IronMail Secure Email Gateway Software Version 4.0.0 Security Target

April 27, 2006
Document No. CipherTrust E2-IM4.0.0

CipherTrust
4800 North Point Parkway
Suite 400
Alpharetta, GA 30022
Phone: 678-969-9399
Fax: 678-969-9398

DOCUMENT INTRODUCTION

This document provides the basis for an evaluation of a specific Target of Evaluation (TOE), the IronMail Secure Email Gateway Software Version 4.0.0. This Security Target (ST) defines a set of assumptions about the aspects of the environment, a list of threats that the product intends to counter, a set of security objectives, a set of security requirements and the IT security functions provided by the TOE which meet the set of requirements.

REVISION HISTORY

Rev	Description
	October 2, 2003, initial release
	April 27, 2004, Updated release for RVP releasing naming convention
	October 7, 2004, Clarified versioning of appliance and software
	March 28, 2006, Updated Table 8 Assurance Correspondence
	April 4, 2006, Updated Figure 1 Typical TOE Deployment
	April 21, 2006, Updated Figure 1 and added explanatory text
	April 27, 2006, Updated explanatory text for Figure 1

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS LIST

CC.....**Common Criteria**
DNS.....**Domain Name System**
EAL2 ..... **Evaluation Assurance Level 2**
IMAP.....**Internet Message Access Protocol**
IT .....**Information Technology**
MIME..... **Multipurpose Internet Mail Extensions**
NIAP.....**National Information Assurance Partnership**
PP.....**Protection Profile**
POP.....**Post Office Protocol**
SF.....**Security Function**
SFP ..... **Security Function Policy**
SMTP .....**Simple Mail Transfer Protocol**
SOF.....**Strength of Function**
ST..... **Security Target**
TOE .....**Target of Evaluation**
TSC..... **TSF Scope of Control**
TSF .....**TOE Security Functions**
TSFI..... **TSF Interface**
TSP ..... **TOE Security Policy**

CHAPTER 1

1. Security Target Introduction

This Security Target (ST) describes the objectives, requirements and rationale for the IronMail Secure Email Gateway Software Version 4.0.0. The language used in this Security Target is consistent with the *Common Criteria for Information Technology Security Evaluation, Version 2.1*, the ISO/IEC JTC 1/SC27, *Guide for the Production of PPs and STs, Version 0.9* and all National Information Assurance Partnership (NIAP) and international interpretations through September 9, 2003. As such, the spelling of terms is presented using the internationally accepted English.

1.1 Security Target Reference

This section provides identifying information for the IronMail Secure Email Gateway Software Version 4.0.0 Security Target by defining the Target of Evaluation (TOE).

1.1.1 Security Target Name

IronMail Secure Email Gateway Software Version 4.0.0 Security Target.

April 27, 2006

1.1.2 Authors

This Security Target was prepared by CipherTrust, Inc.

1.1.3 TOE Reference

IronMail Secure Email Gateway Software Version 4.0.0.

1.1.4 Security Target Evaluation Status

This ST is currently under evaluation.

1.1.5 Evaluation Assurance Level

Assurance claims conform to EAL2 (Evaluation Assurance Level 2) from the *Common Criteria for Information Technology Security Evaluation, Version 2.1*.

1.1.6 Keywords

Email, Spam filtering, Content filtering.

1.2 TOE Overview

This Security Target forms the basis of evaluation for the IronMail Secure Gateway Email Software Version 4.0.0. The TOE resides in an all-inclusive device positioned at the network gateway and is used for protecting organisations from email threats such as spam, liabilities arising from offensive content present in email messages and general mail policy violations. The TOE processes incoming messages through a number of filtering queues, which check the content of the messages for compliance against relevant organisational policies. Only those messages that have not been filtered by any queue are delivered to their destination. Messages may be selectively forwarded, quarantined or saved in order to facilitate forensic examination by third party tools.

1.2.1 Security Target Organisation

Chapter 1 of this ST provides introductory and identifying information for the TOE.

Chapter 2 describes the TOE and provides some guidance on its use.

Chapter 3 provides a security environment description in terms of assumptions, threats and organisational security policies.

Chapter 4 identifies the security objectives of the TOE and of the Information Technology (IT) environment.

Chapter 5 provides the TOE security and functional requirements, as well as requirements on the IT environment.

Chapter 6 is the TOE Summary Specification, a description of the functions provided by the IronMail Secure Gateway Email Software Version 4.0.0 to satisfy the security functional and assurance requirements.

Chapter 7 identifies claims of conformance to a registered Protection Profile (PP).

Chapter 8 provides a rationale for the security objectives, requirements, TOE summary specification and PP claims.

1.3 Common Criteria Conformance

This Security Target is compliant with the *Common Criteria for Information Technology Security Evaluation, Version 2.1*, the ISO/IEC JTC 1/SC27, *Guide for the Production of PPs and STs, Version 0.9,* and all National Information Assurance Partnership (NIAP) and international interpretations through October 17, 2003. This Security Target is functional requirements (Part 2 of CC) conformant and assurance requirements (Part 3 of CC) conformant for EAL2.

1.4 Protection Profile Conformance

The IronMail Secure Gateway Email Software Version 4.0.0 Security Target does not claim conformance to any registered Protection Profile.

1.5 Document Conventions

The CC defines four operations on security functional requirements. The font conventions below identify the conventions for the operations defined by the CC.

Assignment:	*italicized text contained within the block beginning with the text* [assignment:]
Selection:	*italicized text contained within the block beginning with the text* [selection:]
Refinement:	***indicated with bold text and italics***
Iteration:	indicated with typical CC requirement naming followed by a number in parenthesis for each iteration (e.g., FAU_ARP.1 (1))

CHAPTER 2

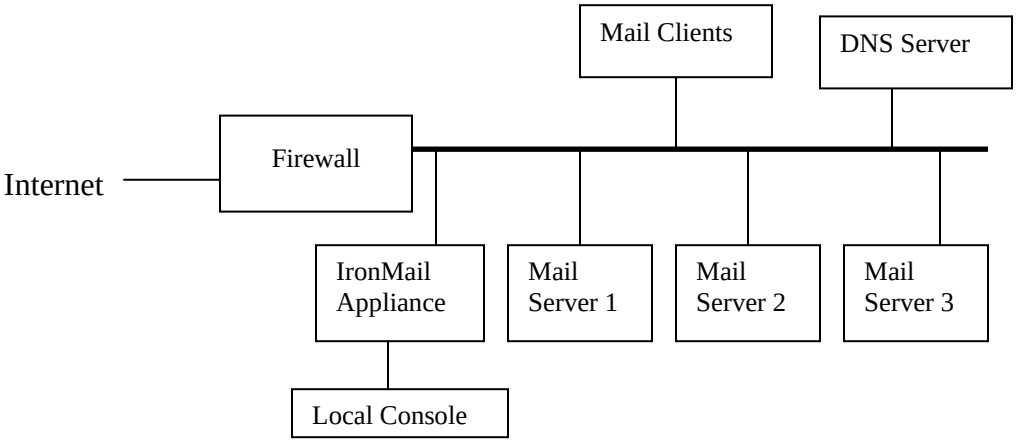
2. TOE Description

This section provides the context for the TOE evaluation by identifying the product type and describing the evaluated configuration.

2.1 IronMail Secure Gateway Email Software Version 4.0.0 TOE Description

The TOE provides an integrated solution for countering email threats. It resides within an integrated device, hereafter referred to as the IronMail appliance. RFC 822, Multipurpose Internet Mail Extensions (MIME) encoded messages are checked for policy violations and the presence of offensive content. Any message that violates the TOE's notion of security is isolated and acted upon so as to mitigate any threat being posed by it before it reaches the internal network. The TOE is also able to detect and curtail the flow of spam into the internal network in order to ensure the availability of system resources such as storage space and CPU time. The TOE relies on information obtained from DNS in order to detect spam; the DNS server can reside on any host on the internal network and is assumed to always provide reliable information to the TOE.

Figure 1 - Typical TOE deployment



Typically the TOE is not in the physical path between the various participants in email exchanges (mail clients and servers, both internal and external). However, logically the IronMail Appliance mediates all email exchanges. This requires the clients and servers to be configured to forward all email traffic through the IronMail Appliance, routers and firewalls to redirect all email traffic through the IronMail Appliance, or a combination of both.

The TOE is based on a fully functional mail server engine and a queuing architecture designed to parse and scan messages easily and quickly. TOE queues are subsystems that process messages in an ordered fashion. The queuing architecture scrutinises every message received for harmful content. Every queue wakes up at periodic intervals to see if new messages are available for it to process. The wake up cycle is dynamic - as the TOE's message load increases, the queues wake up after shorter sleep cycles in order to process messages more quickly. After each queuing subsystem finishes processing the message it passes it on to the next queue in line. Each queue is designed to perform a specific task; once every queue has finished processing the message, the TOE delivers it

to its final destination assuming that no queue had to quarantine, drop, re-route, or take some other action on the message.

The TOE can be configured to recognise multiple administrators, each of whom is granted privileges to configure selected or all components of the TOE. These administrators may access the TOE either locally through an attached console or remotely through a secured web connection. Administrators may also access the command line interface from a workstation using an SSH (Secure Shell) application on port 22. In either case the user is authenticated by the Operating System. Once authenticated, administrators can use the relevant TSF Interface (TSFI) to configure the behaviour of different TOE subsystems by defining rules that identify spam, malicious content and policy violations. Rules may also be defined to allow specific messages to bypass the various filtering queues.

The TOE provides a logging component that allows the authorised administrator of the TOE to monitor the behaviour of the TOE and its different subsystems. Log records are generated for events such as policy matches and configuration changes made on the TOE. TOE "health" can be monitored through logs generated by the TOE's internal monitoring and notification subsystems.

The TOE runs over "hardened" NetBSD and runs trusted software -- it can verify the integrity of the code that it executes and data that it consumes, by periodically checking program and file system integrity. The TOE constantly monitors the operation of its internal components; any component that fails is automatically restarted. Notifications are sent to the configured users alerting them to the occurrences of various events in the system in order to facilitate timely remedial action.

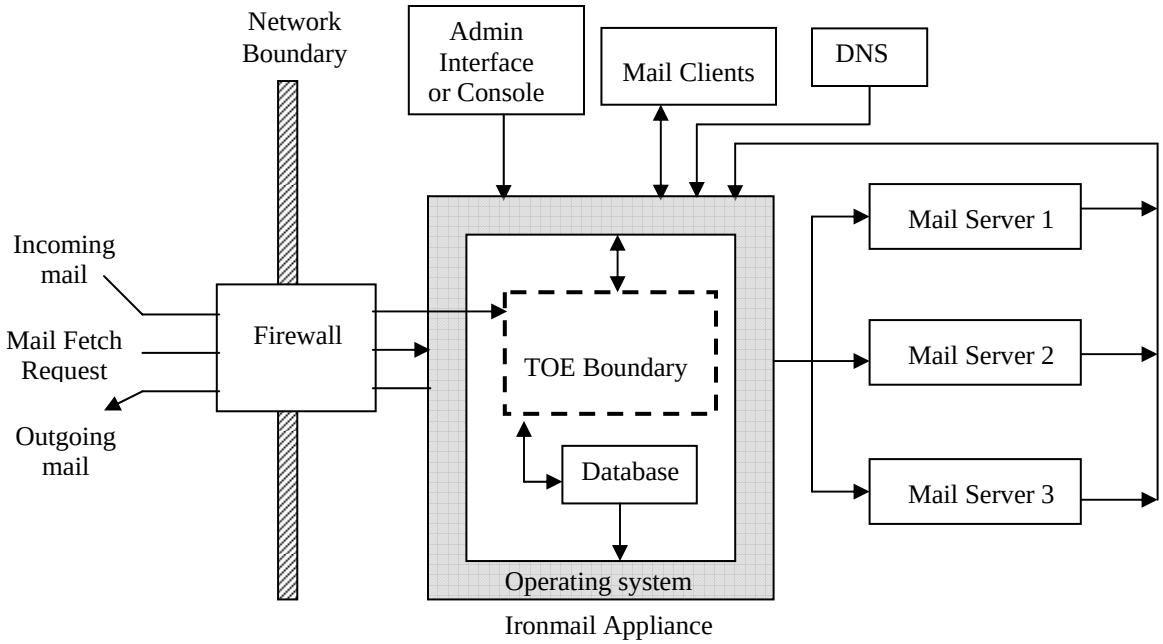
Mail clients fetch mail from the mail servers via POP3 (Post Office Protocol version 3), IMAP4 (Internet Message Access Protocol Version 4) and their secure variants (POP3S and IMAP4S) by proxying these connections through the IronMail appliance. When users attempt to retrieve their email, the IronMail appliance processes the requests and proxies them to the internal mail server(s). It passes the username and password to the internal mail server, which is responsible for doing the actual validation of the request. If validated, the IronMail appliance proxies the internal mail servers' response back to the client. The IronMail appliance can also proxy and encrypt the web sessions for users who ordinarily would have connected directly to the internal web-enabled mail servers.

