Security Target

MILKYWAY NETWORKS BLACK HOLE FIREWALL

Version 3.01E2

for SPARCstations

November 1997

CEPL-5b

Executive Summary

The Communications Security Establishment (CSE) operates the Trusted Product Evaluation Program (TPEP), the goal of which is to provide third-party critical analysis and testing of commercially developed computer security products which might be used by the Government of Canada. One type of computer security product evaluated within the TPEP is the firewall. This TPEP security target documents the results of the CSE evaluation of Milkyway Networks Black Hole Firewall version 3.01E2, against the requirements specified by the *Common Criteria for Information Technology Security Evaluation* [COM96]. Details of Black Hole, in terms of its architecture, features, and evaluated configuration, can be found in the document entitled *Final Evaluation Report for Milkyway Networks Black Hole Firewall Version 3.01E2 for SPARCstations* [CSE97a].

Black Hole is designed to protect resources on an internal (private) network from users on an external (public) network. Access through the firewall is mediated on the basis of rules defined by the administrator, who defines the firewall's users, services, and rules. Black Hole includes support for user identification and authentication. It also supports host-to-host connection restrictions of: common Internet services (such as Telnet, File Transfer Protocol [FTP], HyperText Transfer Protocol [HTTP], and Gopher); the connection-oriented Transmission Control Protocol (TCP) service; and the connectionless User Datagram Protocol (UDP) service. Black Hole-protected networks can also communicate with one another through the use of a virtual private network (VPN), which establishes an encrypted channel through the external network.

Users make connection requests to Black Hole (although they may not be aware of its presence, and may believe they are connecting directly to their desired hosts). A Black Hole subsystem (called The Oracle) mediates each connection request, based upon the source and destination addresses of the request, and possibly based upon other information (such as a user identity). If a connection is permitted, a proxy is created which transfers data between the endpoints of the connection. Each endpoint appears to be talking directly to the other, when, in fact, it is communicating with the proxy. Black Hole performs filtering at the: IP (Internet Protocol) *packet* level; TCP *circuit* level; and *application* layer (FTP and mail).

Black Hole detects and records occurrences of security-relevant events, as defined by an administrator. These records can then be sorted and reviewed. The administrator can also configure alarms, which are triggered based on patterns in the collected audit information.

A Black Hole system administrator must be fully cognizant of the security policies and objectives of the internal network which is under Black Hole's protection. It is assumed that system administration of the Black Hole product is conducted in a benign environment (that is, system administrators do not have malicious intent, and do not make errors), and that the Black Hole product is under secure physical control at all times. There are no non-administrative accounts on Black Hole.

The evaluated configuration of Black Hole runs on the SPARCstation 5, 10 and 20 hardware platforms, utilizing a stripped-down version of the Sun OS 4.1.4 operating system, upon which run the applications that compose the Black Hole product. The evaluation team has determined that the highest Evaluation Assurance Level (EAL) for which Milkyway Networks Black Hole Version 3.01E2 satisfies all of the requirements is an EAL3-Augmented rating, as described in

Chapter 4, IT Security Requirements. All components satisfied an EAL3 level of assurance, with some exceeding the EAL3 level and obtaining either the higher EAL4 or EAL7 levels of assurance.

The EAL3-Augmented rating means that an analysis of Black Hole's security functions, using a functional and interface specification and the high-level design of the subsystems of the product, was performed in order to understand the security behavior of the product. This analysis was supported by: independent testing of the security functions; evidence of developer "gray box" testing; complete, independent confirmation of the developer's test results; and evidence of the developer's search for obvious vulnerabilities (for example, those in the public domain). As well, there is added assurance through both development environment controls, and automated configuration management of the target of evaluation (TOE).

Although a firewall evaluated at the EAL3 level of assurance is generally sufficient for mediating access between the Internet and a Protected B network, or for connecting two identically-labeled, single-level networks requiring need-to-know separation, the appropriateness for such a firewall in a particular environment ultimately rests with the certification/accreditation authority of the site. Sample network configurations for which Black Hole might be used are provided in the final evaluation report.

Foreword

This security target is being issued by the Communications Security Establishment (CSE) to document how Milkyway Networks Black Hole Firewall Version 3.01E2 meets requirements taken from the *Common Criteria for Information Technology Security Evaluation* (CC) [COM96]. The requirements met by Black Hole are detailed in Chapter 4 — IT Security Requirements. This report provides the details of each of these requirements, and how each requirement is met.

