SEP.1	TSF Domain Separation
FPT_STM.1	Reliable Time Stamps
FTA_SSL.1	TSF-Initiated Session Locking
FTA_TAB.1	Default TOE Access Banners

5.2.1 Class FAU: Security audit

5.2.1.1 FAU_STG.1-NIAP-0429 Protected Audit Trail Storage

FAU_STG.1.1-NIAP-0429 Refinement: The **IT Environment** shall protect the stored audit records in the audit trail **file(s)** from unauthorised deletion.

FAU_STG.1.2-NIAP-0429 Refinement: The **IT Environment** shall be able to *prevent* unauthorised modifications to the audit records in the audit trail **file(s)**.

Application Note: This instance of FAU_STG.1-NIAP-0429 applies to the audit trail file(s) as a whole, while the instance levied against the TOE applies to individual records within the files.

5.2.2 Class FDP: User Data Protection

5.2.2.1 FDP_RIP.1 Subset Residual Information Protection

FDP_RIP.1.1 Refinement: The **IT Environment** shall ensure that any previous information content of a resource is made unavailable upon the [selection: *allocation of the resource to, deallocation of the resource from*] the following objects: all objects used by the TOE.

5.2.3 Class FIA: Identification and Authentication

5.2.3.1 FIA_AFL.1 Authentication Failure Handling

FIA_AFL.1.1 Refinement: The **IT Environment** shall detect when [selection: [assignment: *positive integer number*], *"an administrator configurable positive integer within [assignment: range of acceptable values]"*] unsuccessful authentication attempts occur related to the unsuccessful authentication attempts since the last successful authentication for the Central Administrator or Workstation User.

FIA_AFL.1.2 Refinement: When the defined number of unsuccessful authentication attempts has been met or surpassed, the **IT Environment** shall [assignment: *list of actions*].

5.2.3.2 FIA_SOS.1 Verification of Secrets

FIA_SOS.1.1 Refinement: The **IT Environment** shall provide a mechanism to verify that secrets meet [assignment: *a defined quality metric* ***that is sufficient to satisfy SOF-basic requirements***].

5.2.3.3 FIA_UAU.2 User Authentication Before any Action

FIA_UAU.2.1 Refinement: The **IT Environment** shall require each **Central Administrator or Workstation User** to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

Application Note: Network Users are not subject to the I&A requirements.

5.2.3.4 FIA_UAU.6 Re-Authenticating

FIA_UAU.6.1 Refinement: The **IT Environment** shall re-authenticate the **Central Administrator or Workstation User** under the conditions <u>the session is locked due to inactivity</u>.

5.2.3.5 FIA_UID.2 User Identification Before any Action

FIA_UID.2.1 Refinement: The **IT Environment** shall require each **Central Administrator or Workstation User** to identify itself before allowing any other TSF-mediated actions on behalf of that user.

Application Note: Network Users are not subject to the I&A requirements.

5.2.4 Class FPT: Protection of the TSF

5.2.4.1 FPT_ITT.1 Basic Internal TSF Data Transfer Protection

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

5.2.4.2 FPT_RVM.1 Non-Bypassability of the TSP

FPT_RVM.1.1 Refinement: The **IT Environment** shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

5.2.4.3 FPT_SEP.1 TSF Domain Separation

FPT_SEP.1.1 Refinement: The **IT Environment** shall maintain a security domain for **the TOE** own execution that protects **the TOE** from interference and tampering by untrusted subjects.

FPT_SEP.1.2 Refinement: The **IT Environment** shall enforce separation between the security domains of subjects in the TSC.

5.2.4.4 FPT_STM.1 Reliable Time Stamps

FPT_STM.1.1 Refinement: The **IT Environment** shall be able to provide reliable time-stamps for **the TOE's** use.

5.2.4.5 FTA_SSL.1 TSF-Initiated Session Locking

FTA_SSL.1.1 Refinement: The **IT Environment** shall lock an interactive session **of the Central Administrator or Workstation User** after [assignment: *time interval of user inactivity*] by:

a) Clearing or overwriting display devices, making the current contents unreadable;

b) Disabling any activity of the user's data access/display devices other than unlocking the session.

FTA_SSL.1.2 Refinement: The **IT Environment** shall require the following events to occur prior to unlocking the **Central Administrator or Workstation User** session: re-authentication.

5.2.4.6 FTA_TAB.1 Default TOE Access Banners

FTA_TAB.1.1 Refinement: Before establishing a user session, the **IT Environment** shall display an advisory warning message regarding unauthorised use of the **system**.

5.3 TOE SECURITY ASSURANCE REQUIREMENTS

The TOE assurance requirements for this PP are the Basic Robustness Assurance Package and are equivalent to EAL2 augmented by ALC_FLR.2 and AVA_MSU.1. The assurance requirements are summarized in Table 5.4 below. Please see Section 6.7, 'Rationale for Assurance Requirements' for more information on the Basic Robustness Assurance Package.

Table 5.4 – Assurance Requirements

Assurance Class	Assurance Components	
Configuration management	ACM_CAP.2	Configurations items
Delivery and operation	ADO_DEL.1	Delivery procedures
	ADO_IGS.1	Installation, generation, and start-up procedures
Development	ADV_FSP.1	Informal Functional specification
	ADV_HLD.1	Descriptive high-level design
	ADV_RCR.1	Informal correspondence demonstration
Guidance documents	AGD_ADM.1	Administrator guidance
	AGD_USR.1	User guidance
Life cycle support	ALC_FLR.2	Flaw Reporting Procedures
Tests	ATE_COV.1	Evidence of coverage
	ATE_FUN.1	Functional testing
	ATE_IND.2	Independent testing - sample
Vulnerability assessment	AVA_MSU.1	Examination of guidance

Assurance Class	Assurance Components	
	AVA_SOF.1	Strength of TOE security function evaluation
	AVA_VLA.1	Developer Vulnerability analysis

ACM_CAP.2 Configuration Items

Application Note: This requirement applies CCIMB 003.

Developer action elements:
ACM_CAP.2.1D - The developer shall provide a reference for the TOE.
ACM_CAP.2.2D - The developer shall use a CM system.
ACM_CAP.2.3D - The developer shall provide CM documentation.
Content and presentation of evidence elements:
ACM_CAP.2.1C - The reference for the TOE shall be unique to each version of the TOE.
ACM_CAP.2.2C - The TOE shall be labeled with its reference.
ACM_CAP.2.3C - The CM documentation shall include a configuration list.
ACM_CAP.2.4C – The configuration list shall uniquely identify all configuration items that comprise the TOE.
ACM_CAP.2.5C - The configuration list shall describe the configuration items that comprise the TOE.
ACM_CAP.2.6C - The CM documentation shall describe the method used to uniquely identify the configuration items.
ACM_CAP.2.7C - The CM system shall uniquely identify all configuration items.

ADO_DEL.1 Delivery Procedures

Developer action elements:
ADO_DEL.1.1D - The developer shall document procedures for delivery of the TOE or parts of it to the user.
ADO_DEL.1.2D - The developer shall use the delivery procedures.
Content and presentation of evidence elements:
ADO_DEL.1.1C - The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.
Evaluator action elements:
ADO_DEL.1.1E - The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADO_IGS.1 Installation, generation, and start-up procedures

Application Note: This requirement applies CCIMB 051.

Developer action elements:
ADO_IGS.1.1D - The developer shall document procedures necessary for the secure installation, generation, and start-up of the TOE.
Content and presentation of evidence elements:
ADO_IGS.1.1C - The installation, generation and start-up documentation shall describe all the steps necessary for secure installation, generation, and start-up of the TOE.
Evaluator action elements:

ADO_IGS.1.1E - The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADO_IGS.1.2E - The evaluator shall determine that the installation, generation, and start-up procedures result in a secure configuration.