2.1.1 Physical Boundary

This relationship between the TOE and the various network components is depicted in Figure 2. The TOE is positioned at the network gateway between the firewall and the mail servers. Every mail that enters the internal network first passes through the TOE. Similarly, only the TOE can deliver outgoing messages.

Internal mail clients interface with the TOE instead of interfacing with the mail servers directly. The IT Environment is configured such that the TOE is the only entry point for email messages to the mail servers irrespective of whether these messages originate inside or outside the internal network.

Figure 2 - Physical Boundary



The TOE runs over a hardened operating system and interfaces with a database that ensures high integrity and speed. Intermediate results from the different queues are stored and then retrieved by subsequent queues from this internal database.

Mail clients do not interface directly with the mail servers for fetching mail; instead they connect to the IronMail appliance, which proxies their requests on to the actual mail servers. The results are similarly proxied back to the client. The IronMail appliance can also encrypt outgoing email messages to ensure the confidentiality of mails sent across domains. The capability for doing secured delivery of email and the secure channels needed to communicate requests and responses between the mail clients, the TOE and the Mail Servers are all handled outside the TOE by the Operating System.

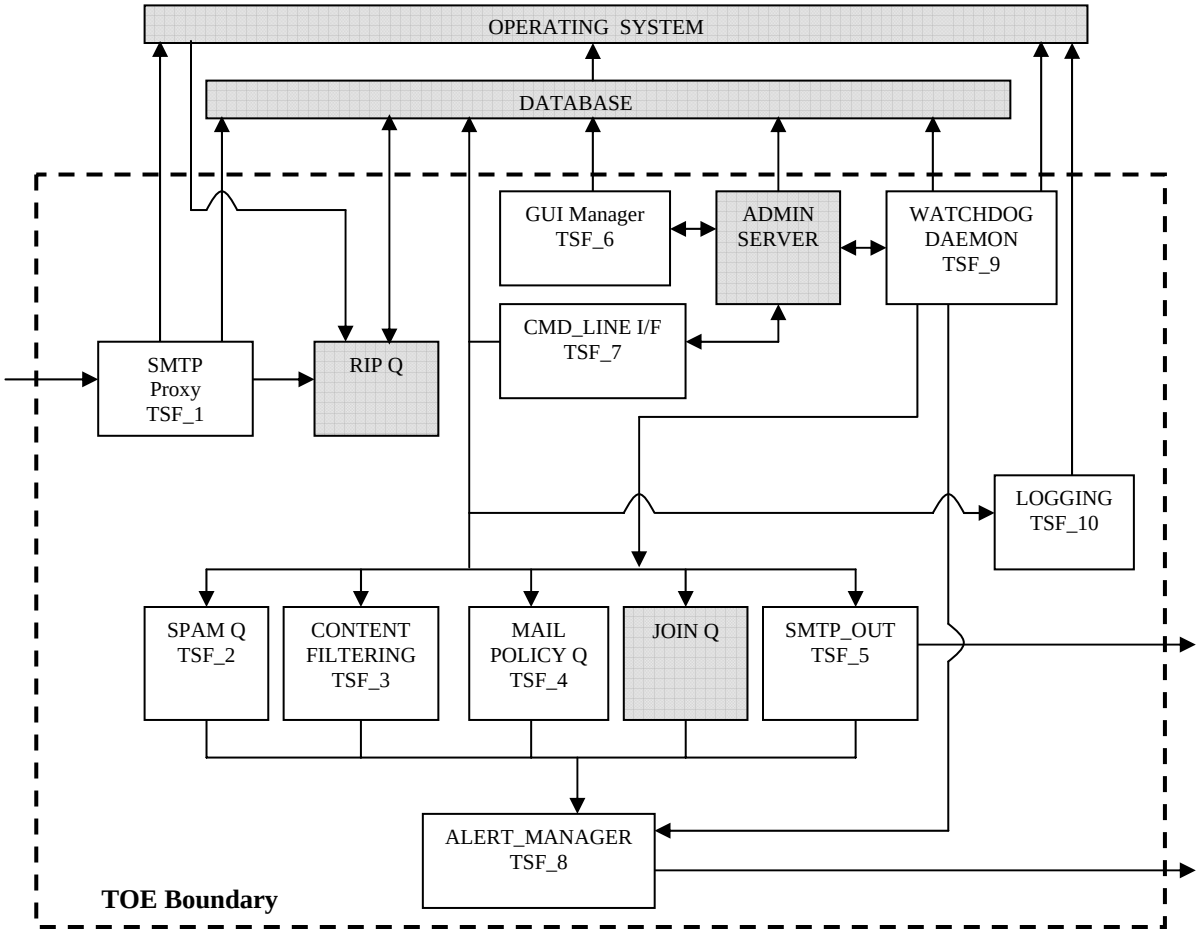
Administrators can either connect to the IronMail appliance locally using the attached console, or remotely using a secure connection from a web browser or a command line interface via SSH, in order to maintain and monitor the TOE's operation. Authentication services and the creation of the secure connection between the two communicating entities are again provided by the Operating System.

2.1.2 Logical Boundary

The logical boundary of the TOE is as shown in Figure 3. Essential TOE components are represented as TOE Security Functions (TSFs) and are numbered from 1 through 10. Internal TOE subsystems that are themselves not responsible for any Security Function Policy (SFP) are unnumbered.

The first component in the queuing framework that receives an incoming message is the Simple Mail Transfer Protocol (SMTP) proxy (TSF_1). This component is responsible for listening to incoming email messages and beginning the subsequent queue processing.

Figure 3 - Logical Boundary



The TSFs that are exposed to the administrator are outlined below.

1. The SPAM_Q (TSF_2) inspects messages for characteristics of spam. When a message is found to be spam-like, an administrator-defined action such as drop, quarantine or rename is performed on it.
2. The CONTENT_FILTERING_Q (TSF_3) inspects messages for the presence of inappropriate content. Content filtering is also enforced over the message attachments to ensure that messages containing specified attachment types do not pass through the TOE.
3. The MAIL_POLICY_Q (TSF_4) allows the monitoring of messages over selection criteria such as the sender, recipient and subject information.

In order to ensure that data is present in a form that can always be understood by the various filters, the TOE makes use of two internal queues that handle parsing, decoding and final re-assembly of the message.

1. The RIP Q is the first to process an email message. Its task is to "rip" the message into its constituent MIME parts. It stores the original message in its internal message store, and copies each message part to an internal database where it is picked up by subsequent queues for spam, content and mail policy filtering. Messages that cannot be parsed are directly pushed to the JOIN Q, where it can either be dropped or repackaged and sent to the original or an alternate email address.
2. The JOIN Q is the last to process an email message. Its task is to reassemble the message back into a whole. If any of the intermediate queues perform an action such as a rewrite of the subject line or deletion of an offensive word, the Join Queue deletes the original message from the internal message store, reassembles the message from the TOE-edited parts stored in the database and sends it to the SMTP Outbound Service for final delivery.

The SMTPO Service (TSF_5) is responsible for delivery messages out of the TOE. It is the only means by which messages can be sent out of the network domain.

The GUI_MANAGER (TSF_6) and CMD_LINE_I/F (TSF_7) components provide graphical and command line interfaces for the authorised users of the TOE to configure and maintain the TOE.

The WATCHDOG_DAEMON (TSF_9) and the ALERT_MANAGER (TSF_8) monitor and notify the authorised administrator about events that occur during TOE operation. The Admin Server is responsible for managing processes running on the TOE. It communicates with the authorised users of the TOE through the TSF_6 GUI Manager and TSF_7 CMD_LINE_I/F, and internally with the various queue processes (TSF_2 – TSF_5) and the database. The Admin Server has the ability to start and stop queue processes based on requests from the GUI Manager (TSF_6) or the Watchdog Daemon (TSF_9).

The LOGGING (TSF_10) component provides auditing support for the TOE.

2.2 IronMail Secure Email Gateway Software Version 4.0.0 Evaluated Configuration

The evaluated physical network configuration for the IronMail appliance is a device situated between the firewall and the internal mail servers, located and configured in a manner which ensures that every message that is sent into or out of the internal network always passes through it.

IronMail uses "hardened" NetBSD as its Operating System and MySQL as its internal database, both of whose security requirements are not in the TOE Scope of Control (TSC). Authentication services, domain separation and secure delivery services are provided by the hardened OS and are consequently not part of the TOE either. However, the management of roles is done directly by the TOE and is therefore mapped to appropriate security functional components.

Trusted channels, implicit with the mail client requirements for POP3S and IMAP4S and secured delivery of email messages out of the TOE are provided by the OS and are therefore not within the TSC.

IronMail Version 4.0.0 also includes the following subsytems that are out of the scope of this Security Target.

1. Mail Intrusion Detection
2. Anti-virus Queue
3. Anomaly Detection Engine
4. Application Inspection Engine

Some subsystems such as the MAIL_POLICY_Q (TSF_4) have licensing requirements before they can be used in an operational environment. This ST assumes that such licenses have already been procured.

CHAPTER 3

3. Security Environment

3.1 Introduction

This chapter identifies the following:

A) Significant assumptions about the TOE's operational environment.

B) Information Technology related threats to the organisation countered by the TOE.

C) Environmental threats requiring controls to provide sufficient protection.

D) Organisational security policies for the TOE as appropriate.

Using the above listing, this chapter identifies assumptions (A), threats (T) and organisational security policies (P).

3.2 Assumptions

The specific conditions listed in the following subsections are assumed to exist in the TOE environment. These assumptions include both practical realities in the development of the TOE security requirements and the essential environmental conditions on the use of the TOE.

3.2.1 Connectivity Assumptions

A.DB_INTEGRITY	The integrity of data maintained by the MySQL database is always ensured.
A.DNS	DNS information received by the TOE is reliable.

3.2.2 Personnel Assumptions

A.NO_EVIL_ADMIN	Authorized administrators are non-hostile and are appropriately trained to use, configure and maintain the TOE.

3.2.3 Physical Assumptions

A.PHYSICAL_SECURITY	The TOE resides in a physically controlled access facility that prevents unauthorized physical access.

3.3 Threats

3.3.1 Threats Against the TOE

T.BYPASS	A threat agent may bypass one or more of the TOE's security functions and send malicious data to mail servers being protected by the TOE.
T.COMP_FAILURE	A threat agent may take advantage of unexpected termination of one or more of the TOE's Security Functions (SF), and send inappropriate information through the TOE in violation of its mail policy.

T.CONTENT	A threat agent may circulate dirty, offensive or proprietary information in violation of the TOE policy.
T.NEW_EXPLOITS	A threat agent may modify the message content suitably or use variants in the sender or recipient information in order to defeat the protection services offered by the TOE.
T.NO_AUDIT	A threat agent may perform security relevant operations on the TOE without being held accountable for it.
T.NO_REGULATE	A threat agent may try to violate the mail dissemination policy of the TOE by sending information that the TOE may not want to forward or receive, either because of its origin, destination or subject content.
T.OPAQUE	A threat agent may send malicious content in an encrypted form in order to violate the TOE's content distribution policy.
T.RESOURCE_CONSUME	Threat agents may flood the TOE with spam, consuming resources such as memory, bandwidth, processor time and data storage and thus limit the TOE's ability to execute its security functions efficiently.
T.UNTRUSTED_CODE	A threat agent may download untrusted code to the TOE causing abnormal processes to be executed, which violate the integrity and availability of system assets.

3.3.2 Threats Against the TOE Operational Environment

T.E.AUTH_CAPTURE	A threat agent may execute a process on the TOE that captures the authentication data of a valid user of the TOE in order to gain unauthorized access to the TOE.
T.E.BRUTE_FORCE	A threat agent may repeatedly try and guess authentication data in order to gain unauthorized access to the TOE.
T.E.EXT_CAPTURE	An external attacker may sniff the communication channel between end-user mail clients and the TOE in order to capture or modify messages and authentication data sent between the two.
T.E.IA	A threat agent may attempt to compromise the TOE by attempting actions that it is not authorized to perform on the TOE.
T.E.INT_CAPTURE	A malicious insider may sniff the communication channel between the TOE and the internal mail servers in order to capture or modify messages and authentication data sent between the two.
T.MASQUERADE	A threat agent masquerading as the TOE may capture valid identification and authentication data for a legitimate user

of the TOE in order to gain unauthorized access to the TOE.