Note: MILKYWAY Networks has recently changed the name of this product to "SecurIT Firewall", the version number will remain the same.

Readers who are interested in: an overall description of Black Hole; its evaluated configuration; recommendations for use; and the evaluation team's security analysis of the product, are directed to the *Final Evaluation Report for Milkyway Networks Black Hole Firewall Version 3.01E2 for SPARCstations* [CSE97a].

For the following: information on the Trusted Product Evaluation Program (TPEP); specifics of how Black Hole was evaluated; copies of the CC; and additional copies of this report or other CSE reports, contact:

TPEP Manager
Communications Security Establishment
PO Box 9703, Terminal
Ottawa, Ontario K1G 3Z4
Tel: (613) 991-7434
Fax: (613) 991-7411

For advice on how this evaluated product could be used as part of your information security solution, contact your Departmental Security Officer, or:

ITS Systems Consulting
Communications Security Establishment
PO Box 9704, Terminal
Ottawa, Ontario K1G 3Z4
DND Clients: (613) 991-7470
non-DND Clients: (613) 991-7546
Fax: (613) 991-7411

For Black Hole product information, contact:

Milkyway Networks Corporation
2650 Queensview Drive, Suite 150
Ottawa, ON K2B 8H6
Tel: (613) 596-5549
Fax: (613) 596-5615
http: //www.milkyway.com

Table of Contents

List of Tables

List of Abbreviations and Acronyms

ACM	configuration management assurance class
ADO	delivery and operation procedure assurance class
ADV	development procedures assurance class
AGD	guidance documentation assurance class
AL	assurance level
ALC	life-cycle support assurance class
ATE	testing assurance class
AVA	vulnerability assessment assurance class
CC	*Common Criteria for Information Technology Security Evaluation*
CCOR	Common Criteria Observation Report
CEPL	Canadian Evaluated Products List
CI	configuration item
CM	configuration management
CSE	Communications Security Establishment
CTCPEC	Canadian Trusted Computer Product Evaluation Criteria
CVS	Concurrent Versioning System
DBMS	database management system
DDTS	Distributed Defect Tracking System
DNS	domain name service
EAL	Evaluated Assurance Level
FAU	auditing functional class
FDP	data protection functional class
FER	final evaluation report
FIA	I&A functional class
FPT	security function protection functional class
FTP	File Transfer Protocol
GUI	graphical user interface
HTTP	Hypertext Transfer Protocol
I&A	identification & authentication
ICMP	Internet Control Message Protocol
ID	identification
IP	Internet Protocol
ISS	Internet Security Scanner
IT	information technology
ITS	information technology security
NCSA	National Computer Security Association
NIC	network interface card
PB	Information is designated Protected-B
POST	power-on self test
QA	quality assurance
RCS	Revision control System
RDBMS	relational database management system
SFP	security function policy
SMTP	Simple Mail Transfer Protocol

SQL	Structured Query Language
SSL	secure sockets layer
ST	security target
TCP	Transmission Control Protocol
TCSEC	Trusted Computer Security Evaluation Criteria
TOE	target of evaluation
TPEP	Trusted Product Evaluation Program
TRA	threat and risk assessment
TSC	TOE scope of control
TSF	target of evaluation (TOE) security function
TSP	TOE security policy
UDP	User Datagram Protocol
VPN	virtual private network
WAIS	Wide Area Information Service
WWW	World Wide Web

Introduction

This chapter presents: an overview of security targets (STs), as defined by the *Common Criteria for Information Technology Security Evaluation* (CC) [COM96]; a brief description of the Black Hole product; and a description of the organization of this document.

Security Target Overview

The purpose of an ST is to provide: a description of the environment in which a security product is to be used; the security objectives that would counter the threats in that environment; the security requirements that would be necessary to achieve those security objectives; and a description of how those requirements are met. An ST is structured in the language of the CC.

Because it defines a security product's: environment; threats; security objectives; and security requirements, the ST becomes a basis for agreement between the developers, evaluators, and consumers on the security properties of the product, and on the scope of the evaluation. The audience for an ST is not confined to those responsible for the design of the product and for its evaluation, but may also include those responsible for managing, marketing, purchasing, installing, configuring, operating and/or using the product.

To this end, this document describes security objectives and requirements, as well as the functional and assurance security measures provided by Black Hole.