ADV_FSP.1 Informal Functional Specification

Developer action elements:
ADV_FSP.1.1D - The developer shall provide a functional specification.
Content and presentation of evidence elements:
ADV_FSP.1.1C - The functional specification shall describe the TSF and its external interfaces using an informal style.
ADV_FSP.1.2C - The functional specification shall be internally consistent.
ADV_FSP.1.3C - The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing details of effects, exceptions and error messages, as appropriate.
ADV_FSP.1.4C - The functional specification shall completely represent the TSF.
Evaluator action elements:
ADV_FSP.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_FSP.1.2E The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

ADV_HLD.1 Descriptive High-Level Design

Developer action elements:
ADV_HLD.1.1D - The developer shall provide the high-level design of the TOE.
Content and presentation of evidence elements:
ADV_HLD.1.1C - The presentation of the high-level design shall be informal.
ADV_HLD.1.2C - The high-level design shall be internally consistent.
ADV_HLD.1.3C - The high-level design shall describe the structure of the TSF in terms of subsystems.
ADV_HLD.1.4C - The high-level design shall describe the security functionality provided by each subsystem of the TSF.
ADV_HLD.1.5C - The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.
ADV_HLD.1.6C - The high-level design shall identify all interfaces to the subsystems of the TSF.
ADV_HLD.1.7C - The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.
Evaluator action elements:
ADV_HLD.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_HLD.1.2E The evaluator shall determine that the high-level design is an accurate and complete instantiation of the all TOE security functional requirements.

ADV_RCR.1 Informal correspondence demonstration

Application Note: The intent of this requirement is for the vendor to provide, and the evaluator to confirm, that there exists accurate, consistent, and clear mappings between each level of design decomposition. Thus there can be no TOE security functions defined at a lower layer of abstraction absent from a higher level of abstraction and vice versa.

Developer action elements:
ADV_RCR.1.1D - The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided.
Content and presentation of evidence elements:
ADV_RCR.1.1C - For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.
Evaluator action elements:
ADV_RCR.1.1E - The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AGD_ADM.1 Administrator guidance

Developer action elements:
AGD_ADM.1.1D - The developer shall provide administrator guidance addressed to system administrative personnel.
Content and presentation of evidence elements:
AGD_ADM.1.1C - The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.
AGD_ADM.1.2C - The administrator guidance shall describe how to administer the TOE in a secure manner.
AGD_ADM.1.3C - The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.
AGD_ADM.1.4C - The administrator guidance shall describe all assumptions regarding user behavior that are relevant to secure operation of the TOE.
AGD_ADM.1.5C - The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.
AGD_ADM.1.6C - The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.
AGD_ADM.1.7C - The administrator guidance shall be consistent with all other documentation supplied for evaluation.
AGD_ADM.1.8C - The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.
Evaluator action elements:
AGD_ADM.1.1E - The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AGD_USR.1 User guidance

Developer action elements:
AGD_USR.1.1D - The developer shall provide user guidance.

Content and presentation of evidence elements:
AGD_USR.1.1C - The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.
AGD_USR.1.2C - The user guidance shall describe the use of user-accessible security functions provided by the TOE.
AGD_USR.1.3C - The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.
AGD_USR.1.4C - The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behavior found in the statement of TOE security environment.
AGD_USR.1.5C - The user guidance shall be consistent with all other documentation supplied for evaluation.
AGD_USR.1.6C - The user guidance shall describe all security requirements for the IT environment that are relevant to the user.
Evaluator action elements:
AGD_USR.1.1E - The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_DVS.1 Identification of security measures

Developer action elements:
ALC_DVS.1.1D - The developer shall produce development security documentation.
Content and presentation of evidence elements:
ALC_DVS.1.1C - The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.
ALC_DVS.1.2C - The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.
Evaluator action elements:
ALC_DVS.1.1E - The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ALC_DVS.1.2E - The evaluator shall confirm that the security measures are being applied.

ALC_FLR.2 Flaw Reporting Procedures

Application Note: This requirement applies CCIMB 062.

Developer action elements:
ALC_FLR.2.1D - The developer shall document the flaw remediation procedures.
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.
Content and presentation of evidence 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 procedures for processing reported security flaws shall ensure that any reported flaws are corrected and the correction issued to TOE users.
ALC_FLR.2.6C - The procedures for processing reported security flaws shall provide safeguards that any corrections to these security flaws do not introduce any new flaws.
Evaluator action elements:
ALC_FLR.2.1E - The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_COV.1 Evidence of coverage

Developer action elements:
ATE_COV.1.1D - The developer shall provide evidence of the test coverage.
Content and presentation of evidence elements:
ATE_COV.1.1C - The evidence of the test coverage shall show the correspondence between the tests identified in the test documentation and the TSF as described 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.

ATE_FUN.1 Functional testing

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 of evidence elements:
ATE_FUN.1.1C - The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.
ATE_FUN.1.2C - The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.
ATE_FUN.1.3C - The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.
ATE_FUN.1.4C - The expected test results shall show the anticipated outputs from a successful execution of the tests.
ATE_FUN.1.5C - The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.
Evaluator action elements:
ATE_FUN.1.1E - The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2 Independent testing - sample

Developer action elements:
ATE_IND.2.1D - The developer shall provide the TOE for testing.

Content and presentation of evidence 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 test a subset of the TSF as appropriate to confirm that the TOE operates as specified.
ATE_IND.2.3E - The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

AVA_MSU.1 Examination of Guidance

Developer action elements:
AVA_MSU.1.1D - The developer shall provide guidance documentation.
Content and presentation of evidence elements:
AVA_MSU.1.1C - The guidance documentation 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.
AVA_MSU.1.2C - The guidance documentation shall be complete, clear, consistent and reasonable.
AVA_MSU.1.3C - The guidance documentation shall list all assumptions about the intended environment.
AVA_MSU.1.4C - The guidance documentation shall list all requirements for external security measures (including external procedural, physical and personnel controls).
Evaluator action elements:
AVA_MSU.1.1E - The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AVA_MSU.1.2E - The evaluator shall repeat all configuration and installation procedures to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.
AVA_MSU.1.3E - The evaluator shall determine that the use of the guidance documentation allows all insecure states to be detected.

AVA_SOF.1 Strength of TOE security function evaluation

Developer action elements:
AVA_SOF.1.1D - The developer shall perform a strength of TOE security function analysis for each mechanism identified in the Security Target as having a strength of TOE security function claim.
Content and presentation of evidence elements:
AVA_SOF.1.1C - For each mechanism with a strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the minimum strength level defined in the PP/ST.
AVA_SOF.1.2C - For each mechanism with a specific strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the specific strength of function metric defined in the PP/ST.

Evaluator action elements:
AVA_SOF.1.1E - The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AVA_SOF.1.2E - The evaluator shall confirm that the strength claims are correct.

AVA_VLA.1 Developer Vulnerability Analysis

Application Note: This requirement applies CCIMB 051.

Developer action elements:
AVA_VLA.1.1D - The developer shall perform a vulnerability analysis.
AVA_VLA.1.2D - The developer shall provide vulnerability analysis documentation.
Content and presentation of evidence elements:
AVA_VLA.1.1C - The vulnerability analysis documentation shall describe the analysis of the TOE deliverables performed to search for obvious ways in which a user can violate the TSP.
AVA_VLA.1.2C - The vulnerability analysis documentation shall describe the disposition of obvious vulnerabilities.
AVA_VLA.1.3C – The vulnerability analysis documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.
Evaluator action elements:
AVA_VLA.1.1E - The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AVA_VLA.1.2E - The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure obvious vulnerabilities have been addressed.

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.