3.4 Organisational Security Policies

None

CHAPTER 4

4. Security Objectives

4.1 Security Objectives for the TOE

All of the objectives listed in this section ensure that all of the security threats listed in Chapter 3 have been countered. The security objectives (O) for the IronMail Secure Email Gateway Software Version 4.0.0 are:

O.CONFIGURABILITY	The TOE shall provide administrative tools to enable authorised administrators to effectively configure and maintain the TOE.
O.CONTENT_FILTER	The TOE shall take specified action on incoming messages based on their message or attachment content.
O.LOG	The TOE shall generate logs of all the security-relevant operations performed on the TOE.
O.MAIL_POLICY	The TOE shall be able to prevent specific types of information sent to or from specific entities, from passing through the TOE.
O.NOTIFICATION	The TOE shall generate and deliver alerts upon detecting failure of any of its functional components.
O.REF_MEDIATION	All inbound or outbound mail into or out of the TOE, unless explicitly allowed by the TOE administrator, shall be examined by each of the TOE's configured filters before being forwarded to its destination.
O.SPAM_FILTER	The TOE shall be able to define characteristics for spam and take configured action when such characteristics are recognised.
O.TOE_INTEGRITY	The integrity of the TOE trusted code base shall be ensured at all times.

4.2 Security Objectives for the IT Environment

O.E.AUTHENTICATION	The IT Environment shall require that users of the TOE be identified and authenticated before allowing any TSF-mediated activity to be performed by them.
O.E.BOUNDED_AUTH	The IT Environment shall bound the number of failed authentication attempts to some configurable value in order to prevent brute force attacks against the TOE.
O.E.DOMAIN_SEP	The IT Environment shall ensure that the execution of code within the TOE cannot be interfered with or tampered by any untrusted subject.
O.E.EXT_CHAN	The IT Environment shall ensure that messages and authentication data sent between external mail clients and

	the TOE is protected from unauthorised disclosure and modification.
O.E.INT_CHAN	The IT Environment shall ensure that messages and authentication data sent between the TOE and the internal mail servers is protected from unauthorised disclosure and modification.
O.E.NO_BYPASS	There shall be no possible way for messages to reach the protected network without first being processed by the TOE.
O.E.TRUSTED_ENV	The TOE shall reside at a physically secure location, safe from compromise by malicious insiders or outsiders.
O.E.TRUSTED_INFO	The integrity of the information received by the TOE from trusted external subsystems shall never be compromised.
O.E.TRUSTED_PATH	The IT Environment shall maintain a trusted path for allowing authorised users of the TOE to identify and authenticate themselves to it.
O.E.TS_INTEGRITY	The IT Environment shall ensure the reliability of timestamps exported to the TOE.

CHAPTER 5

5. IT Security Requirements

This section contains the functional requirements that are provided by the TOE. These requirements consist of functional components from Part 2 of the CC.

5.1 TOE Security Functional Requirements

Table 1 - Functional Components of the TOE

CC Component	Name	Dependency
FAU_ARP.1(1)	Security Alarms ***for Spam Detection***	FAU_SAA.1(1)
FAU_ARP.1(2)	Security Alarms ***for Content Match***	FAU_SAA.1(2)
FAU_ARP.1(3)	Security Alarms ***for Mail Policy Violation***	FAU_SAA.1(3)
FAU_ARP.1(4)	Security Alarms ***for Encrypted Mail Policy Violation***	FAU_SAA.1(4)
FAU_ARP.1(5)	Security Alarms ***for System Alert Notification***	FAU_SAA.1(5)
FAU_GEN.1	Audit Data Generation	FPT_STM.1, satisfied in the environment
FAU_SAA.1(1)	Potential Violation ***Analysis for Spam Detection***	FAU_GEN.1
FAU_SAA.1(2)	Potential Violation Analysis ***for Content Match***	FAU_GEN.1
FAU_SAA.1(3)	Potential Violation Analysis ***for Mail Policy***	FAU_GEN.1
FAU_SAA.1(4)	Potential Violation Analysis ***for Encrypted Mail Policy***	FAU_GEN.1
FAU_SAA.1(5)	Potential Violation Analysis ***for System Alert Notification***	FAU_GEN.1
FAU_SAR.1	Audit Review	FAU_GEN.1
FAU_SEL.1	Selective Audit	FAU_GEN.1 FAU_MTD.1
FAU_STG.1	Protected Audit Trail Storage	FAU_GEN.1
FMT_MOF.1(1)	Management of Security Functions behavior	FMT_SMR.1
FMT_MOF.1(2)	Management of Security Functions	FMT_SMR.1

	behavior	
FMT_MTD.1	Management of TSF data	FMT_SMR.1
FMT_SMF.1	Specification of Management Functions	None
FMT_SMR.1	Security Roles	FIA_UID.1, satisfied in the environment
FPT_RVM.1(1)	Non-bypassability of the TSP	None
FPT_TST.1	TSF Self Test	FPT_AMT.1, satisfied in the environment

Table 1 lists the Security Functional Requirements and the security objectives each requirement helps to address. All functional dependencies associated with the components in Table 1 have been satisfied.

The functional requirements are described in detail in the following subsections. Additionally, these requirements are derived verbatim from Part 2 of the *Common Criteria for Information Technology Security Evaluation, Version 2.1* with the exception of italicised items listed in brackets and refinements indicated by bold italics. The bracketed items include either "assignments" that are TOE specific or "selections" from the Common Criteria that the TOE enforces. Iterations are indicated with typical CC requirement naming followed by a number in parenthesis for each iteration.

5.1.1 Security Audit (FAU)

5.1.1.1 FAU_ARP.1(1) Security Alarms *for Spam Detection*

Hierarchical to: No other components.

FAU_ARP.1.1 The TSF shall take [assignment: *one of the following actions:*

A) Drop the message.

B) Deliver the original message but also send a copy it as an attachment to an alternate email address.

C) Forward the message to an alternate email address instead of the original recipient.

D) Add additional information to the message in order to facilitate future processing by subsequent queues.

E) Quarantine the message for a specified number of days.

] upon detecting ***that the email message qualifies as spam***.

Dependencies: FAU_SAA.1(1) Potential violation analysis ***for Spam Detection***.

5.1.1.2 FAU_ARP.1(2) Security Alarms *for Content Match*

Hierarchical to: No other components.

FAU_ARP.1.1 The TSF shall take [assignment: *one of the following actions:*

A) Drop the message.

B) Reroute the message to a different mail server for further processing.

C) Quarantine the message for the specified number of days.

D) Deliver the original message but also send a copy it as an attachment to an alternate email address.

F) Forward the message to an alternate email address instead of the original recipient.

E) Drop the attachment from the email.

F) Add information to the message subject to indicate a match for the given message attachment.

G) Replace matched content with alternate text.

And optionally notify an alternate recipient about the content match.

] upon detection of ***specific content in email messages or their attachments***.

Dependencies: FAU_SAA.1(2) Potential Violation Analysis ***for Content Match***.

5.1.1.3 FAU_ARP.1(3) Security Alarms *for Mail Policy Violation*

Hierarchical to: No other components.

FAU_ARP.1.1 The TSF shall take [assignment: *one of the following actions:*

A) Drop the message.

B) Reroute the message to a different mail server for further processing.

C) Quarantine the message for the specified number of days.

D) Deliver the original message but also send a copy it as an attachment to an alternate email address.

G) Forward the message to an alternate email address instead of the original recipient.

E) Add information to the message to indicate the policy match.

And optionally notify an alternate recipient about the policy match.

] upon detection of a potential security violation ***in mail policy***.

Dependencies: FAU_SAA.1(3) Potential Violation Analysis ***for Mail Policy***.

5.1.1.4 FAU_ARP.1(4) Security Alarms *for Encrypted Mail Policy Violation*

Hierarchical to: No other components.

FAU_ARP.1.1 The TSF shall take [assignment: *one of the following actions:*

A) Drop the encrypted message.

B) Drop the plain message.

C) Allow the encrypted message

D) Drop the encrypted message

] upon detection of a potential security violation ***in encrypted mail policy***.

Dependencies: FAU_SAA.1(4) Potential Violation Analysis ***for Encrypted Mail Policy***.

5.1.1.5 FAU_ARP.1(5) Security Alarms *for System Alert Notification*

Hierarchical to: No other components.

FAU_ARP.1.1 The TSF shall take [assignment: *one of the following actions:*

A) Send an email to the designated address, alerting the occurrence of the event.

B) Send a page message to the designated page address, alerting the occurrence of the event.

C) Set an SNMP trap, alerting the occurrence of the event.

] upon detection of a potential security violation ***through system alert notifications***.

Dependencies: FAU_SAA.1(5) Potential violation analysis ***for System Alert Notification***.

5.1.1.6 FAU_GEN.1 Audit Data Generation

Hierarchical to: No other components.

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

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

b) All auditable events for the [selection: *unspecified*] level of audit; and

c) [assignment: *events listed in Table 2*].

Table 2 - Auditable Events

Functional Component	Auditable Event	Additional Audit Record Contents
FAU_ARP.1(1)	Actions taken due to detection of spam	Policy that was matched, message details
FAU_ARP.1(2)	Actions taken due to content match	Policy that was matched, message details
FAU_ARP.1(3)	Actions taken due to imminent security violations in mail policy	Policy that was matched, message details
FAU_ARP.1(4)	Actions taken due to imminent security violations in encrypted mail policy	Policy that was matched, message details
FAU_ARP.1(5)	Actions taken due to system notification events	Notification event that was generated
FAU_GEN.1	Startup and Shutdown of audit	None

FAU_SAA.1(1)	Enabling and disabling of the spam queue or individual spam tools	None
FAU_SAA.1(2)	Enabling and disabling of the content filtering queue or individual rules	None
FAU_SAA.1(3)	Enabling and disabling of the mail policy queue or individual rules	None
FAU_SAA.1(4)	Enabling and disabling of the encrypted mail policy rules	None
FAU_SAA.1(5)	Enabling and disabling of the delivery of alerts	None
FAU_SAR.1	None	None
FAU_SEL.1	All modifications to the Audit configuration while audit collection function is operating	None
FAU_STG.1	None	None
FMT_MOF.1(1)	None	None
FMT_MOF.1(2)	None	None
FMT_MTD.1	None	None
FMT_SMF.1	Use of the management functions	None
FMT_SMR.1	Modification to the group of users that are part of a role	None
FPT_RVM.1(1)	None	None
FPT_TST.1	None	None

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

a) Date and time of the event, type of event, subject identity, 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, [assignment: *additional audit record contents specified in Table 2*].

Dependencies: FPT_STM.1 Reliable Time Stamps.

5.1.1.7 FAU_SAA.1(1) Potential Violation Analysis *for Spam Detection*

Hierarchical to: No other components.

FAU_SAA.1.1 The TSF shall be able to apply a set of rules in monitoring the audited events and based upon these rules indicate a potential violation of the TSP.

FAU_SAA.1.2 The TSF shall enforce the following rules for monitoring audited events:

a) Accumulation or combination of [assignment: *following events*

i) Messages explicitly identified as spam.

ii) Messages sent to specific addresses that are configured as spam traps.

iii) Message headers containing a specific value in the given field.

iv) Unknown or inconsistent source or destination addresses for the message

] known to indicate a potential security violation;

b) [assignment: *additional rules as follows*

i) *Deny any messages determined to be spam based on a comparison between a confidence value relative to the given message and a predefined threshold value.*

ii) *Permit any message that is explicitly allowed to bypass the spam filtering subsystem.*

].

Dependencies: FAU_GEN.1 Audit Data Generation.

5.1.1.8 FAU_SAA.1(2) Potential Violation Analysis *for Content Match*

Hierarchical to: No other components.

FAU_SAA.1.1 The TSF shall be able to apply a set of rules in monitoring the audited events and based upon these rules indicate a potential violation of the TSP.

FAU_SAA.1.2 The TSF shall enforce the following rules for monitoring audited events:

a) Accumulation or combination of [assignment: *following events*

i) Presence of dirty or offensive words in messages or specified attachment types.

ii) Presence of specific attachment types in the message.

] known to indicate a potential security violation;

b) [assignment: *additional rules as follows*

i) Permit any message that is explicitly allowed to bypass the content filtering subsystem.

].

Dependencies: FAU_GEN.1 Audit Data Generation.

5.1.1.9 FAU_SAA.1(3) Potential Violation Analysis *for Mail Policy*

Hierarchical to: No other components.

FAU_SAA.1.1 The TSF shall be able to apply a set of rules in monitoring the audited events and based upon these rules indicate a potential violation of the TSP.