Black Hole Overview

Black Hole is a software product, designed to isolate private networks from untrusted public networks in a transparent manner. Black Hole supports user identification and authentication (I&A). It also supports host-to-host connection restrictions of: common Internet services (such as Telnet, File Transfer Protocol [FTP], HyperText Transfer Protocol [HTTP] and Gopher); the connection-oriented Transmission Control Protocol (TCP) service; and the connectionless User Datagram Protocol (UDP) service. It also offers: a graphical administrative user interface (the GUI Admin.); alarm messaging; traffic statistics; and audit tools.

For a detailed description of the Black Hole product, see *Final Evaluation Report for Milkyway Networks Black Hole Firewall Version 3.01E2 for SPARCstations* [CSE97a].

Black Hole has been evaluated against security requirements defined in the CC. The *functional* classes of requirements include: identification and authentication (FIA); auditing (FAU); data protection (FDP); and security function protection (FPT). The *assurance* classes of requirements include the documentation of: development procedures (ADV); life cycle support (ALC); testing (ATE); configuration management (ACM); guidance documentation (AGD); delivery and operation procedures (ADO); and vulnerability assessment (AVA). For details, see Chapter 4, IT Security Requirements.

Document Organization

This report is composed of five chapters and two appendices.

- Chapter one provides a high-level description of Black Hole.
- Chapter two describes the security environment, in terms of the threats, organizational security policies, and usage assumptions in which Black Hole is designed to be used.
- Chapter three describes the security objectives of Black Hole (that is, its role in reducing the threats described in chapter two).
- Chapter four lists the CC security requirements (both functional and assurance) that are satisfied by Black Hole.
- Chapter five provides a rationale that this ST defines a complete and cohesive set of security requirements.
- Appendix A provides an explanation of Black Hole's security policy paradigm.
- Appendix B discusses Black Hole's security policies.
- A glossary and a bibliography follow the appendices.

Document Conventions

To improve the readability of this report, certain document conventions are followed:

- Document titles are in *italics*.
- Emphasized text is in *italics* or **bolded** font, depending upon the context.
- Technical terms (such as filenames), and UNIX commands and processes, are in the `courier` font.
- Bibliography references are enclosed in [brackets].
- There are special text conventions which apply only to Chapter 4, IT Security Requirements. Those conventions are explained as they are used.

1. Target of Evaluation Description

The purpose of this chapter is to provide the reader with a high-level overview of Black Hole, the target of evaluation (TOE). The bulk of this report centers upon how Black Hole meets its requirements, without focusing upon how the product works. Consequently, there is a need to describe the functions and parts of Black Hole, so that the reader is given a context for understanding the rest of the report. (Readers interested in further details of the product and its internal workings are directed to the product's final evaluation report (FER) [CSE97a]).

Black Hole is designed to protect resources on an internal (private) network from users on an external (public) network. Access through the firewall is mediated on the basis of rules defined by an administrator. The administrator uses a GUI at the console to: define the users, services, and rules that are used by Black Hole; define the events that are to be audited and logged; review the audit logs so created; and perform other maintenance functions.

When the product is first installed, all services are disabled; the system administrator must explicitly enable a service before it can be used. The administrator can then implement a security policy, by configuring components on the firewall. The firewall contains an access control database, which is used to determine if a request for service should be allowed or denied. If a request is made for which no rule in the access control database exists, the request is denied and logged.

Users make connection requests to Black Hole (although they may not be aware of its presence, and may believe they are connecting directly to the desired host). A Black Hole subsystem (called The Oracle) mediates the connection request based upon the source and destination addresses of the request, and possibly upon other information (such as a user identity). If a connection is permitted, a proxy is created which transfers data between the endpoints of the connection. Each endpoint appears to be talking directly to the other, when in fact it is communicating with the proxy. Black Hole performs filtering at: the Internet Protocol (IP) *packet* level; the TCP *circuit* level; and (for FTP and mail) the *application* layer.

Black Hole detects and records occurrences of security-relevant events, as defined by the administrator. These events are detected by the appropriate subsystem and forwarded to the logging daemon, which writes them into the audit file. The records can then be sorted and reviewed.