{This page intentionally left blank}

6 RATIONALE

This section describes the rationale for the Security Objectives and Security Functional Requirements as defined in Section 4 and Section 5, respectively. Additionally, this section describes the rationale for not satisfying all of the dependencies and the rationale for the strength of function (SOF) claim.

6.1 MAPPING OF THREATS, POLICIES, AND ASSUMPTIONS TO OBJECTIVES

The following table presents a mapping of the threats, assumptions, and policies to the objectives defined in this PP.

Table 6.1 – Mapping of Threats, Policies, and Assumptions to Objectives

	A.AUDIT_BACKUP	A.NO_EVIL	A.PHYSICAL	A.SECURE_COMMS	A.SECURE_UPDATES	T.ACCIDENTAL_ADMIN_ERROR	T._AUDIT_ COMPROMISE	T.MASQUERADE	T.POOR_DESIGN	T.POOR_IMPLEMENTATION	T.POOR_TEST	T.RESIDUAL_DATA	T.TSF_COMPROMISE	T.UNATTENDED_SESSION	T.UNIDENTIFIED_ACTIONS	T.VIRUS	P.ACCESS_BANNER	P.ACCOUNTABILITY	P.CRYPTOGRAPHY	P.MANUAL_SCAN	P.ROLES
O.ADMIN_GUIDANCE						X															
O.ADMIN_ROLE																					X
O.AUDIT_GENERATION															X			X			
O.AUDIT_PROTECT							X														
O.AUDIT_REVIEW															X						
O.CONFIGURATION_IDENTIFICATION									X	X											
O.CORRECT_TSF_OPERATION											X		X								
O.CRYPTOGRAPHY																			X		
O.DOCUMENTED_DESIGN									X		X										
O.MANAGE													X							X	
O.PARTIAL_FUNCTIONAL_TEST										X	X										
O.PARTIAL_SELF_PROTECTION							X						X								
O.VIRUS																X				X	
O.VULNERABILITY_ANALYSIS									X	X	X										
OE.AUDIT_BACKUP	X																				
OE.AUDIT_STORAGE							X														
OE.DISPLAY_BANNER																	X				

	A.AUDIT_BACKUP	A.NO_EVIL	A.PHYSICAL	A.SECURE_COMMS	A.SECURE_UPDATES	T.ACCIDENTAL_ADMIN_ERROR	T._AUDIT_ COMPROMISE	T.MASQUERADE	T.POOR_DESIGN	T.POOR_IMPLEMENTATION	T.POOR_TEST	T.RESIDUAL_DATA	T.TSF_COMPROMISE	T.UNATTENDED_SESSION	T.UNIDENTIFIED_ACTIONS	T.VIRUS	P.ACCESS_BANNER	P.ACCOUNTABILITY	P.CRYPTOGRAPHY	P.MANUAL_SCAN	P.ROLES
OE.DOMAIN_SEPARATION							X						X								
OE.NO_BYPASS							X						X								
OE.NO_EVIL		X																			
OE.PHYSICAL			X																		
OE.RESIDUAL_INFORMATION							X					X	X								
OE.SECURE_COMMS				X																	
OE.SECURE_UPDATES					X																
OE.TIME_STAMPS															X			X			
OE.TOE_ACCESS								X						X				X			

6.2 RATIONALE FOR TOE SECURITY OBJECTIVES

Table 6.2 – Security Objectives to Threats and Policies Mappings

Threat/Policy/Assumption	Addressed By	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 a way that is insecure.
T.AUDIT_ COMPROMISE: A user or process may cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking a user's action.	**O.AUDIT_PROTECT:** The TOE will provide the capability to protect audit information. **OE.AUDIT_STORAGE:** The IT environment will contain mechanisms to provide secure storage and management of the audit log. **OE.RESIDUAL_ INFORMATION:** The TOE will ensure that any information contained in a protected resource	**O.AUDIT_PROTECT** contributes to mitigating this threat by controlling access to the individual audit log records. No one is allowed to modify audit records, the System Administrator is the only one allowed to delete audit records, and the TOE has the capability to prevent auditable actions from occurring if the audit trail is full. **OE.AUDIT_STORAGE** contributes to mitigating this threat by restricting the ability of users in the IT Environment to access the audit log file. **OE.RESIDUAL_INFORMATION** prevents a user not authorized to read the audit trail from access to audit information that might otherwise

Threat/Policy/Assumption	Addressed By	Rationale
	within its Scope of Control is not released when the resource is reallocated. **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. **OE.DOMAIN_SEPARATION:** The IT environment will provide an isolated domain for the execution of the TOE. **OE.NO_BYPASS:** The IT environment shall ensure the TOE security mechanisms cannot be bypassed in order to gain access to the TOE resources.	be persistent in a resource used by the TOE (e.g., memory). By preventing residual information in a resource, audit information will not become available to any user or process except those explicitly authorized for that data. **O.PARTIAL_SELF_PROTECTION** contributes to countering this threat by ensuring that the TSF can protect itself from users via its own interfaces. This limits access to the audit information to the functions defined for the specified roles. **OE.DOMAIN_SEPARATION** contributes to countering this threat by ensuring that the TSF is protected from users through mechanisms other than its own interfaces. If the OS could not maintain and control a domain of execution for the TSF separate from other processes, the TSF could not be trusted to control access to the resources under its control, which includes the audit trail which are always invoked is also critical to the migration of this threat. **OE.NO_BYPASS** ensures audit compromise can not occur simply by bypassing the TSF.
T.MASQUERADE: A user or process may masquerade as another entity in order to gain unauthorized access to data or TOE resources.	**OE.TOE_ACCESS:** The IT Environment will provide mechanisms that control a user's logical access to the TOE.	**OE.TOE_ACCESS** mitigates this threat by requiring authorized administrators and workstation users to be identified and authenticated, a necessary step in controlling the logical access to the TOE and its resources by constraining how and when users can access the TOE. 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. **O.DOCUMENTED_DESIGN:** The design of the TOE is adequately and accurately documented. **O.VULNERABILITY_ANALYSIS:** The TOE will undergo some	**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** ensures that the design of the TOE is documented, permitting detailed review by evaluators. **O.VULNERABILITY_ANALYSIS_TEST** ensures that the design of the TOE is analyzed for design flaws.

Threat/Policy/Assumption	Addressed By	Rationale
	vulnerability analysis to demonstrate the design and implementation of the TOE does not contain any obvious flaws.	
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. **O.PARTIAL_FUNCTIONAL_TESTING:** The TOE will undergo some security functional testing that demonstrates the TSF satisfies some of its security functional requirements. **O.VULNERABILITY_ANALYSIS:** The TOE will undergo some vulnerability analysis demonstrate the design and implementation of the TOE does not contain any obvious flaws.	**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 implementation. **O.PARTIAL_FUNCTIONAL_TESTING** increases the likelihood that any errors that do exist in the implementation will be discovered through testing. **O.VULNERABILITY_ANALYSIS_TEST** 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 may result in incorrect TOE behavior being undiscovered thereby causing potential security vulnerabilities.	**O.DOCUMENTED_DESIGN** The design of the TOE will be adequately and accurately documented. **O.PARTIAL_FUNCTIONAL_TESTING:** The TOE will undergo some security functional testing that demonstrates the TSF satisfies the security functional requirements. **O.CORRECT_TSF_OPERATION:** The TOE will provide the capability to test the TSF to ensure the correct operation of the TSF at a customer's site. **O.VULNERABILITY_ANALYSIS:** The TOE will undergo some vulnerability analysis demonstrate the design and implementation of the TOE does not contain any	**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. **O.PARTIAL_FUNCTIONAL_TESTING** increases the likelihood that any errors that do exist in the implementation will be discovered through testing. **O.CORRECT_TSF_OPERATION** provides assurance that the TSF continues to operate as expected in the field. **O.VULNERABILITY_ANALYSIS_TEST** 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.