FAU_SAA.1.2 The TSF shall enforce the following rules for monitoring audited events:

a) Accumulation or combination of [assignment: *following events*

i) *Messages sent by a specific user, group or domain.*

ii) *Messages destined to a specific user, group or domain.*

iii) *Messages containing specific text in the subject line.*

] known to indicate a potential security violation;

b) [assignment: *additional rules as follows*

i) *Permit any message that is explicitly allowed to bypass the Mail Policy filtering subsystem.*

].

Dependencies: FAU_GEN.1 Audit Data Generation.

5.1.1.10 FAU_SAA.1(4) Potential Violation Analysis *for Encrypted Mail Policy*

Hierarchical to: No other components.

FAU_SAA.1.1 The TSF shall be able to apply a set of rules in monitoring the audited events and based upon these rules indicate a potential violation of the TSP.

FAU_SAA.1.2 The TSF shall enforce the following rules for monitoring audited events:

b) Accumulation or combination of [assignment: *following events*

i) *Encrypted messages sent from a specific user, group or domain*

ii) *Encrypted messages destined to a specific user, group or domain*

iii) *Plain messages sent from a specific user, group or domain*

iv) *Plain messages destined to a specific user, group or domain*

] known to indicate a potential security violation;

b) [assignment: *additional rules as follows*

i) *Permit any encrypted or plain message that is explicitly allowed to bypass the Mail Policy filtering subsystem.*

].

Dependencies: FAU_GEN.1 Audit Data Generation.

5.1.1.11 FAU_SAA.1(5) Potential Violation Analysis *for System Alert Notification*

Hierarchical to: No other components.

FAU_SAA.1.1 The TSF shall be able to apply a set of rules in monitoring the audited events and based upon these rules indicate a potential violation of the TSP.

FAU_SAA.1.2 The TSF shall enforce the following rules for monitoring audited events:

a) Accumulation or combination of [assignment: *following events*

i) *Unexpected termination of a subsystem.*

ii) *Restarting of a subsystem after it was stopped.*

iii) *Errors encountered by a subsystem.*

iv) *Running out of disk space.*

] known to indicate a potential security violation;

b) [assignment: *additional rules as follows*

i) *The TOE shall not generate notifications for events that are not mapped to any alert mechanism.*

].

Dependencies: FAU_GEN.1 Audit Data Generation.

5.1.1.12 FAU_SAR.1 Audit Review

Hierarchical to: No other components.

FAU_SAR.1.1 The TSF shall provide [assignment: *authorised administrators*] with the capability to read [assignment: *action and incident logs, email usage and traffic patterns*] from the audit records.

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

Dependencies: FAU_GEN.1 Audit Data Generation.

5.1.1.13 FAU_SEL.1 Selective Audit

Hierarchical to: No other components.

FAU_SEL.1.1 The TSF shall be able to include or exclude auditable events from the set of audited events based on the following attributes:

a) [selection: *event type*].

b) [assignment: *log level*].

Dependencies: FAU_GEN.1 Audit Data Generation,

FMT_MTD.1 Management of TSF Data.

5.1.1.14 FAU_STG.1 Protected Audit Trail Storage

Hierarchical to: No other components.

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

FAU_STG.1.2 The TSF shall be able to [selection: *prevent*] modifications to the audit records.

Dependencies: FAU_GEN.1 Audit Data Generation.

5.1.2 Security Management (FMT)

5.1.2.1 FMT_MOF.1(1) Management of Security Functions Behaviour

Hierarchical to: No other components.

FMT_MOF.1.1 The TSF shall restrict the ability to [selection: *enable*] the functions [assignment:

a) Spam Filter

b) Content Filter

c) Mail Policy Filter

] to [assignment: *authorised administrators*].

Dependencies: FMT_SMR.1 Security Roles.

5.1.2.2 FMT_MOF.1(2) Management of Security Functions Behaviour

Hierarchical to: No other components.

FMT_MOF.1.1 The TSF shall restrict the ability to [selection: *disable*] the functions [assignment:

a) Spam Filter

b) Content Filter

c) Mail Policy Filter

] to [assignment: *authorised administrators*].

Dependencies: FMT_SMR.1 Security Roles.

5.1.2.3 FMT_MTD.1 Management of TSF Data

Hierarchical to: No other components.

FMT_MTD.1.1 The TSF shall restrict the ability to [selection**: *perform operations as specified in Table 3*** [assignment: *and no other operation*]] the [assignment: *list of TOE data as specified in Table 3*] to [assignment: *authorised administrators*].

Dependencies: FMT_SMR.1 Security Roles.

Table 3 - Management of TOE data

Functional Component	Operation	TOE Data
FAU_ARP.1(1)	Change	Action taken when spam is detected
FAU_ARP.1(2)	Change	Action taken when specific content is matched
FAU_ARP.1(3)	Change	Action taken when mail policy rules are matched

FAU_ARP.1(4)	Change	Action taken when encrypted mail policy rules are matched
FAU_ARP.1(5)	Change	Action taken when system alerts are generated
FAU_SAA.1(1)	Add, remove, modify Add, remove	Rules that identify spam Rules that allow bypass of the spam queue
FAU_SAA.1(2)	Add, remove, modify Add, remove	Rules that match specific content Rules that allow bypass of the content filtering queue
FAU_SAA.1(3)	Add, remove, modify Add, remove	Rules that mach specific mail policy rules Rules that allow bypass of the mail policy queue
FAU_SAA.1(4)	Add, remove, modify Add, remove	Rules that mach specific encrypted mail policy rules Rules that allow bypass of the encrypted mail policy queue
FAU_SAA.1(5)	Add, remove, modify	Rules that map alerts to mechanisms
FAU_SAR.1	Add, remove, modify	Group of users allowed to read audit records
FAU_SEL.1	Modify	Rights to view or change audit events
FMT_MOF.1(1)	Add, remove, modify	Roles that can interact with the TSF
FMT_MOF.1(2)	Add, remove, modify	Roles that can interact with the TSF
FMT_MTD.1	Add, remove, modify	Group of users that can interact with the TSF data
FMT_SMR.1	Add, remove, modify	Group of users that are part of a role
FPT_TST.1	Change	The condition under or time after which self-testing

		occurs.

5.1.2.4 FMT_SMF.1 Specification of Management Functions

Hierarchical to: No other components.

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

a) Add, remove and modify rules that identify messages that qualify as spam.

b) Add, remove and modify rules that identify inappropriate content in messages.

c) Add, remove and modify mail policy rules for messages that are sent to or from specified entities.

d) Add, remove and modify mail policy rules for messages containing specific content in their subject line.

e) Add, remove and modify rules that identify messages that violate the TOE's encryption policy for the given sender and receiver.

f) Add, remove and modify rules that map security-relevant events that occur on the TOE to different alert mechanisms.

g) Add and remove rules that allow specific messages to bypass any of the spam, content filtering, mail policy or encrypted mail policy queues.

h) Enable and Disable the Spam Filter, the Content Filter and the Mail Policy Filter.

i) Select the action taken when rules for spam filtering, content filtering, mail policy and encrypted mail policy are matched and when system alerts are generated.

j) Add, remove and modify the group of users that are part of a role for viewing or modifying audited events and accessing TSF data and functions.

k) Change the condition under or time after which self-testing occurs.

]

Dependencies: No Dependencies.

5.1.2.5 FMT_SMR.1 Security Roles

Hierarchical to: No other components.

FMT_SMR.1.1 The TSF shall maintain the roles [assignment:

i. admin

ii. users authorised to read or write specific TOE program areas as defined by the authorised administrator

].

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

Dependencies: FIA_UID.1 Timing of Identification.

5.1.3 Protection of the TSF (FPT)

5.1.3.1 FPT_RVM.1(1) Non-Bypassability of the TSP

Hierarchical to: No other components.

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

Dependencies: No dependencies.

5.1.3.2 FPT_TST.1 TSF Testing

Hierarchical to: No other components.

FPT_TST.1.1 The TSF shall run a suite of self tests [selection: *periodically during normal operation* [assignment: *any other time when initiated by the authorised administrator*]] to demonstrate the correct operation of the TSF.

FPT_TST.1.2 The TSF shall provide authorised users with the capability to verify the integrity of TSF data.

FPT_TST.1.3 The TSF shall provide authorised users with the capability to verify the integrity of stored TSF executable code.

Dependencies: FPT_AMT.1 Abstract Machine Testing.

5.2 TOE Security Assurance Requirements

The TOE meets the assurance requirements for EAL2. These requirements are summarised in Table 4.

Table 4 - Assurance Requirements

Assurance Class	Component ID	Component Title	Dependencies
Configuration Management	ACM_CAP.2	Configuration Items	None
Delivery and Operation	ADO_DEL.1	Delivery Procedures	None
Delivery and Operation	ADO_IGS.1	Installation, Generation, and Start-Up Procedures	AGD_ADM.1
Development	ADV_FSP.1	Informal Functional Specification	ADV_RCR.1
Development	ADV_HLD.1	Descriptive High-Level Design	ADV_FSP.1, ADV_RCR.1
Development	ADV_RCR.1	Informal Correspondence Demonstration	None
Guidance Documents	AGD_ADM.1	Administrator Guidance	ADV_FSP.1

Assurance Class	Component ID	Component Title	Dependencies
Guidance Documents	AGD_USR.1	User Guidance	ADV_FSP.1
Tests	ATE_COV.1	Evidence of Coverage	ADV_FSP.1, ATE_FUN.1
Tests	ATE_FUN.1	Functional Testing	None
Tests	ATE_IND.2	Independent Testing - Sample	ADV_FSP.1, AGD_ADM.1, AGD_USR.1, ATE_FUN.1
Vulnerability Assessment	AVA_SOF.1	Strength of TOE Security Function Evaluation	ADV_FSP.1, ADV_HLD.1
Vulnerability Assessment	AVA_VLA.1	Developer Vulnerability Analysis	ADV_FSP.1, ADV_HLD.1, AGD_ADM.1, AGD_USR.1

5.3 Strength of Function Claim of the TOE

The claimed minimum strength of function is SOF-basic.

The objectives defined in section 4 counter the threats in section 3.3 that arise from attackers with a low attack potential.

5.4 Security Requirements for the IT Environment

Table 5 - Functional Components of the IT Environment

CC Component	Name	Dependency
FIA_AFL.1	Authentication Failure Handling	FIA_UAU.1
FIA_UAU.1	Timing of Authentication	FIA_UID.1
FIA_UAU.2	User Authentication before any action	FIA_UID.1
FIA_UID.1	Timing of Identification	None
FIA_UID.2	User Identification before any action	None
FPT_AMT.1	Abstract Machine Testing	None
FPT_PHP.1	Passive detection of physical attack	FMT_MOF.1(1), FMT_MOF.1(2), Satisfied in the TOE

FPT_RVM.1(2)	Non-Bypassability of the ***TOE***	None
FPT_SEP.1	TSF Domain Separation	None
FPT_STM.1	Reliable Time Stamps	None
FTP_ITC.1(1)	Inter-TSF Trusted Channel	None
FTP_ITC.1(2)	Inter-TSF Trusted Channel	None
FTP_ITC.1(3)	Inter-TSF Trusted Channel	None
FTP_TRP.1(1)	Trusted ***Local*** Path	None
FTP_TRP.1(2)	Trusted ***Remote*** Path	None

5.4.1 Identification and Authentication (FIA)

5.4.1.1 FIA_AFL.1 Authentication Failure Handling

Hierarchical to: No other components.

FIA_AFL.1.1 The ***IT Security Environment*** shall detect when [assignment: *three*] unsuccessful authentication attempts occur related to [assignment: *user authentication*].

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met or surpassed, the ***IT Security Environment*** shall [assignment: *send a notification to a configured user about alerting about this condition, disable the account*].

Dependencies: FIA_UAU.1 Timing of Authentication.

5.4.1.2 FIA_UAU.1 Timing of Authentication

Hierarchical to: No other components.

FIA_UAU.1.1 The ***IT Security Environment*** shall allow [assignment: *no actions*] on behalf of the user to be performed before the user is authenticated.

FIA_UAU.1.2 The ***IT Security Environment*** shall require each user to be successfully authenticated before allowing any other ***TOE***-mediated actions on behalf of that user.

Dependencies: FIA_UID.1 Timing of Identification.

5.4.1.3 FIA_UAU.2 User Authentication Before any Action

Hierarchical to: FIA_UAU.1 Timing of Authentication.

FIA_UAU.2.1 The ***IT Security Environment*** shall require each user to be successfully authenticated before allowing any other ***TOE***-mediated actions on behalf of that user.

Dependencies: FIA_UID.1 Timing of Identification.