1.1 Evaluated Configuration

It is recognized that not all sites will use the product in an identical fashion. In order to provide the maximum amount of useful information, the evaluation team has identified what it believes to be a typical configuration and environment for government clients. This *evaluated configuration* (see Chapter 8, Final Evaluation Report for Milkyway Networks Black Hole Firewall Version 3.01E2 for SPARCstations [CSE97a]) describes the features, configurations, and options (both hardware and software) that the product had during its evaluation. With respect to the CC requirements, these available features may fall into one of the following three classes:

- **Vital features.** Vital features meet at least one of the security requirements. If the feature were removed, then the corresponding requirement(s) would no longer be met.

- **Benign features.** Benign features meet no CC requirements, but pose no threat. If the features are excluded, the security requirements will still be met. Conversely, their use will not adversely affect the product's adherence to the ST requirements.

- **Prohibited features.** Prohibited features violate the ST requirements. (The only reason for mentioning these in this report or in the FER is because they are mentioned in the vendor's customer documentation.). These features, if included in the shipped product, must be removed upon installation.

The evaluated configuration of Black Hole comprises the vital and benign features (described above) and excludes the prohibited features. **It is therefore emphasized that running the product outside its evaluated configuration negates the security claims made in this report and in the final evaluation report**.

2. Security Environment

This chapter describes the security environment for this ST, which consists of the following:

- **Threats**. Threats are either countered directly by Black Hole, or are addressed by its operating environment. Threats drive the selection of *security objective (*see Chapter 3, Security Objectives).

- **Assumptions**. Assumptions are constraints placed on the operating environment (connectivity, network functionality, etc.). Assumptions are chosen to satisfy non-IT security objectives (see Chapter 3, Security Objectives).

Note: the tags in the left margin (T.SPOOF, A.CASCADE, etc.) exist solely to provide a convenient shorthand for referring to items in Chapter 5, Rationale; they have no other significance.

2.1 Threats to Security

This section describes the threat environment for Black Hole.

2.1.1 Threats Addressed by the Target of Evaluation

Black Hole specifically addresses the threat possibilities discussed below.

T.ACORRUPT Unauthorized users on the external network may logically tamper with audit data stored on the firewall. Such tampering may include: deleting or modifying audit data stored on the firewall, causing a proliferation of audit events such that the available audit storage is filled, or causing a failure of firewall audit functionality such that security-critical events are not logged.

T.DCORRUPT Unauthorized users on the external network may logically tamper with the security configuration data stored on the firewall. Such tampering may include: deletion, modification, or destruction of the security attributes on the firewall used in enforcing the firewall security policy.

T.PROBE Unauthorized users on the external network may illicitly perform probes or launch attacks against the internal network or the firewall itself. Such attacks might yield information about the configuration of the internal network, and the hosts of which it is composed, and might include: scanning ports of hosts, probing ranges of IP addresses, or running automated attack tools such as SATAN or ISS. The specific threat to be countered is that such obvious probes and attacks will not be audited.

T.REPLAY Unauthorized users on the external network may replay user authentication information to gain logical access to the internal network, or to the firewall

itself. User authentication information could be collected through the use of sniffing utilities on the external network.

T.SPOOF Unauthorized users on the external network may spoof the address information contained in the IP packet header to make it appear that a packet is from a host on the internal network, although it in fact originated from the external network. Spoofed packets may be used to probe or attack hosts on the internal network, or to exploit trust relationships between hosts on the internal network.

T.SYSACC Unauthorized users on the external network may gain access to the firewall administrative account. Such access could be achieved by defeating the access control mechanisms for the firewall administrative account. Attacks might include: brute-forcing the administrative password.

T.SACCESS Unauthorized users on the external network may gain access to services offered by the internal network. The specific threat is the exploitation of a lack of access control mechanisms and policies, as might be expected in the absence of a firewall.

T.FLAW Unauthorized users on the external network may exploit flaws in the implementation of certain services or protocols to gain access to the internal network, or to the firewall itself. Such attacks might include, for example, the exploitation of FTP or sendmail “bugs”.

2.1.2 Threats to be Addressed by the Environment

Black Hole does not explicitly address the threat possibilities discussed below. They must be countered by the environment, countered by procedural means, or accepted as potential system risks.

T.DENIAL A user from the external network may attack the firewall with the goal of causing a denial of service. Such an attack might be perpetrated, for example, by opening an excessive number of connections, or by generating an inordinate amount of network traffic. While some firewalls can partially protect against this type of threat, it is difficult to completely eliminate the risk.