Threat/Policy/Assumption	Addressed By	Rationale
	obvious flaws.	
T.RESIDUAL_DATA: A user or process may gain unauthorized access to data through reallocation of memory used by the TOE to scan files or process administrator requests.	**OE.RESIDUAL_INFORMATION:** The IT Environment will ensure that any information contained in a protected resource within the TOE Scope of Control is not released when the resource is reallocated.	**OE.RESIDUAL_INFORMATION** counters this threat by ensuring that memory contents are not persistent when resources are released by the TOE and allocated to another user/process.
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).	**OE.RESIDUAL_INFORMATION:** The IT Environment will ensure that any information contained in a protected resource within the TOE Scope of Control is not released when the resource is reallocated. **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. **OE.DOMAIN_SEPARATION:** The IT environment will provide an isolated domain for the execution of the TOE. **O.MANAGE:** The TOE will provide all the functions and facilities necessary to support the authorized users in their management of the TOE. **O.CORRECT_TSF_OPERATION:** The TOE will provide the capability to test the TSF to ensure the correct operation of the TSF at a customer's site. **OE.NO_BYPASS:** The IT environment shall ensure the TOE security mechanisms cannot be bypassed in order to gain	**OE.RESIDUAL_INFORMATION** is necessary to mitigate this threat, because even if the security mechanisms do not allow a user to explicitly view TSF data, if TSF data were to inappropriately reside in a resource that was made available to a user, that user would be able to inappropriately view the TSF data. **O.PARTIAL_SELF_PROTECTION** is necessary so that the TSF protects itself and its resources from inappropriate access through its own interfaces. **OE.DOMAIN_SEPARATION** is necessary so that the TSF is protected from other processes executing on the workstation. **O.MANAGE** is necessary because an access control policy is not 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. **O.CORRECT_TSF_OPERATION** provides assurance that the TSF continues to operate as expected in the field. **OE.NO_BYPASS** ensures TSF compromise can not occur simply by bypassing the TSF.

Threat/Policy/Assumption	Addressed By	Rationale
	access to the TOE resources.	
T.UNATTENDED_ SESSION: A user may gain unauthorized access to an unattended session.	**OE.TOE_ACCESS:** The IT environment will provide mechanisms that control a user's logical access to the TOE.	**OE.TOE_ACCESS** helps to mitigate this threat by including mechanisms that place controls on user's sessions. Locking a session reduces the opportunity of someone gaining unauthorized access to the session when the console is unattended.
T.UNIDENTIFIED_ACTIONS: The administrator may not have the ability to notice potential security violations, thus limiting the administrator's ability to identify and take action against a possible security breach.	**O.AUDIT_REVIEW:** The TOE will provide the capability to selectively view audit information, **O.AUDIT_GENERATION:** The TOE will provide the capability to detect and create records of security relevant events associated with users. **OE.TIME_STAMPS:** The IT environment shall provide reliable time stamps for accountability and protocol purposes.	**O.AUDIT_REVIEW** helps to mitigate this threat by providing the Security Administrator with a required minimum set of configurable audit events that could indicate a potential security violation. By configuring these auditable events, the TOE monitors the occurrences of these events (e.g. set number of authentication failures, set number of information policy flow failures, self-test failures, etc.). **O.AUDIT_GENERATION** helps to mitigate this threat by recording actions for later review. **OE.TIME_STAMPS** helps to mitigate this threat by ensuring that audit records have correct timestamps.
T.VIRUS: A malicious agent may attempt to introduce a virus onto a workstation via network traffic or removable media to compromise data on that workstation, or use that workstation to attack additional systems.	**O.VIRUS:** The TOE will detect and take action against known viruses introduced to the workstation via network traffic or removable media.	**O.VIRUS** mitigates this threat by providing mechanisms to prevent a virus from being introduced onto a workstation.
P.ACCESS_BANNER: The system shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the system.	**OE.DISPLAY_BANNER:** The IT Environment will display an advisory warning regarding use of the system.	**OE.DISPLAY_BANNER** satisfies this policy by ensuring that the system displays a banner that provides all authorized users with a warning about the unauthorized use of the system.
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. **OE.TIME_STAMPS:** The IT environment shall provide reliable time stamps and the capability for the administrator to set the time	**O.AUDIT_GENERATION** addresses this policy by recording security-relevant events. The administrator's ID is recorded when any security relevant change is made to the TOE. **OE.TIME_STAMPS** plays a role in supporting this policy by requiring the IT environment to provide a reliable time stamp. The audit mechanism is required to include the current date and time in each audit record. **OE. TOE_ACCESS** supports this policy by requiring the IT environment to identify and authenticate all authorized administrators and

Threat/Policy/Assumption	Addressed By	Rationale
	used for these time stamps. **OE.TOE_ACCESS:** The TOE will provide mechanisms that control a user's logical access to the TOE.	workstation users prior to allowing any TOE access. While the user ID of these users can be assured, since they are authenticated, this PP allows unauthenticated users to access the TOE and the identity is then a presumed network identifier (e.g., IP address).
P.CRYPTOGRAPHY: Only NIST FIPS validated cryptography (methods and implementations) are acceptable for key management (i.e.; generation, access, distribution, destruction, handling, and storage of keys) and cryptographic services (i.e.; encryption, decryption, signature, hashing, key exchange, and random number generation services).	**O.CRYPTOGRAPHY:** The TOE shall use NIST FIPS 140-2 validated cryptographic services.	**O.CRYPTOGRAPHY** requires that cryptographic services conform to the policy by mandating FIPS 140-2 validation.
P.MANUAL_SCAN: The authorized users of the workstations shall initiate manual anti-virus scans of removable media (e.g., floppy disks, CDs) introduced into the workstation before accessing any data on that removable media.	**O.VIRUS:** The TOE will detect and take action against known viruses introduced to the workstation via network traffic or removable media. **O.MANAGE:** The TOE will provide all the functions and facilities necessary to support the authorized users in their management of the TOE.	**O.VIRUS** requires the TOE to provide the capability to perform manual scans of removable media. **O.MANAGE** provides the workstation user with the ability to invoke the manual scan capability.
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 an authorized administrator role to isolate administrative actions.	**O.ADMIN_ROLE** addresses this policy by requiring the TOE to support an administrator role, and restrict specific actions to that role.
A.AUDIT_BACKUP: Administrators will back up the audit files and monitor disk usage to ensure audit information is not lost.	**OE.AUDIT_BACKUP:** Audit log files are backed up and can be restored, and audit log files will not run out of disk space.	**OE.AUDIT_BACKUP** addresses the assumption by requiring the audit log files to be backed up, and by requiring monitoring of disk space usage to ensure space is available.
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, appropriately trained and follow all administrator guidance.	**OE.NO_EVIL** restates the assumption.

Threat/Policy/Assumption	Addressed By	Rationale
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.	**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.	**OE.PHYSICAL** restates the assumption.
A.SECURE_COMMS: It is assumed that the IT environment will provide a secure line of communications between distributed portions of the TOE and between the TOE and remote administrators.	**OE.SECURE_COMMS:** The IT environment will provide a secure line of communications between distributed portions of the TOE and between the TOE and remote administrators.	**OE.SECURE_COMMS** restates the assumption. The workstation OS will provide a secure line of communication for the TOE.
A.SECURE_UPDATES: Administrators will implement secure mechanisms for receiving and validating updated signature files from the Anti-Virus vendors, and for distributing the updates to the central management systems.	**OE.SECURE_UPDATES:** Enterprises using the TOE shall ensure that signature file updates are received from the vendor via secure mechanisms, the updates are validated before being used, and the updates are distributed to central management systems within the Enterprise via secure mechanisms.	**OE.SECURE_UPDATES** restates the assumption. Administrators use secure mechanisms to receive and validate the updates from the vendor, then use secure mechanisms to distribute the updates to the central management systems.