5.4.1.4 FIA_UID.1 Timing of Identification

Hierarchical to: No other components.

FIA_UID.1.1 The ***IT Security Environment*** shall allow [assignment: *no actions*] on behalf of the user to be performed before the user is identified.

FIA_UID.1.2 The ***IT Security Environment*** shall require each user to be successfully identified before allowing any other ***TOE***-mediated actions on behalf of that user.

Dependencies: No dependencies.

5.4.1.5 FIA_UID.2 User Identification Before any Action

Hierarchical to: FIA_UID.1 Timing of Identification.

FIA_UID.2.1 The ***IT Security Environment*** shall require each user to identify itself before allowing any other ***TOE***-mediated actions on behalf of that user.

Dependencies: No dependencies.

5.4.2 Protection of the TSF (FPT)

5.4.2.1 FPT_AMT.1 Abstract Machine Testing

Hierarchical to: No other components.

FPT_AMT.1.1 The ***IT Security Environment*** shall run a suite of tests [selection: *during initial start-up*] to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the ***TOE***.

Dependencies: No dependencies.

5.4.2.2 FPT_PHP.1 Passive Detection of Physical Attack

Hierarchical to: No other components.

FPT_PHP.1.1 The ***IT Security Environment*** shall provide unambiguous detection of physical tampering that might compromise the ***TOE***.

FPT_PHP.1.2 The ***IT Security Environment*** shall provide the capability to determine whether physical tampering with the ***TOE's*** devices or ***TOE's*** elements has occurred.

Dependencies: FMT_MOF.1(1) Management of Security Functions Behaviour.

FMT_MOF.1(2) Management of Security Functions Behaviour.

5.4.2.3 FPT_RVM.1(2) Non-Bypassability of the *TOE*

Hierarchical to: No other components.

FPT_RVM.1.1 The ***IT Security Environment*** shall ensure that the ***TOE is never bypassed.***

Dependencies: No dependencies.

5.4.2.4 FPT_SEP.1 TSF Domain Separation

Hierarchical to: No other components.

FPT_SEP.1 The ***IT Security Environment*** shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

FPT_SEP.1.2 The ***IT Security Environment*** shall enforce separation between the security domains of subjects in the ***TOE***.

Dependencies: No dependencies.

5.4.2.5 FPT_STM.1 Reliable Time Stamps

Hierarchical to: No other components.

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

Dependencies: No dependencies.

5.4.3 Trusted Path/Channels (FTP)

5.4.3.1 FTP_ITC.1(1) Inter-TSF Trusted Channel

Hierarchical to: No other components.

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

FTP_ITC.1.2 The ***IT Security Environment*** shall permit [selection: ***the TOE***] to initiate communication via the trusted channel.

FTP_ITC.1.3 The ***IT Security Environment*** shall initiate communication via the trusted channel for [assignment:

a) Querying information from DNS.

b) Querying/storing information from/to the internal database.

c) Fetching mail from the mail servers due to a request by a mail client.

].

Dependencies: No dependencies.

5.4.3.2 FTP_ITC.1(2) Inter-TSF Trusted Channel

Hierarchical to: No other components.

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

FTP_ITC.1.2 The ***IT Security Environment*** shall permit [selection: ***the remote mail client***] to initiate communication via the trusted channel.

FTP_ITC.1.3 The ***IT Security Environment*** shall initiate communication via the trusted channel for [assignment:

a) Fetching mail from the mail servers protected by the TOE.

].

Dependencies: No dependencies.

5.4.3.3 FTP_ITC.1(3) Inter-TSF Trusted Channel

Hierarchical to: No other components.

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

FTP_ITC.1.2 The ***IT Security Environment*** shall permit [selection: ***remote users***] to initiate communication via the trusted channel.

FTP_ITC.1.3 The ***IT Security Environment*** shall initiate communication via the trusted channel for [assignment:

a) Accessing the web-based configuration interface to the TOE.

b) Accessing the Command Line Interface to the TOE.

].

Dependencies: No dependencies.

5.4.4 FTP_TRP.1(1) Trusted *Local* Path

Hierarchical to: No other components.

FTP_TRP.1.1 The ***IT Security Environment*** shall provide a communication path between itself and [selection: *local*] users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from modification or disclosure.

FTP_TRP.1.2 The ***IT Security Environment*** shall permit [selection: *local users*] to initiate communication via the trusted path.

FTP_TRP.1.3 The ***IT Security Environment*** shall require the use of the trusted path for [selection: *initial user authentication,* [assignment: *no other services*]].

Dependencies: No dependencies.

5.4.5 FTP_TRP.1(1) Trusted *Remote* Path

Hierarchical to: No other components.

FTP_TRP.1.1 The ***IT Security Environment*** shall provide a communication path between itself and [selection: *remote*] users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from modification or disclosure.

FTP_TRP.1.2 The ***IT Security Environment*** shall permit [selection: *remote users*] to initiate communication via the trusted path.

FTP_TRP.1.3 The ***IT Security Environment*** shall require the use of the trusted path for [selection: *initial user authentication,* [assignment: *no other services*]].

Dependencies: No dependencies.

CHAPTER 6

6. TOE Summary Specification

6.1 TOE Security Functions

The security functions implemented by the TOE are:

TSF_1 SMTP PROXY

The SMTP Proxy is responsible for listening for incoming email messages and beginning the subsequent queue processing. The SMTP inbound service processes all messages coming into the TOE, whether originating inside or outside the local network. It consists of two separate services SMTPI (for normal inbound messages) and SMPTS (for secure incoming messages). Connection requests that do not conform to SMPTP specifications are dropped.

The SMTPI service listens on port 25 for all incoming messages while the secure SMTPI proxy (SMTPS) processes messages coming into the TOE via the secure port 465. While email may be transferred securely over port 25, the SMTPIS Service listens for email exclusively on port 465.

Messages received by the SMTP service are written to the TOE's internal database, for further processing by other queues.

TSF_2 SPAM QUEUE

The Spam Queue uses a variety of tools to inspect messages for characteristics of spam. When a message is found to be spam-like, an administrator-defined action such as drop, quarantine or rename is performed on it. Tools used by the TOE to detect spam are given below. The TOE may be configured to signal the message as spam based on a match from any one tool or by from a combination of tools, using a weighted representation of results from each of the configured tools.

A) User Reported Spam: Mail identified by the user as spam is forwarded to the static address globaluserreports@spamcollector.ciphertrust.com. The TOE can then configure Mail Monitoring rules to identify future messages sent from the same address as spam, and take specific action based on it.

B) Enterprise Spam Traps: These addresses are not assigned to any users in the domain but rather are used as "honey-pots" for spammers in websites and forums such as newsgroups, where the likelihood of these addresses being captured by spammers is high. The TOE uses the pre-configured address enterprise@spamcollector.ciphertrust.net assigned for collecting such messages and acts on these messages using relevant Mail Monitoring rules for spam.

C) Header Analysis - DNS Lookup of sender domains: The "From" address in the message is parsed to obtain a domain name over which a DNS query is performed. The rule is considered to match if no valid MX-record is received for the DNS query.

D) Header Analysis - Identical EHLO and sender domain name check: The domain name that is parsed as part of EHLO command in the SMTP handshake is compared with the domain name in the "From" address. The rule is considered to match if the two values are different.

E) Header Analysis - Check EHLO domain name with RDNS lookup of IP the address: A reverse DNS lookup for the IP address of the sending host is performed. If a valid hostname is returned, this is parsed to get the domain name. This domain name and the domain name parsed as part of the EHLO command in the SMTP handshake are compared. The rule is considered to match if the two values are different.

F) Header Analysis - Identical "To" and "From" Addresses: The "To" and "From" fields are parsed from the header. If there is a single "To" recipient, then this is checked with the "From" address. The rule is considered to match if the two values are identical.

G) Header Analysis - Missing or blank both "To" and "CC" fields: The "To" and "CC" (if exists) are extracted from the headers. The rule is considered to match if both of the above values are missing or blank.

H) User-Defined Header Analysis Rules: These rules can check for the presence or absence of specific headers in the message or if the value of a specified field in the header matches a particular value.

I) Overall Spam Confidence Level: These rules allow the authorised administrator to associate a cumulative spam confidence level for the message from each of the configured spam tools and take actions if some threshold is exceeded.

The actions that can be taken when spam is detected include dropping the message, copying the message to an alternate email address, forwarding the message to an alternate address and quarantining the message for the specified number of days. The TOE can also prepend the message subject with specified text or add a new RFC822 header in order to indicate that the message qualifies as spam.

The TOE also allows the creation of spam whitelist rules - or rules that enable specified message types to bypass the spam detection queue. The authorized administrator can specify individual email addresses, domain names, or IP addresses, and for messages originating or destined to them, indicate which spam policies may be bypassed.

TSF_3 CONTENT FILTERING QUEUE

The Content Filtering queue scans the message contents for specific text or attachment types, which are considered malicious or inappropriate for circulation by the TOE. The content filtering queue operates over Attachment Filtering and Content Filtering Policies, each of which can be selectively enabled or disabled.

The TOE administrators can create groups of files and file extensions based on different criteria and then choose how the TOE should respond when message attachments arrive at specified file groupings for specific users or groups.

The TOE allows the administrators to create policies based on keywords or phrases in email or their attachments. Content filtering policies are defined in three steps - defining dictionaries that identify words or phrases that are disallowed, creating rules based on detection of multiple words present in the dictionary and identifying the users or groups to which these policies are applied.

Matched content from the attachment or content filter can be replaced with content that complies with the TOE information dissemination policy. Other actions possible include dropping the

attachment or the entire message, re-routing the message to a different mail server and quarantining the message for the specified number of days. Messages may also be forwarded or copied to alternate addresses.

The TOE also allows the creation of content filtering whitelist rules - or rules that enable specified message types to bypass the content filtering queue. The authorized administrator can specify individual email addresses, domain names, or IP addresses, and for messages originating or destined to them, indicate which content filtering policies may be bypassed.

The TOE can also be configured to send out notifications upon receiving mail matching any of the content filtering rules. The notification messages may be customized to indicate which policy was matched and the corresponding action that was taken by the TOE.

TSF_4 MAIL POLICY QUEUE

This queue allows the TOE to specify Mail Monitoring rules, which allow specific action to be taken on a message based on its sender, recipient or subject line content.

When the TOE encounters a message matching the criteria specified, it can automatically perform one of nine possible actions. These include forwarding the message to an alternate address in addition or without to the original recipient and copying the message as an attachment to an alternate address. The TOE may also re-route the message to different server, quarantine the message or drop the message altogether. Messages that are matched by the mail policy queue can have their subject lines re-written to indicate that fact.

The Mail Policy queue also enables the TOE to take action based on the encryption properties of the mail. Encrypted mail originating from or destined to specific addresses may have requirements of confidentiality or transparency. Actions can be taken based on whether the incoming mail is plain or encrypted and include dropping the message or quarantining it for a specified number of days.

The TOE can be configured to send out notifications upon receiving mail matching any of the mail monitoring rules. The notification messages may be customized to indicate which policy was matched and the corresponding action that was taken by the TOE.

The TOE also allows the creation of mail policy whitelist rules - or rules that enable specified message types to bypass the mail policy queue. The authorized administrator can specify individual email addresses, domain names, or IP addresses, and for messages originating or destined to them, indicate which mail monitoring rule may be bypassed.

TSF_5 SMTP_OUT

SMTPO Service or the Outbound Service is responsible for delivering messages out of the TOE. The TOE’s SMTPO Service wakes up at periodic intervals to see which messages have been processed by all the other queues. The administrator may view the contents of the outbound queue - that is, view the messages ready for delivery, but not yet delivered, and re-prioritise the delivery of individual or all messages addressed to a specific domain, or delete them altogether. Delivery can fail if there is some data error, if the host is unreachable, if the receiver domain is invalid or same as the TOE's name or if either the sender or recipient is refused. The SMTO service delivers all messages out of the TOE, irrespective of whether their destination is inside or outside the network.

TSF_6 GUI_MANAGER

The GUI Manager provides a web-based browser interface for the administrators to identify and authorise themselves to the TOE and to set and configure the various queue processes. Users may access the GUI Manger through a web browser by connecting to the IronMail appliance's configured address using the secure HTTP protocol.

The TOE administrator may create user accounts for additional personnel who are granted permission to perform specific duties in administering the TOE. The administrator can select which program areas the users are allowed to access, and whether their access is “read only” or “read/write.” There is one “super user” account for the TOE administrator. This “super user” account name is “admin.” Only the admin user account has access to this User Accounts window. Actual authentication of the users is done by the TOE O/S. The TOE can be configured to automatically log out an administrator after a period of inactivity.