T.ABUSE Users on the internal network may attack other hosts on the internal network, or willfully disclose information to untrusted users or the external network. Also, trusted users on the external network may abuse their privileges.

T.ACCIDENT Users on the internal network may inadvertently disclose information to untrusted users or the external network, through negligence (for example, disclosure through electronic mail).

T.SNIFF Network traffic on the internal or external networks can be collected and

analyzed by personnel on those networks.

T.SESSION Sessions established from the external network to hosts on the internal network may be hijacked.

T.TROJAN Users on the internal network may import hostile executables. For example, a user may download a trojan horse through a web browser.

T.ADMIN Administrators may compromise security through negligence (incorrect configuration) or through deliberate, hostile intent.

T.PHYSICAL An attacker may gain physical access to the firewall and launch direct physical attacks (circumventing operating system controls through floppy boot, for example), or bypass the firewall altogether by connecting the internal and external networks together.

2.2 Secure Usage Assumptions

The following secure usage assumptions are made:

A.COMMS Information transmitted through wire lines is protected in a manner commensurate with the data sensitivity, or an explicit judgment has been made that the information can be transmitted plaintext.

A.SECURE The firewall is physically secure with access limited to authorized personnel only.

A.CUSTOMIZE No custom applications may be installed on the firewall. Examples of custom applications include: untrusted code downloaded off the Internet, such as a news server, or new services in general.

A.LOCAL Administrators interact with the firewall at the local console.

A.NOEVIL Administrators are assumed to be non-hostile and trusted to perform their duties correctly, ensuring that the firewall is correctly configured and maintained. This includes both installation and normal operation of the firewall.

A.SINGL_PT The firewall is the only interconnection point between networks.

A.CASCADE The internal network must be accredited up to the level of information being processed, and must not have cascading network connections to higher data sensitivities.

3. Security Objectives

This section lists the security objectives for this ST.

The purpose of the objectives is to identify the high-level requirements that address the threats identified as part of the security environment. *IT objectives* are addressed by the selection of specific CC requirements (see Chapter 4, IT Security Requirements), whereas *non-IT objectives* are addressed by secure usage assumptions (see Section 2.2, Secure Usage Assumptions).

3.1 IT Security Objectives

O.ACCESS All accesses between subjects and objects shall be mediated by the firewall. The firewall shall be capable of revoking the access privileges of subjects to objects. Residual information must be handled with an appropriate object reuse mechanism.

O.ADMIN There shall be a clearly-defined administrative role on the firewall such that only authorized administrators are able to perform security-relevant functions. The firewall shall provide administrative functions to configure the access permissions of subjects and objects and manage the audit trail.

O.AUDIT The firewall shall be capable of generating and storing audit information for all security-relevant events. The audit trail shall be human-readable and suitable for analysis by searching and sorting tools. In the case of audit storage exhaustion or audit system failure, the firewall shall be capable of suspending the occurrence of auditable events.

O.IDENT Users of the firewall must be uniquely identified and authenticated. The firewall shall support both reusable and one-time authentication methods. Administrators need only be authenticated with reusable passwords.

O.PROTECT The firewall shall protect from tampering, destruction, or modification all firewall data structures, configuration information, and executables responsible for enforcing the firewall security policy, including the audit trail. The firewall shall implement a reference validation mechanism, and enforce a separate domain for execution.

O.ASSURE The firewall shall be designed and implemented such that the probability of implementation flaws being exploited by intruders is minimized, within the constraints of the target environment envisioned for this ST. The level of assurance for this ST is EAL3-Augmented.

3.2 Non-IT Security Objectives

O.MANAGE The firewall, and the internal networks, must be installed, managed, and operated in a manner which maintains the security policy.

O.PHYSICAL	The firewall must be maintained in a physically secure location, with access restricted to authorized administrators.
O.CONNECT	The firewall must be the only point of interconnection between networks.
O.LOCAL	Administration must be performed from the local console only.
O.TRAIN	Administrators and users must be trained to establish and maintain sound security policies and practices.
O.REVIEW	Audit facilities must be used and managed effectively. In particular, audit logs must be reviewed regularly.

4. IT Security Requirements

This chapter defines the IT security requirements (as defined in the CC) which are satisfied by Black Hole. Each requirement is chosen to satisfy one or more security objectives (identified in Chapter 3, Security Objectives), and is structured in terms of:

- any dependencies that the requirement may have (on other CC requirements);
- a description of how the requirements are met (that is, the security functions that are claimed to meet the functional requirements, or the measures taken to meet the assurance requirements); and
- the security objective to which the requirement is related.