6.3 MAPPING OF IT ENVIRONMENT OBJECTIVES TO SECURITY FUNCTIONAL REQUIREMENTS

The following table presents a mapping of the IT Environment Objectives to IT Environment Security Functional Requirements defined in this PP.

Table 6.3 – Mapping of IT Environment Objectives to IT Environment SFRs

	OE.AUDIT_STORAGE	OE.DISPLAY_BANNER	OE.DOMAIN_SEPARATION	OE.NO_BYPASS	OE.RESIDUAL_INFORMATION	OE.SECURE_COMMS	OE.TIME_STAMPS	OE.TOE_ACCESS
FAU_STG.1-NIAP-0429	X							
FDP_RIP.1					X			
FIA_AFL.1								X

	OE.AUDIT_STORAGE	**OE.DISPLAY_BANNER**	**OE.DOMAIN_SEPARATION**	**OE.NO_BYPASS**	**OE.RESIDUAL_INFORMATION**	**OE.SECURE_COMMS**	**OE.TIME_STAMPS**	**OE.TOE_ACCESS**
FIA_SOS.1								X
FIA_UAU.2								X
FIA_UAU.6								X
FIA_UID.2								X
FPT_ITT.1						X		
FPT_RVM.1				X				
FPT_SEP.1			X					
FPT_STM.1							X	
FTA_SSL.1								X
FTA_TAB.1		X						

6.4 RATIONALE FOR THE SECURITY OBJECTIVES AND SECURITY REQUIREMENTS FOR THE IT ENVIRONMENT

Table 6.4 – Rationale for IT Environment Objectives

Objective	Requirements Addressing the Objective	Rationale
OE.AUDIT_STORAGE: The IT environment will provide a means for secure storage of the TOE audit log files.	FAU_STG.1-NIAP-0429	**FAU_STG.1-NIAP-0429** requires the OS to protect the audit log file from unauthorized deletion.
OE.DISPLAY_BANNER: The system will display an advisory warning regarding use of the system.	FTA_TAB.1	**FTA_TAB.1** meets this objective by requiring the system to display a banner before a user can establish an authenticated session.
OE.DOMAIN_SEPARATION: The IT environment will provide an isolated domain for the execution of the TOE.	FPT_SEP.1	**FTP_SEP.1** requires the OS to provide an isolated domain for the TOE.
OE.NO_BYPASS: The IT environment shall ensure the TOE security mechanisms cannot be bypassed in order to gain access to the TOE resources.	FPT_RVM.1	**FPT_RVM.1** requires the OS to ensure that the TOE will not be bypassed.

Objective	Requirements Addressing the Objective	Rationale
OE.RESIDUAL_INFORMATION: The IT environment will ensure that any information contained in a protected resource within the TOE Scope of Control is not released when the resource is reallocated.	FPT_RIP.2	**FDP_RIP.2** is used to ensure the contents of resources are not available to subjects other than those explicitly granted access to the data.
OE.SECURE_COMMS: The IT environment will provide a secure line of communications between distributed portions of the TOE.	FPT_ITT.1	**FPT_ITT.1** ensures that secure communication between the central management system and the workstations will be available to the TOE.
OE.TIME_STAMPS: The IT environment will provide reliable time stamps.	FPT_STM.1	**FPT_STM.1** requires that the IT Environment provide time stamps for the TOE's use.
OE.TOE_ACCESS: The IT Environment will provide mechanisms that control a user's logical access to the TOE.	FIA_AFL.1 FIA_SOS.1 FIA_UID.2 FIA_UAU.2 FIA_UAU.6 FTA_SSL.1	**FIA_AFL.1** provides a detection mechanism for unsuccessful authentication attempts by remote administrators, authenticated proxy users and authorized IT entities. The requirement enables a Security Administrator settable threshold that prevents unauthorized users from gaining access to authorized user's account by guessing authentication data by locking the targeted account. Thus, limiting an unauthorized user's ability to gain unauthorized access to the TOE. **FIA_SOS.1** ensures that the strength of the I&A mechanism will be adequate. **FIA_UID.2** requires that a user be identified to the TOE in order to access to the TOE. **FIA_UAU.2** requires that a user be authenticated by the TOE before accessing the TOE. **FIA_UAU.6** requires that a user be re-authenticated after a session is locked. **FTA_SSL.1** requires that sessions be locked after a period of inactivity. The combination of these SFRs ensures that users will successfully complete an I&A process of sufficient strength before they can gain access to the TOE.

6.5 MAPPING OF TOE OBJECTIVES TO SECURITY REQUIREMENTS

The following table presents a mapping of the TOE Objectives to TOE Security Requirements defined in this PP.

Table 6.5 – Mapping of TOE Objectives to TOE SFRs and SARs

	O.ADMIN_GUIDANCE	O.ADMIN_ROLE	O.AUDIT_GENERATION	O.AUDIT_PROTECT	O.AUDIT_REVIEW	O.CONFIGURATION_IDENTIFICATION	O.CORRECT_TSF_OPERATION	O.CRYPTOGRAPHY	O.DOCUMENTED_DESIGN	O.MANAGE	O.PARTIAL_FUNCTIONAL_TEST	O.PARTIAL_SELF_PROTECTION	O.VIRUS	O.VULNERABILITY_ANALYSIS
ACM_CAP.2						X								
ADO_DEL.1	X													
ADO_IGS.1	X													
ADV_FSP.1									X					
ADV_HLD.1									X					
ADV_RCR.1									X					
AGD_ADM.1	X													
AGD_USR.1	X													
ALC_FLR.2						X								
ATE_COV.1											X			
ATE_FUN.1											X			
ATE_IND.2											X			
AVA_MSU.1	X													
AVA_VLA.1														X
FAU_GEN.1-NIAP-0347			X				X							
FAU_GEN.2-NIAP-0410			X				X							
FAU_SAR.1					X		X							
FAU_SAR.2				X										
FAU_SAR.3					X		X							
FAU_STG.1-NIAP-0429				X										
FAU_STG.NIAP-0414-NIAP-0429				X										
FAV_ACT_EXP.1							X						X	
FAV_ALR_EXP.1							X						X	
FAV_SCN_EXP.1							X						X	
FCS_COP.1								X						
FMT_MOF.1		X								X				
FMT_MTD.1		X								X				
FMT_SMF.1		X								X				
FMT_SMR.1		X								X				
FPT_SEP_EXP.1												X		

6.6 RATIONALE FOR THE SECURITY OBJECTIVES AND SECURITY REQUIREMENTS FOR THE TOE

Table 6.6 – Rationale for TOE Objectives

Objective	Requirements Addressing the Objective	Rationale
O.ADMIN_GUIDANCE: The TOE will provide administrators with the necessary information for secure management.	ADO_DEL.1 ADO_IGS.1 AGD_ADM.1 AGD_USR.1 AVA_MSU.1	**ADO_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. **ADO_IGS.1** ensures the administrator has the information necessary to install the TOE in the evaluated configuration. Often times a vendor's product contains software that is not part of the TOE and has not been evaluated. The Installation, Generation and Startup (IGS) documentation ensures that once the administrator has followed the installation and configuration guidance the result is a TOE in a secure configuration. **AGD_ADM.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. **AGD_USR.1** is intended for non-administrative users, but could be used to provide guidance on security that is common to both administrators and non-administrators (e.g., password management guidelines). Since the non-administrative users of this TOE are limited to proxy users it is expected that the user guidance would discuss the secure use of proxies and how the single-use authentication mechanism is used. The use of the single-use authentication mechanism would not have to be repeated in the administrator's guide. **AVA_MSU.1** ensures that the guidance documentation is complete and consistent, and notes all requirements for external security