To preserve TOE program and file system integrity the TOE can be configured to periodically check every executable in the system to make sure that it has not been modified, and perform file system integrity checks to ensure that no files have been deleted or newly introduced into the system. New updates or patches may be automatically downloaded to maintain integrity of the TOE code base. All of the above tests may also be initiated manually by the TOE administrator through the relevant interface in the GUI manager.

The TOE provides a unified policy manager available through the GUI Manager. This allows administrators to define, monitor and enforce email policy across all email servers within the organization. Every time a message is received by TOE, an internal lookup is performed to determine which policies are to be enforced on it. The policies in effect as of the moment the message arrived at the TOE are enforced. Changed policies only affect new messages entering the TOE after the policy was updated. Rules are directional - when the SMTP queue encounters a rule, it checks if the rule direction agrees with that of the message. If the directions don't match, the rule is bypassed.

Queues in the TOE may be selectively enabled or disabled. If enabled, the GUI manager allows the administrator to define the behavior of the various queuing subsystems as outlined below.

A) Mail Policy Queue:

The Mail Monitoring Rule Management table, empty until rules are created, displays information about each Mail Monitoring rule. While individual rules are created on this page, they are not turned into policies until applied to users or groups. Rules may be applied to inbound messages, outbound messages or both. Rules can be configured to take action based on messages originating from or destined to a particular user, group or domain and to generate notifications when mail monitoring rules are matched successfully. Rules can be deleted by clicking on the delete check box for the appropriate rule and clicking submit.

The Mail Policy queue also enforces policies defined in the Encrypted Message Filtering Table while processing inbound messages. The Encrypted Message Filtering Rule Management table, empty until rules have been created, displays information about the individual rules relating to encrypted messages. The Rule ID is a hyperlink opening a secondary browser window in which the rule may be edited. The rules can be configured to allow the TOE to take some action based

on the encryption properties of the messages originating from or destined to a particular user or group. Rules that are no longer required may be deleted if required.

B) Spam Queue:

Enterprise Spam Traps allows the authorized administrator to enter list of addresses (spam traps) that if used would readily identify spam. User-Defined Header Analysis Rules allows the authorized administrator to define a rule for the header field, a condition and the value to be checked. Rules may be enabled or disabled. Multiple thresholds and action can be defined for User-Defined Header Analysis Rules. Finally, Overall Spam Confidence Level (OSCL) gives the user an option to assign each message a confidence percentage that contributes toward its classification as spam.

Messages passing through the spam queue can also be tagged with an additional header that qualifies it as having been processed by this queue. The header name can be fifteen characters in length and it cannot contain a ":" character.

C) Content Filtering Queue:

Filtering policy rules may be defined based on the message attachment and message content. For each filtering policy defined, the appropriate values such as the attachment prefixes, the specific content to be matched against and the addresses (that the quarantined or delayed delivery messages are to be forwarded to) can be specified. Multiple forwarding addresses may also be specified. Matching values can also take an associated replacement text or attachment that is used as a substitute. Standard footer values may also be defined for messages that have been modified or processed by the content filter. The TOE allows the authorized administrators to create dictionaries populated with words, phrases, weighted word lists and text patterns for use in content filtering.

D) Quarantine Queue:

This is not strictly a message-processing queue, but rather a logical “holding area” where other queue services may send message if certain conditions are met. The TOE allows administrators to create multiple quarantine queues to facilitate the management of its email policies. The administrator may view the contents of the quarantine queues at any time. The administrator may delete, re-prioritize, change the scheduled delivery time, or re-direct to an alternate address any message in any of the quarantine queues.

E) Outbound Queue:

This is the TOE's SMTPO Service, responsible for delivering messages out of the TOE. The administrator may view the contents of the Outbound Queue—that is, view the messages ready for delivery, but not yet delivered—and re-prioritize the delivery of either individual messages or all messages addressed to a specific domain, or delete them.

Many of the TOE's policies provide an option to notify users if a TOE policy performs an action on an email. The Custom Mail Notification page is where administrators may personalize the

notification email that is delivered to the user. The TOE provides a number of templates corresponding to the different queuing subsystems to support user notification. Selecting a template for a policy populates text fields in the lower half of the page with sample text. The sample text may be edited and personalized as required.

The Queue Whitelist program area allows administrators to finely differentiate which users or domains may bypass any or all of the TOE's policy and queue services. The administrator can enter individual email addresses, domain names, or IP addresses, and for each, indicate if the rule for those entities applies to inbound or outbound messages, and which TOE policies those messages may bypass. Only class A, B and C subnets are allowed while specifying the IP addresses. The anti-spam queue allows bypass of specific spam tools used by the spam filter and the policy manager queue allows bypass of the mail policy, encrypted mail policy and content filtering queues. The whitelist rules may be deleted but not be edited. To change a whitelist rule, it must first be deleted from the table and then recreated as desired.

The GUI manager also provides the user an interface for configuring and viewing logged events and monitoring general TOE health.

TSF_7 COMMAND LINE INTERFACE

The TOE allows administrators to access much of the functionality found in the graphical user interface (GUI) from a command line. If a keyboard and monitor are attached to the TOE and TOE is currently running, the monitor will display a logon prompt. Once the administrator enters a valid username and password various TOE operations may be accessed by simple commands, where these commands are composed of a command word followed by one or more parameters. Unlike the graphical user interface where the user is automatically logged out after a specified length of time, administrators remain logged on until they manually log off.

The information displayed in the command line interface adopts the same restrictions and parameters that are set in the GUI. For example, if logging for the SMTPO Service is set to “1” in the GUI, only that level of information is displayed in the CLI interface.

The TOE's CLI is a limited shell, and administrators may not gain root access on it.

TSF_8 ALERT MANAGER

The Alert Manager delivers alerts based on policy configurations. The TOE constantly monitors its core subsystems, as well as its ability to communicate with internal mail servers. If any part of the TOE's functionality fails to perform as designed, the TOE generates an alert.

There are a finite number of anomalies that the TOE can report on. Each anomaly is “hard coded” with one of seven “alert levels” indicating the degree of criticality of the problem. The TOE administrators can create an alert mechanism (email, pager, SNMP trap) for any or all of the “alert levels”.

Administrators may create any logical grouping of services that serves their needs. Individual subsystems may be moved from one grouping or “class” to another or deleted altogether. The purpose of creating classes of subsystems is to be “granular” in terms of which alert notifications are received. If an alert mechanism for any grouping of events is not created, no alerts are generated for that set of events.

TSF_9 WATCHDOG DAEMON

The Watchdog Daemon is a process that runs continuously checking for the heartbeat of all other processes thus ensuring that all processes are up and running.

The daemon can configured to recognize specific problems in the TOE such as subsystems stopping unexpectedly, or restarting after they were stopped and also events such as the TOE running out of disk-space.

The Watchdog Daemon communicates with the Alert Manager (TSF_8) and internally with the Admin Server, the database and the operating system. The Watchdog Daemon monitors each of the Queue processes (TSF_2 – TSF_5) directly, to ensure they are running as expected. The Watchdog Daemon can automatically start and stop processes - if a queue process is down, the Watchdog Daemon communicates with the Admin Server to start the queue and the Alert Manager (TSF_8) to send an alert notifying the administrator of the problem.

TSF_10 LOGGING ENGINE

The Logging Engine performs all logging and auditing of the Administrator activities. Log4j (an open-sourced logging framework under Jakarta Apache project) is extended so as to support auditing of TOE events. The logging framework allows the administrator to control the output logs and configure them externally through customisable log levels and output mechanisms.

The TOE can generate daily reports in HTML, showing detailed information about the incoming and outgoing messages processed by the TOE each day. Detailed and summary policy compliance reports may also be generated. Additionally, the reports may be archived as “CSV”(comma separated values) files, for analysis in third-party applications.

While Reports provide a “high level” overview of the TOE’s message-processing activity, Logs show “low level”, or detailed information about message processing at the level of the individual message. The TOE offers six log levels numbered 1 through 6 with increasing amount of logging present at each level. Depending on the logging level configured for each TOE subsystem, the logs can report on the specific steps taken while processing individual messages and whether or not the TOE even received the message. Summary logs can be exported in "real time" as SysLogs.

The TOE generates detailed logs of events generated by each one of its subsystems namely the anti-spam, content filtering and mail monitoring queues, and the internal rip, join and quarantine queues. Important fields from the message and the rules that they match are saved in these log records. Activities of users who access the Web Administration interface is also recorded. Thus any new rule added, any queue disabled and any selection made by the authorized users of the TOE can be audited in this way. The TOE log reports also records every connection made on the CLI interface, every command invoked over it, every alert generated and every system test performed by it.

All messages that the TOE processes (with the exception of messages the TOE drops because of an email policy’s action) may be saved to disk and archived.

6.2 TOE Security Function Rationale

Table 6 demonstrates the correspondence between the security functional requirements identified in Sections 5.1 and the TOE security functions identified in Section 6.1.

Table 6 - Mappings Between TOE Security Functional Requirements and TOE Security Functions

	TSF_1	TSF_2	TSF_3	TSF_4	TSF_5	TSF_6	TSF_7	TSF_8	TSF_9	TSF_10
FAU_ARP.1 (1)		X								
FAU_ARP.1 (2)			X							
FAU_ARP.1 (3)				X						
FAU_ARP.1 (4)				X						
FAU_ARP.1 (5)								X	X	
FAU_GEN.1	X	X	X	X	X	X	X	X	X	X
FAU_SAA.1(1)		X								
FAU_SAA.1(2)			X							
FAU_SAA.1(3)				X						
FAU_SAA.1(4)				X						
FAU_SAA.1(5)								X	X	
FAU_SAR.1										X
FAU_SEL.1										X
FAU_STG.1								X	X	X
FMT_MOF.1(1)						X	X			
FMT_MOF.1(2)						X	X			
FMT_MTD.1						X	X			
FMT_SMF.1						X	X			
FMT_SMR.1						X	X			
FPT_RVM.1(1)	X				X					
FPT_TST.1								X	X	

The SMTP queue (TSF_1) ensures that every email message passes through the TOE before being forwarded to its destination by the SMTP outbound component (TSF_5). The queuing architecture of the TOE ensures that no mail bypasses any filtering queue unless the TOE

administrator explicitly configures it as such. The ability of the TOE to monitor and process every incoming mail help satisfy FPT_RVM.1(1).

TSF_2, TSF_3, TSF_4 and TSF_8 automatically detect and respond to conditions that are met when the incoming message contains specific data -- the spam queue detects spam, the content filtering queue detects inappropriate content in the messages, the mail policy queue detects policy violations and the alert manager detects the failure of TOE components. In each case, the defined set of responses correspond with the appropriate requirement. Thus, the spam queue satisfies (FAU_SAA.1(1), FAU_ARP.1(1)), the content filtering queue satisfies (FAU_SAA.1(2), FAU_ARP.1(2)), the mail policy queue satisfies (FAU_SAA.1(3), FAU_ARP.1(3)) and (FAU_SAA.1(4), FAU_ARP.1(4)), and the alert manager satisfies (FAU_SAA.1(5), FAU_ARP.1(5)).

The TOE provides integrated policy definition capability through the GUI Manager (TSF_6) and the Command Line Interface (TSF_7). These components allow the authorised administrator to enable, disable and configure the different queues, thus satisfying FMT_MOF.1(1) and FMT_MOF.1(2), manage the different authorised roles and their privileges for configuring TOE subsystems, thus satisfying FMT_SMR.1, and configure the TSF data shown in Table 7 below. The ability to configure the above parameters helps satisfy requirements for FMT_SMF.1 and FMT_MTD.1.

Table 7 - Management of TOE data

Operation	TOE Data	Satisfies FMT_MTD.1 requirements for the following
Modify	Log level	FAU_SEL.1
Add, Delete	Whitelist rules	FAU_SAA.1(1), FAU_SAA.1(2), FAU_SAA.1(3), FAU_SAA.1(4), FAU_SAA.1(5)
Add, Remove, Modify	Mail monitoring Rules	FAU_SAA.1(3), FAU_ARP.1(3)
Add, Remove, Modify	Encrypted Mail monitoring Rules	FAU_SAA.1(4), FAU_ARP.1(4)
add, remove, modify	Spam Rules,	FAU_SAA.1(1), FAU_ARP.1(1)
Modify	Spam OSC threshold, Enterprise Spam Trap addresses	FAU_SAA.1(1)
Add, remove, modify	Content Filter Rules,	FAU_SAA.1(2), FAU_ARP.1(2)
Add, remove, modify	Actions to be taken when the Alert Manger detects failure of some component	FAU_SAA.1(5), FAU_ARP.1(5)
Add, remove,	List of users assigned read or	FAU_SAR.1, FMT_MOF.1(1),

Operation	TOE Data	Satisfies FMT_MTD.1 requirements for the following
modify	write privileges for configuring various TOE functionality	FMT_MOF.1(2), FMT_MTD.1, FMT_SMR.1
Change	Time interval for File and Program integrity testing	FPT_TST.1

Logs are generated by every component of the system when ever some security relevant action is performed, which corresponds to the FAU_GEN.1 requirement. The Logging engine (TSF_10) provides the ability to selectively view and generate events including all auditable events listed in Table 2, which satisfies FAU_SAR.1 and FAU_SEL.1 requirements. The log records are stored in the TOE's local hard disk and only the authorised administrator is able to view or delete these records. Out-of-space events are detected by the Watchdog daemon (TSF_9) and are communicated by the Alert Manager (TSF_8) to the TOE administrator, satisfying FAU_STG.1 requirements.

The Watchdog daemon (TSF_9) monitors the health of the TOE. If a Queue process is down, the Watchdog Daemon communicates with the internal Admin Server, which then restarts the appropriate queue. The TOE periodically performs program and file system integrity checks to ensure that the TOE code and data have not been compromised. These features help satisfy requirements for FPT_TST.1.

6.3 Assurance Measures

The TOE stresses assurance through vendor actions that are within the bounds of current best commercial practice. The TOE provides, primarily via review of vendor-supplied evidence, independent confirmation that these actions have been competently performed.

The general level of assurance for the TOE is:

A) Consistent with current best commercial practice for IT development and provides a product that is competitive against non-evaluated products with respect to functionality, performance, cost, and time-to-market.

B) The TOE assurance also meets current constraints on widespread acceptance, by expressing its claims against EAL2 from part 3 of the Common Criteria.

Table 8 demonstrates the correspondence between the security assurance requirements listed in Sections 5.2 to the developer evidence.

Table 8 - Assurance Correspondence

Component ID	Developer Evidence
ACM_CAP.2	CipherTrust Configuration Management Overall Structure 5/12/2004
ADO_DEL.1	CipherTrust IronMail – TOE Delivery Procedures 4/27/2004
ADO_IGS.1	CipherTrust IronMail Setup Guide 4/19/2004

Component ID	Developer Evidence
ADV_FSP.1	CTV_FSP Functional Specification For NIAP CC Evaluation, August 2003
ADV_HLD.1	CT_HLD High-Level Design Document
ADV_RCR.1	ADV_RCR.1 Informal Correspondence Demonstration For NIAP CC Evaluation, August 2003
AGD_ADM.1	IronMail CMC User's Guide Version 1.5.0
AGD_USR.1	IronMail Version 4.0.0 User Manual
ATE_COV.1	Alert Manager Test Cases V2
ATE_FUN.1	IronMail Product Requirements Alert Manager for Version 4.0.0 Document
ATE_IND.2	CipherTrust IronMail Secure Email Gateway Software Version 4.0.0 Functional Test Report March 7, 2006 F2-0306-002
AVA_SOF.1	Developer Strength of Function Analysis Document (AVA_SOF.1) January 18, 2006
AVA_VLA.1	Vulnerability Assessment Document IronMail v4.0 January 17, 2006

CHAPTER 7

7. Protection Profile Claims

This chapter provides detailed information in reference to the Protection Profile conformance identification that appears in Chapter 1, Section 1.4 Protection Profile Conformance.

7.1 Protection Profile Reference

This Security Target does not claim conformance to any registered Protection Profile.

7.2 Protection Profile Refinements

This Security Target does not claim conformance to any registered Protection Profile.

7.3 Protection Profile Additions

This Security Target does not claim conformance to any registered Protection Profile.

7.4 Protection Profile Rationale

This Security Target does not claim conformance to any registered Protection Profile.

CHAPTER 8

8. Rationale

Tables 9 and 10 demonstrate the correspondence between the security objectives listed in Sections 4.1 - 4.2 to the assumptions, threats and policies identified in Sections 3.2, 3.3 and 3.4.

8.1 Security Objectives Rationale

Table 9 - Correspondence between Assumptions, Threats and Policies to Objectives

Policies/Threats/ Assumptions	Objectives
A.DB_INTEGRITY	O.E.TRUSTED_INFO
A.DNS	O.E.TRUSTED_INFO
A.NO_EVIL_ADMIN	O.E.TRUSTED_ENV
A.PHYSICAL_SECURITY	O.E.TRUSTED_ENV
T.BYPASS	O.REF_MEDIATION, O.E.NO_BYPASS
T.COMP_FAILURE	O.NOTIFICATION
T.CONTENT	O.CONTENT_FILTER
T.NEW_EXPLOITS	O.CONFIGURABILITY
T.NO_AUDIT	O.LOG, O.E.TS_INTEGRITY
T.NO_REGULATE	O.MAIL_POLICY
T.OPAQUE	O.MAIL_POLICY
T.RESOURCE_CONSUME	O.SPAM_FILTER
T.UNTRUSTED_CODE	O.TOE_INTEGRITY
T.E.AUTH_CAPTURE	O.E.DOMAIN_SEP
T.E.BRUTE_FORCE	O.E.BOUNDED_AUTH
T.E.EXT_CAPTURE	O.E.EXT_CHAN
T.E.IA	O.E.AUTHENTICATION
T.E.INT_CAPTURE	O.E.INT_CHAN
T.E.MASQUERADE	O.E.TRUSTED_PATH

Table 10 - Correspondence between Objectives and Assumptions, Threats and Policies

Objectives	Policies/Threats/ Assumptions
O.CONFIGURABILITY	T.NEW_EXPLOITS

Objectives	Policies/Threats/ Assumptions
O.CONTENT_FILTER	T.CONTENT
O.LOG	T.NO_AUDIT
O.MAIL_POLICY	T.OPAQUE, T.NO_REGULATE
O.NOTIFICATION	T.COMP_FAILURE
O.REF_MEDIATION	T.BYPASS
O.SPAM_FILTER	T.RESOURCE_CONSUME
O.TOE_INTEGRITY	T.UNTRUSTED_CODE
O.E.AUTHENTICATION	T.E.IA
O.E.BOUNDED_AUTH	T.E.BRUTE_FORCE
O.E.DOMAIN_SEP	T.E.AUTH_CAPTURE
O.E.EXT_CHAN	T.E.EXT_CAPTURE
O.E.INT_CHAN	T.E.INT_CAPTURE
O.E.NO_BYPASS	T.BYPASS
O.E.TRUSTED_ENV	A.NO_EVIL_ADMIN, A.PHYSICAL_SECURITY
O.E.TRUSTED_INFO	A.DB_INTEGRITY, A.DNS
O.E.TRUSTED_PATH	T.E.MASQUERADE
O.E.TS_INTEGRITY	T.NO_AUDIT

8.1.1 Rationale for TOE Security Objectives

8.1.1.1 T.BYPASS

T.BYPASS is the threat of a malicious entity bypassing one or more of the TOE's security functions in order send malicious data to the internal mail servers without the TOE detecting it. O.E.NO_BYPASS ensures in the IT environment that data cannot be sent into the internal network through any route that bypasses the TOE altogether. O.REF_MEDIATION ensures that every inbound our outbound mail that reaches the TOE, unless specifically allowed by the TOE administrator, must pass through each of its configured filters before being forwarded onto their respective destinations. The combination of the above objectives successfully counters T.BYPASS.

8.1.1.2 T.COMP_FAILURE

This threat covers the event of unexpected termination of one or more of the TOE's security functions, which may allow a threat agent to send inappropriate information through the TOE. By achieving O.NOTIFICATION, alerts are generated whenever any TOE subsystem fails. The

alerts can then trigger mechanisms that prevent such mail from passing through the TOE while the subsystem is inactive.

8.1.1.3 T.CONTENT

This is the threat of dirty, offensive, proprietary or otherwise inappropriate content being sent through the TOE. By implementing O.CONTENT_FILTER the TOE can take specific action on such messages, thus directly countering the above threat.

8.1.1.4 T.NEW_EXPLOITS

T.NEW_EXPLOITS is the threat where a malicious sender may modify the message content suitably or use variants in the sender or recipient information in order to defeat the protection services offered by the TOE. By implementing O.CONFIGURABILITY, the TOE administrator can ensure that an up-to-date knowledge base of known malicious entities or variants in messages that constitute policy violations is installed on the TOE.

8.1.1.5 T.NO_AUDIT

T.NO_AUDIT is the threat of the TOE administrator not being able to detect compromise of the TOE due to lack of any accounting information. The above threat is countered by implementing O.LOG, which ensures that the TOE maintains a log of all the security-relevant operations performed on the TOE.

In order to be able to reliably correlate events there must be some temporal attribute or a timestamp associated with every audit record. The TOE must additionally implement O.E.TS_INTEGRITY to ensure that the timestamps used in the audit records are reliable.

8.1.1.6 T.NO_REGULATE

This is the threat of an entity attempting to send content that the TOE may not want to receive, either because of its origin, destination or subject content. This threat can be countered by implementing O.MAIL_POLICY, which allows the TOE to configure specific actions to be taken on incoming mail based on its sender, its recipient or its subject content.

8.1.1.7 T.OPAQUE

Inappropriate content such as proprietary information for an organization may be sent as encrypted data thus escaping detection by the content filter. This threat is countered by enforcing a policy that allows only specific users, groups or domains to send and receive encrypted information, which is covered by O.MAIL_POLICY.

8.1.1.8 T.RESOURCE_CONSUME

Spam is the primary cause for consumption of resources such as memory, bandwidth, processor time and data storage on the TOE. The TOE can counter T.RESOURCE_CONSUME by being able to define characteristics for identifying spam and take appropriate action when such characteristics are recognized, which is O.SPAM_FILTER.

8.1.1.9 T.UNTRUSTED_CODE

T.UNTRUSTED_CODE is the threat of untrusted code being downloaded to the TOE by some malicious entity, which causes abnormal processes that violate the integrity and availability of system assets to be executed on it. This threat can be countered by ensuring that the integrity of the TOE trusted code base is always maintained or O.TOE_INTEGRITY.

8.1.2 Rationale for IT Environment Security Objectives

8.1.2.1 A.DB_INTEGRITY

O.E.TRUSTED_INFO ensures that the integrity of the information received by the TOE from trusted external subsystems is never compromised. This addresses A.DB_INTEGRITY, or the assumption that the integrity of data maintained by the TOE's MySQL database is always maintained.

8.1.2.2 A.DNS

O.E.TRUSTED_INFO ensures that the integrity of the information received by the TOE from trusted external subsystems is never compromised. This addresses A.DNS, or the assumption that information received through DNS is reliable.

8.1.2.3 A.NO_EVIL_ADMIN

If O.E.TRUSTED_ENV is achieved then the TOE cannot be compromised by inside entities. This includes compromise by the administrators of the TOE who are then assumed to be are non-hostile and appropriately trained to use, configure and maintain the TOE, which is A.NO_EVIL_ADMIN.

8.1.2.4 A.PHYSICAL_SECURITY

If O.E.TRUSTED_ENV is achieved then the TOE cannot be physically compromised malicious entities. This includes the assumption that the TOE resides in a physically controlled access facility that cannot be physically compromised by unauthorized entities including malicious insiders, which is A.PHYSICAL_SECURITY.

8.1.2.5 T.E.AUTH_CAPTURE

This is the threat of a malicious entity gaining unauthorized access to the TOE by executing a process that captures the authentication data of a valid user. By implementing O.E.DOMAIN_SEP, the TOE can maintain its own domain for execution and ensure that it cannot be interfered with or tampered by any untrusted subject thus preventing the above attack.

8.1.2.6 T.E.BRUTE_FORCE

A threat agent may attempt brute force attacks against the TOE authentication mechanism by repeatedly trying to guess authentication data for valid uses of the TOE. The TOE can counter T.E.BRUTE_FORCE by bounding the number of failed authentication attempts and take appropriate actions when this threshold is met, which is O.E.BOUNDED_AUTH.

8.1.2.7 T.E.EXT_CAPTURE

T.E.EXT_CAPTURE is the threat of malicious entities sniffing the channel between the TOE and external mail clients in order to capture or modify authentication data or mail messages sent between the two. This can be countered by ensuring that the communication channel between the TOE and the mail clients is protected from unauthorized disclosure and modification, or O.E.EXT_CHAN.

8.1.2.8 T.E.IA

T.E.IA is the threat of TOE compromise arising due to not doing any identification or authentication of users before giving them access to the TOE. It can be directly countered by

O.E.AUTHENTICATION, which requires that the Authorized Administrator be identified and authenticated before being allowed to perform any TSF-mediated activities.