Objective	Requirements Addressing the Objective	Rationale
		measures.
O.ADMIN_ROLE: The TOE will provide an authorized administrator role to isolate administrative actions.	FMT_MOF.1 FMT_MTD.1 FMT_SMR.1	**FMT_SMR.1** requires that the TOE establish an Central Administrator role. **FMT_MOF.1** and **FMT_MTD.1** specify the privileges that only the Central Administrator may perform.
O.AUDIT_GENERATION: The TOE will provide the capability to detect and create records of security relevant events.	FAU_GEN.1-NIAP-0347 FAU_GEN.2-NIAP-0410	**FAU_GEN.1-NIAP-0347** 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 event that takes place in the TOE. This requirement also defines the information that must be contained in the audit record for each auditable event. This requirement also places a requirement on the level of detail that is recorded on any additional security functional requirements an ST author adds to this PP. **FAU_GEN.2-NIAP-0410** ensures that the audit records associate a user identity with the auditable event. In the case of authorized users, the association is accomplished with the userid. In all other cases, the association is based on the source network identifier, which is presumed to be the correct identity, but cannot be confirmed since these subjects are not authenticated.
O.AUDIT_PROTECT: The TOE will provide the capability to protect audit information.	FAU_SAR.2 FAU_STG.1-NIAP-0429 FAU_STG.NIAP-0414-1-NIAP-0429	**FAU_SAR.2** restricts the ability to read the audit trail to the Audit Administrator, thus preventing the disclosure of the audit data to any other user. However, the TOE is not expected to prevent the disclosure of audit data if it has been archived or saved in another form (e.g., moved or copied to an ordinary file). The FAU_STG family dictates how the audit trail is protected. **FAU_STG.1-NIAP-0429** restricts the ability to delete audit records to the Security Administrator. **FAU_STG.NIAP-0414-1-0429** defines the actions that must be available to the administrator, as well as the action to be taken if there is no response. This helps to ensure that audit records are kept until the Security Administrator deems they are no longer necessary. This requirement also ensures that no one has the ability to modify audit records (e.g., edit any of the information contained in an audit record). This ensures the integrity of the audit trail is maintained.
O.AUDIT_REVIEW: The TOE will provide the capability to selectively view audit information,	FAU_SAR.1 FAU_SAR.3	**FAU_SAR.1** and **FAU_SAR.3** provide the ability to review the audits in a user-friendly manner.

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.	ACM_CAP.2 ALC_FLR.2	**ACM_CAP.2** addresses this objective by requiring that 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 by uniquely identified. This provides a clear identification of the composition of the TOE. **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.CORRECT_TSF_OPERATION: The TOE will provide the capability to test the TSF to ensure the correct operation of the TSF at a customer's site.	FAU_GEN.1-NIAP-0347 FAU_GEN.2-NIAP-0410 FAU_SAR.1 FAU_SAR.3 FAV_SCN_EXP.1 FAV_ALR_EXP.1 FAV_ACT_EXP.1	Correct TSF operation can be determined by injecting a known virus into the TOE and ensuring that the proper events occur. The **FAV** class will detect and act upon the virus. The **FAU_GEN** family will generate an audit event when the virus is detected. The **FAU_SAR** family enables the administrator to review the audit events.
O.CRYPTOGRAPHY: The TOE shall use NIST FIPS 140-2 validated cryptographic services.	FCS_COP.1	FCS_COP.1 requires that the message digest used to verify integrity of the signature file utilize a FIPS 140-2 Approved cryptographic algorithm.
O.DOCUMENTED_DESIGN: The design of the TOE is adequately and accurately documented.	ADV_FSP.1 ADV_HLD.1 ADV_RCR.1	**ADV_FSP.1** requires that the interfaces to the TOE be documented and specified. **ADV_HLD.1** requires that the high level design of the TOE be documented and specified and that said design be shown to correspond to the interfaces. **ADV_RCR.1** requires that there be a correspondence between adjacent layers of the design decomposition.
O.MANAGE: The TOE will provide all the functions and facilities necessary to support the authorized users in their management of the TOE.	FMT_MOF.1 FMT_MTD.1 FMT_SMF.1 FMT_SMR.1	Restricted privileges are defined for the Central Administrator and Workstation Users. **FMT_MOF.1** defines particular TOE capabilities that may only be used by these users. **FMT_MTD.1** defines particular TOE data that may only be altered by these users. **FMT_SMF.1** and **FMT_SMR.1** define the administrative functions and roles provided by the TOE.
O.PARTIAL_FUNCTIONAL_TEST: The TOE will undergo some security functional testing that demonstrates the TSF satisfies some of its security functional requirements.	ATE_COV.1 ATE_FUN.1 ATE_IND.2	**ATE_FUN.1** requires that developer provide test documentation for the TOE, including test plans, test procedure descriptions, expected test results, and actual test results. These needs to identify the functions tested, the tests performed, and test scenarios. They require that the developer run those tests, and show that the expected results were achieved.

Objective	Requirements Addressing the Objective	Rationale
		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. **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.	FPT_SEP_EXP.1	The explicitly specific component **FPT_SEP_EXP.1** was chosen to 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. The explicitly specified version was used to distinguish the aspects of FPT_SEP provided by the TOE vs. the aspects provided by the IT environment.
O.VIRUS: The TOE will detect and take action against known viruses introduced to the workstation via network traffic or removable media.	FAV_ACT_EXP.1 FAV_ALR_EXP.1 FAV_SCN_EXP.1	**FAV_SCN_EXP.1** requires that the TOE scan for viruses. **FAV_ACT_EXP.1** requires that the TOE take action against viruses once they detected. **FAV_ALR_EXP.1** defines alerting requirements to ensure the users are aware that a virus was detected.
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_VLA.1	The **AVA_VLA.1** 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_VLA.1 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 moderate attack potential, which is in keeping with the desired assurance level of this TOE. As with the functional testing, a key element in 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 of moderate (or lower) attack potential to violate the TOE's security policies.

6.7 RATIONALE FOR ASSURANCE REQUIREMENTS

The EAL definitions and assurance requirements in Part 3 of the CC were reviewed and the *Basic Robustness Assurance Package* as defined in Section 5.3 was believed to best achieve the goal of addressing circumstances where developers and users require a low level of independently assured security in commercial products. The assurance package was selected because the TOE is an application executing on a system outside the TOE boundary, and basic is the highest robustness level available to application TOEs.

6.8 RATIONALE FOR DEPENDENCIES

Each functional requirement, including explicit requirements was analyzed to determine that all dependencies were satisfied. All requirements were then analyzed to determine that no additional dependencies were introduced as a result of completing each operation. Table 67 identifies the functional requirement, and its correspondent dependency, Table 6.8 provides the analysis and rationale for dependencies not required in this PP.

In Table 6.7, the "Component" column lists all of the components included in this PP; each one is assigned a unique ID number in the "ID" column. Each component's dependencies (from the CC) are listed in the "Dependency" column. The "Satisfied" column indicates how the dependencies are satisfied, with the number referencing the ID number of the component included in the PP that satisfies the dependencies. N/A is used when there are no dependencies for a component, and a reference to Table 6.8 is included when the dependency is not met but justified in Table 6.8.

Table 6.7 – Dependencies Table

ID	Component	Dependency	Satisfied
1	FAU_GEN.1-NIAP-0347	FPT_STM.1	26
2	FAU_GEN.2-NIAP-0410	FAU_GEN.1, FIA_UID.1	1 17
3	FAU_SAR.1	FAU_GEN.1	1
4	FAU_SAR.2	FAU_SAR.1	3
5	FAU_SAR.3	FAU_SAR.1	3
6	FAU_STG.1-NIAP-0429	FAU_GEN.1	1
7	FAU_STG.NIAP-0414-NIAP-0429	FAU_GEN.1, FAU_STG.1	1 6
8	FAV_ACT_EXP.1	FAV_SCN_EXP.1 FMT_SMR.1	10 21
9	FAV_ALR_EXP.1	FAV_SCN_EXP.1 FMT_SMR.1	10 21
10	FAV_SCN_EXP.1	FMT_SMR.1	21
11	FCS_COP.1	[FDP_ITC.1 or FCS_CKM.1], FCS_CKM.4, FMT_MSA.2	No – see following table for rationale
12	FDP_RIP.1	None	N/A
13	FIA_AFL.1	FIA_UAU.1	15
14	FIA_SOS.1	None	N/A

ID	Component	Dependency	Satisfied
15	FIA_UAU.2	FIA_UID.1	17
16	FIA_UAU.6	None	N/A
17	FIA_UID.2	None	N/A
18	FMT_MOF.1	FMT_SMF.1 FMT_SMR.1	20 21
19	FMT_MTD.1	FMT_SMF.1 FMT_SMR.1	20 21
20	FMT_SMF.1	None	N/A
21	FMT_SMR.1	FIA_UID.1	17
22	FPT_ITT.1	None	N/A
23	FPT_RVM.1	None	N/A
24	FPT_SEP.1	None	N/A
25	FPT_SEP_EXP.1	None	N/A
26	FPT_STM.1	None	N/A
27	FTA_SSL.1	FIA_UAU.1	15
28	FTA_TAB.1	None	N/A

Table 6.8 – Unsupported Dependency Rationale

Requirement	Dependency	Dependency Analysis and Rationale
FCS_COP.1	[FDP_ITC.1 or FCS_CKM.1], FCS_CKM.4, FMT_MSA.2	The only cryptographic function included in the PP is a message digest, which does not use keys.

6.9 RATIONALE FOR STRENGTH OF FUNCTION CLAIM

Part 1 of the CC defines "strength of function" in terms of the minimum efforts assumed necessary to defeat the expected security behavior of a TOE security function. There are three strength of function levels defined in Part 1: SOF-basic, SOF-medium and SOF-high. SOF-basic is the strength of function level chosen for this PP. SOF-basic states, "a level of the TOE strength of function where analysis shows that the function provides adequate protection against casual breach of TOE security by attackers possessing a low attack potential." The rationale for choosing SOF-basic was to be consistent with the Basic Robustness guidelines.

6.10 RATIONALE FOR EXPLICIT REQUIREMENTS

Table 6.9 presents the rationale for the inclusion of the explicit requirements found in this PP.

Table 6.9 – Rationale for Explicit Requirements

Explicit Requirement	Rationale
FAV_ACT_EXP.1	This component defines the actions to be taken by the TOE when a virus is detected. Existing security policy SFRs (e.g., FDP_ACF and FDP_IFF) focus on the access to or flow of user data and are not suitable for the actions taken by Anti-Virus products.

Explicit Requirement	Rationale
FAV_ALR_EXP.1	This component defines the alerting mechanism to be used to inform users when a virus is detected. The mechanism involves an acknowledgement from Workstation Users or Central Administrators that is not accounted for in CC SFRs.
FAV_SCN_EXP.1	This component defines the scanning to be performed by the TOE to detect viruses. Existing security policy SFRs (e.g., FDP_ACF and FDP_IFF) focus on the access to or flow of user data and are not suitable for the mechanisms used by Anti-Virus products.
FPT_SEP_EXP.1	The CC FPT_SEP component can not be satisfied by application TOEs. This component defines the separation that may be performed by applications. It is drawn from [BRCIM].

7 ACRONYMS

Table 7.1 – List of Acronyms

AM	Assurance Maintenance
BR CIM	Basic Robustness Consistency Instruction Manual
CC	Common Criteria
CM	Configuration Management
CMVP	Cryptographic Module Validation Program
DISA	Defense Information Services Agency
DoD	Department of Defense
EAL	Evaluation Assurance Level
FIPS	Federal Information Processing Standard
FOUO	For Official Use Only
FTP	File Transfer Protocol
GIG	Global Information Grid
HTTP	Hypertext Transport Protocol
I&A	Identification and Authentication
ID	Identification
IGS	Installation, Startup and Generation
IP	Internet Protocol
IT	Information Technology
LAN	Local Area Network
NIAP	National Information Assurance Partnership
NIPRNet	Non-classified Internet Protocol Router Network
NIST	National Institute of Standards and Technology
NSA	National Security Agency
PP	Protection Profile
PUB	Publication
RFC	Request for Comments
SFP	Security Function Policy
SIPRNet	Secret Internet Protocol Router Network
SOF	Strength of Function
SMTP	Simple Message Transfer Protocol

SSL	Secure Socket Layer
ST	Security Target
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TOE	Target of Evaluation
TP	Trusted Path
TSC	TSF Scope of Control
TSF	TOE Security Functions
TSFI	TSF Interface
TSP	TOE Security Policy
UDP	User Datagram Protocol

8 REFERENCES

8.1 REQUIREMENTS REFERENCES

[BRCIM] Consistency Instruction Manual For Development of US Government Protection Profiles (PP) For use in Basic Robustness Environments, NIAP Protection Profile Review Board, Release 2.0, 1 March 2004.

{This page intentionally left blank}

9 TERMINOLOGY

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

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

Access Control Information (ACI) — Information stored in the directory that is used to determine which users have been granted access to directory objects and what type of access has been granted (e.g., read, write).

Access Control Decision Function — A specialized function that makes access control decisions by applying access control policy rules to an access request.

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

Administrator — A user who has been specifically granted the authority to manage the TOE or a subset of the TOE, and whose actions may affect the TSP. Administrators may possess special privileges that provide capabilities to override portions of the TSP.

Application Note — Supporting information that is considered relevant or useful for the construction, evaluation, or use of the TOE.

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

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

Attack Potential — The perceived potential for success of an attack, should an attack be launched, expressed in terms of an attacker's expertise, resources and motivation.

Attribute — A property that is associated with an entry. Attributes may be of a user type or operational type. User attributes are those attributes accessible by users. Operational attributes are attributes used by the directory and not accessible by users. An attribute is made up of attribute values and attribute type. The attribute type defines how the attribute value is used and processed. Attributes may be mandatory or optional.

Audit — To conduct an internal or independent review and assessment of records and/or activities.

Auditor — Role required by the TOE for a type of Administrative user that is given privileges commensurate with performing audit functions.

Authentication — Security measure that verifies a claimed identity.

Authentication Data — Information used to verify a claimed identity.

Authority Revocation List — See Revocation List.

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

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

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

[1] Hardware and Software

[2] Stored or communicated.

[3] According to a defined metric.

Basic Access Control — One of three X.500-defined access control schemes for the directory. It is defined in 1997 version of X.501.

Black Box — An abstraction of a device or system in which only its externally visible behaviour is considered and not its implementation or "inner workings".

Common Criteria — The Common Criteria represents the outcome of a series of efforts to develop criteria for evaluation of IT security that are broadly useful within the international community.

Compromise — Violation of a security policy.

Confidentiality — A security policy pertaining to disclosure of data.

Connectivity — The property of the TOE that allows interaction with IT entities external to the TOE. This includes exchange of data by wire or by wireless means, over any distance in any environment or configuration.

Console — A combination of keyboard and screen connected to an operating system port specified for administrator access. Historically this was limited to a hard-wired character-only terminal connected to a serial port.