8.1.2.9 T.E.INT_CAPTURE

T.E.INT_CAPTURE is the threat of malicious insiders sniffing the channel between the TOE and internal mail servers in order to capture or modify authentication data or mail messages sent between the two. This can be countered by ensuring that the communication channel between the TOE and the mail servers is protected from unauthorized disclosure and modification, or O.E.INT_CHAN.

8.1.2.10 T.E.MASQUERADE

T.E.MASQUERADE is the threat of a malicious entity executing a process that masquerades as the TOE. If successful, such a threat may allow capture of identification and authentication data for a legitimate user of the TOE thus allowing the malicious agent unauthorized access to the TOE. O.E.TRUSTED_PATH counters this threat by ensuring that users only identify and authenticate themselves to the TOE through some trusted path.

8.2 Security Requirements Rationale

8.2.1 Security Functional Requirements Rationale for the TOE

Tables 11 and 12 demonstrate the correspondence between the security objectives listed in Sections 4.1 to the security functional requirements identified in Sections 5.1.

Table 11 - Mappings Between TOE Security Objectives and TOE Security Functional Requirements

Objectives	Requirements
O.CONFIGURABILITY	FMT_MOF.1(1), FMT_MOF.1(2), FMT_MTD.1, FMT_SMR.1
O.CONTENT_FILTER	FAU_SAA.1(2), FAU_ARP.1(2)
O.LOG	FAU_GEN.1, FAU_SEL.1, FAU_SAR.1, FAU_STG.1
O.MAIL_POLICY	FAU_ARP.1(3), FAU_SAA.1(3), FAU_ARP.1(4), FAU_SAA.1(4)
O.NOTIFICATION	FAU_ARP.1(5), FAU_SAA.1(5),)
O.REF_MEDIATION	FPT_RVM.1(1)
O.SPAM_FILTER	FAU_ARP.1(1), FAU_SAA.1(1),
O.TOE_INTEGRITY	FPT_TST.1

Table 12 - Mappings Between TOE Security Functional Requirements and TOE Security Objectives

Requirements	Objectives
FAU_ARP.1(1)	O.SPAM_FILTER
FAU_ARP.1(2)	O.CONTENT_FILTER
FAU_ARP.1(3)	O.MAIL_POLICY
FAU_ARP.1(4)	O.MAIL_POLICY
FAU_ARP.1(5)	O.NOTIFICATION
FAU_GEN.1	O.LOG
FAU_SAA.1(1)	O.SPAM_FILTER
FAU_SAA.1(2)	O.CONTENT_FILTER
FAU_SAA.1(3)	O.MAIL_POLICY
FAU_SAA.1(4)	O.MAIL_POLICY
FAU_SAA.1(5)	O.NOTIFICATION
FAU_SAR.1	O.LOG
FAU_SEL.1	O.LOG
FAU_STG.1	O.LOG
FMT_MOF.1(1)	O.CONFIGURABILITY
FMT_MOF.1(2)	O.CONFIGURABILITY
FMT_MTD.1	O.CONFIGURABILITY
FMT_SMF.1	O.CONFIGURABILITY
FMT_SMR.1	O.CONFIGURABILITY
FPT_RVM.1 (1)	O.REF_MEDIATION
FPT_TST.1	O.TOE_INTEGRITY

8.2.1.1 O.CONFIGURABILITY

To implement O.CONFIGURABILITY, the TOE must provide administrative tools that allow the administrator to enable, disable, and configure specific functionality in the TOE. This objective is implemented in the TOE using the management components FMT_MOF.1(1) and FMT_MOF.1(2) Management of security functions behavior, and FMT_MTD.1 Management of TSF data. The assignments in these components list the specific functionality that can be enabled or disabled and the actions that can be taken for managing specific TOE data. The requirement

FMT_SMR.1 Specification of management functions ensures that TOE provides these management functions to the administrators of the TOE.

8.2.1.2 O.CONTENT_FILTER

FAU_ARP.1(2) Security alarms for content match, and FAU_SAA.1(2) Potential violation analysis for content match, implement a detect-response mechanism in the TOE for detection of inappropriate content in the email or its attachments. The assignments in these components list the types of events that indicate a match, including those events that are explicitly bypassed from this analysis and the appropriate action to be taken if such events are detected.

8.2.1.3 O.LOG

O.LOG is implemented in the TOE using relevant functional components from the audit family. FAU_GEN.1 Audit data generation, FAU_SEL.1 Selective audit, FAU_SAR.1 Audit review and FAU_STG.1 Protected audit trail storage, ensure that audit records can be reliably and selectively generated, viewed and stored, thus satisfying the objective O.LOG.

8.2.1.4 O.MAIL_POLICY

FAU_ARP.1(3), Security alarms for Mail Policy Violation, and FAU_SAA.1(3) Potential violation analysis for mail policy, implement a detect-response mechanism in the TOE for detecting violation of TOE policy for mail sent from or received by specific users, groups or domains, or messages containing specific subject line content. The assignments in these components list the types of events that indicate a match, including those events that are explicitly bypassed from this analysis and the appropriate action to be taken if such events are detected.

The detect-response mechanism provided by FAU_ARP.1(4), Security alarms for Encrypted Mail Policy Violation, and FAU_SAA.1(4) Potential violation analysis for encrypted mail policy, include the rules for accepting or denying encrypted mail sent from or received by specific users, groups or domains.

8.2.1.5 O.NOTIFICATION

FAU_ARP.1(5) Security alarms for system alert notification, and FAU_SAA.1(5) Potential violation analysis for system alert notification, implement O.NOTIFICATION as a detect-response mechanism in the TOE. The assignments in these components list the types of events that indicate the failure and subsequent recovery of TOE components and the appropriate action to be taken if such events are detected.

8.2.1.6 O.REF_MEDIATION

O.REF_MEDIATION requires that inbound or outbound mail passing through the TOE unless explicitly bypassed, be examined by each of the TOE's configured filters before being forwarded to its destination. The component FPT_RVM.1(1) Non-bypassability of the TSP, directly implements this objective.

8.2.1.7 O.SPAM_FILTER

FAU_ARP.1(1) Security alarms for spam detection, and FAU_SAA.1(1) Potential violation analysis for spam detection, implement O.SPAM_FILTER as a detect-response mechanism in the TOE. The assignments in these components list the types of events that indicate the presence

of spam, including those events that are explicitly bypassed from this analysis and the appropriate action to be taken if such events are detected.

8.2.1.8 O.TOE_INTEGRITY

O.TOE_INTEGRITY is achieved by ensuring the integrity of stored TSF executable code and TOE data and by running periodic tests that demonstrate the correct operation of the TSF. These tests are directly implemented in the TOE by FPT_TST.1 TSF testing.

8.2.2 Security Functional Requirements Rationale for the IT Environment

Tables 13 and 14 demonstrate the correspondence between the security objectives listed in Sections 4.2 to the security functional requirements identified in Sections 5.3.

Table 13 - Mappings Between IT Environment Security Objectives and IT Environment Security Functional Requirements

Objectives	Requirements
O.E.AUTHENTICATION	FIA_UAU.2, FIA_UID.2, FIA_UID.1
O.E.BOUNDED_AUTH	FIA_AFL.1, FIA_UAU.1
O.E.DOMAIN_SEP	FPT_SEP.1
O.E.EXT_CHAN	FTP_ITC.1(2)
O.E.INT_CHAN	FTP_ITC.1(1)
O.E.NO_BYPASS	FPT_RVM.1(2)
O.E.TRUSTED_ENV	FPT_PHP.1, FPT_AMT.1
O.E.TRUSTED_INFO	FTP_ITC.1(1)
O.E.TRUSTED_PATH	FTP_TRP.1(1), FTP_TRP.1(2)
O.E.TS_INTEGRITY	FPT_STM.1

Table 14 - Mappings Between IT Environment Security Functional Requirements and IT Environment Security Objectives

Requirements	Objectives
FIA_AFL.1	O.E.BOUNDED_AUTH
FIA_UAU.1	O.E.BOUNDED_AUTH
FIA_UAU.2	O.E.AUTHENTICATION
FIA_UID.1	O.E.AUTHENTICATION
FIA_UID.2	O.E.AUTHENTICATION
FPT_AMT.1	O.E.TRUSTED_ENV

Requirements	Objectives
FPT_PHP.1	O.E.TRUSTED_ENV
FPT_RVM.1(2)	O.E.NO_BYPASS
FPT_SEP.1	O.E.DOMAIN_SEP
FPT_STM.1	O.E.TS_INTEGRITY
FTP_ITC.1 (1)	O.E.TRUSTED_INFO, O.E.INT_CHAN
FTP_ITC.1 (2)	O.E.EXT_CHAN
FTP_TRP.1(1)	O.E.TRUSTED_PATH
FTP_TRP.1(2)	O.E.TRUSTED_PATH

8.2.2.1 O.E.AUTHENTICATION

The identification and authentication requirements for O.E.AUTHENTICATION are implemented in the IT environment by FIA_UAU.2 User authentication before any action, and FIA_UID.2 User identification before any action respectively. FIA_UID.1 Timing of identification ensures that the no action is allowed on behalf of the user before that user is identified.

8.2.2.2 O.E.BOUNDED_AUTH

O.E.BOUNDED_AUTH is implemented in the IT environment by FIA_AFL.1 Authentication failure handling. The assignment in this component defines the action that the IT Security Environment must take when a brute force attack at guessing passwords is made by a malicious entity. FIA_UAU.1 Timing of authentication supports the above requirement by ensuring that the no action is allowed on behalf of the user before that user is authenticated.

8.2.2.3 O.E.DOMAIN_SEP

O.DOMAIN_SEP is implemented in the IT environment by FPT_SEP.1 TSF domain separation. The requirements of this component directly implement the objective.

8.2.2.4 O.E.EXT_CHAN

O.E.EXT_CHAN is implemented in the IT environment by FTP_ITC.1(2), which ensures that authentication information and mail fetch requests sent by external mail clients are sent over a channel that is protected from modification and unauthorized disclosure to external hackers.

8.2.2.5 O.E.INT_CHAN

O.E.INT_CHAN is implemented in the IT environment by FTP_ITC.1(1), which ensures that authentication information and mail fetch requests sent by the TOE to the internal mail servers is protected from modification and unauthorized disclosure to malicious insiders.

8.2.2.6 O.E.NO_BYPASS

O.E.NO_BYPASS is implemented in the IT environment by FPT_RVM.1(2) Non-Bypassability of the TOE, which ensures that the TOE is placed in a location where it is capable of processing every message without being bypassed.

8.2.2.7 O.E.TRUSTED_ENV

To implement O.E.TRUSTED_ENV, the TOE must reside in location that is not susceptible to physical attacks. FPT_PHP.1 Passive detection of physical attack implements the above objective in the IT environment by ensuring that any such attack is detected. The integrity of the abstract machine over which the TOE executes is ensured through the requirements of FPT_AMT.1 Abstract Machine Testing.

8.2.2.8 O.E.TRUSTED_INFO

The integrity of information received by the TOE from trusted external subsystems can be ensured by implementing FTP_ITC.1(1) Inter-TSF trusted channel, in the IT environment. The requirements in this component ensure that any information requested by the TOE from external subsystems can be relied upon.

8.2.2.9 O.E.TRUSTED_PATH

O.TRUSTED_PATH is implemented in the IT Environment by FTP_TRP.1(1) Trusted Local Path. and FTP_TRP.1(2) Trusted Remote Path. These requirements ensure the presence of a trusted path for local and remote users to authenticate themselves to the TOE.

8.2.2.10 O.E.TS_INTEGRITY

O.E.TS_INTEGRITY or the objective of enforcing reliable time stamps is implemented in the IT environment by FPT_STM.1 Reliable Time Stamps. The requirements directly implement the objective.

8.2.3 Security Assurance Requirements Rationale

The rationale for the Security Assurance Requirements is defined in Chapter 6, Section 6.3.

8.2.4 Rationale for Satisfaction of Strength of Function Claim

SOF-basic is defined in CC Part 1 section 2.3 as: "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." Because this ST identifies threat agents with low attack potential, the claimed minimum strength of function for the TOE is SOF-basic.

The password mechanism used to authenticate the authorized administrators to the TOE is a part of the IT environment; the TOE itself has no probabilistic or permutational security mechanism that needs SOF analysis.

8.3 TOE Summary Specification Rationale

The rationale for the TOE Summary Specification is defined in Chapter 6, Section 6.2.

8.4 PP Claims Rationale

The rationale for the Protection Profile conformance claims is defined in Chapter 7, Section 7.4 Protection Profile Rationale.