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

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

Cryptographic Algorithm — Asymmetric: A cryptographic algorithm that uses two related keys, a public key and a private key. The two keys have the property that, given the public key, it is computationally infeasible to derive the private key.

Cryptographic Algorithm — Symmetric: A cryptographic algorithm that uses a single, secret key for both encryption and decryption.

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

Cryptographic Key (key) — A parameter used in conjunction with a cryptographic algorithm that determines:
the transformation of plaintext data into ciphertext data,
the transformation of cipher text data into plaintext data,
a digital signature computed from data,
the verification of a digital signature computed from data, or
a digital authentication code computed from data.

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

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

DAT File — A file containing the known signatures scanned for by anti-virus applications.

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

Dependency — A relationship between requirements such that the requirement that is depended upon must normally be satisfied for the other requirements to be able to meet their objectives.

Digital Signature — A non-forgeable transformation of data that allows proof of the source and verification of the integrity of that data.

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

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

Encrypted Channel — A communications channel connecting the TOE to an outside IT entity that has been secured to prevent disclosure of information in the channel.

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

Evaluation Assurance Level (EAL) — A package consisting of assurance components from Part 3 that represents a point on the CC predefined assurance scale.

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.

Human User — Any person who interacts with the TOE.

Intrusion Detection System (IDS) — An example of a trusted external IT entity that identifies events that that may be indicative of an attack on a system. There are various types of IDS including network based IDS, platform based IDS, etc.

Internet Engineering Task Force (IETF) — Open international community concerned with the evolution of the Internet architecture technologies.

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

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

Named Object[4] — An object that exhibits all of the following characteristics:

The object may be used to transfer information between subjects of differing user identities within the TSF.

Subjects in the TOE must be able to request a specific instance of the object.

The name used to refer to a specific instance of the object must exist in a context that potentially allows subjects with different user identities to request the same instance of the object.

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

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

To the sender of data, proof of delivery to the intended recipient,

To the recipient of data, proof of the identity of the user who sent the data.

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

Object — An entity within the TSC that contains or receives information and upon which subjects perform operations. Examples include a RI entry, attribute, or object class.
Operating Environment — The total environment in which a TOE operates. It includes the physical facility and any physical, procedural, administrative and personnel controls.
Organizational Security Policies — One or more security rules, procedures, practices, or guidelines imposed by an organization upon its operations.
Package — A reusable set of either functional or assurance components (e.g. an EAL), combined together to satisfy a set of identified security objectives.
Password — A string of characters (letters, numbers, and other symbols) used to authenticate an identity or to verify access authorization.
Platform — Typically a device that includes the hardware and software elements that support all or part of the functional requirements of the TOE applications.
Precedence Levels — Predetermined levels of importance used in access control decisions.
Product — A package of IT software, firmware and/or hardware, providing functionality designed for use or incorporation within a multiplicity of systems.
Protected Items — Data in the TOE that is protected using access control mechanisms.
Protection Profile (PP) — An implementation-independent set of security requirements for a category of TOEs that meet specific consumer needs.
Refinement — The addition of details to a component.
Remote Trusted User — A trusted user or trusted external IT entity that accesses the directory from a location outside the boundary of the TOE.
Replay — An attack in which a third party captures a command in transmission and replays it at a later time.
Robustness — A characterization of the strength of a security function, mechanism, service or solution, and the assurance (or confidence) that it is implemented and functioning correctly. DoD has three levels of robustness:
Basic: Security services and mechanisms that equate to good commercial practices. Basic robustness equates to EAL-2 augmented by ALC_FLR (Flaw Remediation) as defined in CCIB-98-028, Part 3, Version 2.0 and AVA_MSU.1.
Medium: Security services and mechanisms that provide for layering of additional safeguards above good commercial practices. Medium robustness equates to ACM_AUT.1, ACM_CAP.4, ACM_SCP.2, ADO_DEL.2, ADO_IGS.1, ADV_ARC_EXP.1, ADV_FSP_EXP.1, ADV_HLD_EXP.1, ADV_INT_EXP.1, ADV_IMP.2, ADV_LLD_EXP.1, ADV_RCR.1, ADV_SPM.1, AGD_ADM.1, AGD_USR.1, ALC_DVS.1, ALC_FLR.2, ALC_LCD.1, ALC_TAT.1, ATE_COV.2, ATE_DPT.2, ATE_FUN.1, ATE_IND.2, AVA_CCA_EXP.2, AVA_MSU.2, AVA_SOF.1, AVA_VLA.3 as defined in CCIB-98-028, Part 3, Version 2.0
High: Security services and mechanisms that provide the most stringent protection and rigorous security countermeasures.
Role — A predefined set of rules establishing the allowed interactions between a user and the TOE.
Secret — Information that must be known only to authorized users and/or the TSF in order to enforce a specific SFP.
Secure State — Condition in which all TOE security policies are enforced.
Security Administrator — Role supported by the TOE, which is a type of Administrative user that is given privileges commensurate with maintaining the security-related functionality of the

TOE. Security Administrators may be responsible for security functions on both the platform and the directory.

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

Security Policy — A precise specification of the security rules under which the TOE shall operate, including the rules derived from the requirements of this document and additional rules imposed by the vendor.

Security Target (ST) — A set of security requirements and specifications to be used as the basis for evaluation of an identified TOE.

Selection — The specification of one or more items from a list in a component.

SOF-basic — A level of the TOE strength of function where analysis shows that the function provides adequate protection against casual breach of TOE security by attackers possessing a low attack potential.

SOF-high — A level of the TOE strength of function where analysis shows that the function provides adequate protection against deliberately planned or organized breach of TOE security by attackers possessing a high attack potential.

SOF-medium — A level of the TOE strength of function where analysis shows that the function provides adequate protection against straightforward or intentional breach of TOE security by attackers possessing a moderate attack potential.

Strength of Function (SOF) — A qualification of a TOE security function expressing the minimum efforts assumed necessary to defeat its expected security behavior by directly attacking its underlying security mechanisms.

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

System — A specific IT installation, with a particular purpose and operational environment.

Target of Evaluation (TOE) — An IT product or system and its associated administrator and user guidance documentation that is the subject of an evaluation.

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.

Time stamp — Electronic seal including a time and/or date indication applied over data.

TOE resource — Anything useable or consumable in the TOE.

TOE Security Functions (TSF) — A set consisting of all hardware, software, and firmware of the TOE that must be relied upon for the correct enforcement of the TSP.

TOE Security Functions Interface (TSFI) — A set of interfaces, whether interactive (man-machine interface) or programmatic (application programming interface), through which TOE resources are accessed, mediated by the TSF, or information is obtained from the TSF.

TOE Security Policy (TSP) — A set of rules that regulate how assets are managed, protected and distributed within a TOE.

Trusted — Used to describe any user or IT entity that is authenticated to the TOE with some level of assurance.

Trusted channel — A means by which a TSF and a remote trusted IT product can communicate with necessary confidence to support the TSP.

Trusted path — A means by which a user and a TSF can communicate with necessary confidence to support the TSP. A mechanism by which a trusted user can communicate directly and reliably with the directory and that can only be activated by the user and cannot be imitated by untrusted software.

TSF data — Data created by and for the TOE that might affect the operation of the TOE.

TSF Scope of Control (TSC) — The set of interactions that can occur with or within a TOE and are subject to the rules of the TSP.

User — Any entity (human user or external IT entity) outside the TOE that interacts with the TOE.

User Class — A schema used for determining the rules to be applied to a relying party when deciding the users permissions to the requested protected item (access control decision). Users can be granted permissions based on their distinguished name, identity, subtree information, etc.

User Data — Data created by and for the user that does not affect the operation of the TSF.

User Group — Group that further identifies users in a system.

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