Text filled in by the evaluation team in order to complete unresolved CC requirements is marked in ***bold italic***. The evaluators' remarks that explain how each element is met are marked in Helvetica.

4.1 IT Functional Requirements

Table I lists the functional requirements that are met by Black Hole. Following the table are the elements of those requirements, and a description of how the evaluated firewall meets each element.

Table I - Security Functional Requirements

Security Audit	
FAU_GEN.1	Audit Data Generation
FAU_MGT.1	Audit Trail Management
FAU_POP.1	Human Understandable Format
FAU_PRO.1	Restricted Audit Trail Access
FAU_SAR.1	Restricted Audit Review
FAU_SAR.3	Selectable Audit Review
FAU_STG.3	Prevention of Audit Data Loss
User Data Protection	
FDP_ACC.2	Complete Object Access Control
FDP_ACF.2	Multiple Security Attribute Access Control
FDP_ACF.4	Access Authorisation and Denial
FDP_RIP.3	Full Residual Information Protection on Allocation
FDP_SAM.1	Administrator Attribute Modification
FDP_SAQ.1	Administrator Attribute Query

Identification and Authentication	
FIA_ADA.3	Expanded User Authentication Data Administration
FIA_ADP.1	Basic User Authentication Data Protection
FIA_ATA.1	User Attribute Initialisation
FIA_ATA.2	Basic User Attribute Administration
FIA_ATD.2	Unique User Attribute Definition
FIA_UAU.1	Basic User Authentication
FIA_UAU.2	Single-use Authentication Mechanisms
FIA_UID.2	Unique Identification of Users
Protection of the Trusted Security Functions	
FPT_AMT.2	Abstract Machine Testing During Start-Up
FPT_REV.1	Basic Revocation
FPT_RVM.1	Non-Bypassability of the TSP
FPT_SEP.1	TSF Domain Separation
FPT_TSA.2	Separate Security Administrative Role
FPT_TSM.1	Management Functions

4.1.1 Class FAU Security Audit

Security audit requirements are concerned with: recording the occurrence of security-relevant events; associating those events with individual user identities; and ensuring the protection and integrity of the audit trail.

FAU_GEN.1 Audit Data Generation

Dependencies:

FIA_UID.1 Basic User Identification

Elements:

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;

Black Hole audits the normal start-up and shutdown of the auditing subsystem.

b) All auditable events for the ***basic*** level of audit, as defined in all functional components included in Table II;

Table II - Audit Events

Component	**Event**
FAU_GEN.1	-
FAU_MGT.1	Any attempt to perform an operation on the audit trail
FAU_POP.1	Any specific operation performed to process audit data stored in the audit trail
FAU_PRO.1	Any attempt to read, modify or destroy the audit trail
FAU_SAR.1	-
FAU_SAR.3	-
FAU_STG.3	-
FDP_ACC.2	-
FDP_ACF.2	All requests to perform an operation on an object covered by the SFP
FDP_ACF.4	All attempts to specify the granting or denying of access to an object
FDP_RIP.3	-
FDP_SAM.1	All attempts to modify security attributes, including the identity of the target of the modification attempt
FDP_SAQ.1	All attempts to query security attributes, including the identity of the target of the query attempt
FIA_ADA.3	All requests to use TSF authentication data management mechanisms
FIA_ADP.1	All requests to access user authentication data
FIA_ATA.1	All requests to use the user attribute administration functions and identification of the user attributes that have been modified
FIA_ATA.2	All requests to use the user attribute administration functions and identification of the user attributes that have been modified
FIA_ATD.2	-
FIA_UAU.1	Any use of the authentication mechanism
FIA_UAU.2	Any use of the authentication mechanism
FIA_UID.2	All attempts to use the user identification mechanism, including the user identity provided

FPT_AMT.2	Execution of the tests of the underlying machine and results of the tests
FPT_REV.1	-
FPT_RVM.1	-
FPT_SEP.1	-
FPT_TSA.2	Use of a security-relevant administration function; allocation of a function to a security administrative role; explicit requests to assume the security administrative role
FPT_TSM.1	All attempts to modify (set and update) TSF configuration parameters

Black Hole generates the following categories of log messages:

- system startup/shutdown;
- administrator logon/logoff;
- GUI startup/shutdown;
- subsystem (Guardian, The Oracle, `vpnd`, proxy, etc.) startup/shutdown;
- administrator changes user record;
- administrator changes rule;
- administrator edits or backs up a database or configuration file;
- administrator rolls over audit data log;
- user authentication;
- user or administrator changes user's password;
- toggling of transparent mode;
- kernel-level filtering detects prohibited packet (ICMP redirect, etc.);
- refusal of a connection;
- establishment or termination of a connection;
- attempted use of FTP commands;
- attempted use of prohibited mail commands (`wiz`, `debug`, etc.);
- file system filling up;
- `Syslogd` startup/shutdown;
- alarms sent; and
- changing of system time/date.

Black Hole administrators are considered to be trusted personnel. Consequently, the auditing of administrator events is not designed to prevent administrators from doing anything malicious, but rather as a means of housekeeping, so that an administrator can trace the actions that were taken to determine how the current system state was produced. Administrator actions that change the system state

(for example, logging in, changing databases) are security-relevant and, therefore, auditable. Administrator actions that do not change this state (such as a non-destructive read of a database) are not security-relevant, and need not be audited.

c) Based on all functional components included in the ***ST, [assignment: other auditable events]. No other events were specified.***

No other events are specified.

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, object identity, and ***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 ST, ***no other audit-relevant information was selected.***

The TSF generates audit messages that comprise three parts:

- a message class or level indicating the significance of the message;
- a code number used to trace the message to the program code; and
- a descriptive component.

The descriptive component contains information such as: date and time of the event; type of event; subject identity; object identity; and success or failure of the event.

Related security objective:

O.AUDIT

FAU_MGT.1 Audit Trail Management

Dependencies:

FAU_STG.1 Permanent Audit Trail Storage

Elements:

FAU_MGT.1.1 The TSF shall provide the authorized administrator with the ability ***to create and empty*** the audit trail.

The file structure for the audit trail is created at installation time. To create a new audit file, administrators can use the text editor in the GUI to save an empty file with the desired name. The audit subsystem can be configured to use the new file by editing `syslog.conf`.

The audit trail can be emptied by rolling the audit logs. Black Hole maintains nine audit files, each for a different time period. When the audit log is rolled, the contents of the first audit file are moved to the second, the contents of the second to the third, and so on, leaving the first file empty. The contents of the ninth file are lost.

For example, if an administrator decided to roll the audit log on a daily basis the files would be named `audit`, `audit.0`, `audit.1`, and so on. On the tenth day, the contents of the file named `audit.1` would be replaced with the contents of the file named `audit`, `audit.2` with `audit.1`, and so on leaving the file named `audit` empty. "The contents of `audit.7` will be overwritten with those of audit.6. Administrators can save the contents of audit.7 to tape before rolling if it is desirable to avoid overwriting them."

Related security objective:

O.ADMIN

FAU_POP.1 Human Understandable Format

Dependencies:

FAU_STG.1 Permanent Audit Trail Storage

Elements:

FAU_POP.1.1 The TSF shall be able to generate a human understandable presentation of any audit data stored in the permanent audit trail.

The Black Hole process `syslogd` generates tab-delimited, human-readable text files as output.

Related security objective:

O.AUDIT

FAU_PRO.1 **Restricted Audit Trail Access**

Dependencies:

FAU_STG.1 Permanent Audit Trail Storage

FPT_TSA.1 Basic Security Administration

Elements:

FAU_PRO.1.1 The TSF shall restrict access to the audit trail to the authorized administrator.

The audit trail review tools are accessible only from the console. The console is protected by a login/password mechanism, with the password known only to authorized administrators.

Related security objective:

O.PROTECT

FAU_SAR.1 **Restricted Audit Review**

Dependencies:

FAU_STG.1 Permanent Audit Trail Storage

FPT_TSA.1 Basic Security Administration

FAU_PRO.1 Restricted Audit Trail Access

Elements:

FAU_SAR.1.1 The TSF shall provide audit review tools, with the ability to view the audit data.

The GUI provides a text editor to review the audit trail. Administrators can also use non-destructive audit review tools such as `sort`, `more`, `cat`, and `grep`, from the shell prompt.

FAU_SAR.1.2 The TSF shall restrict use of the audit review tools to the authorized administrator.

The audit trail review tools are accessible only from the console. The console is protected by a login/password mechanism, with the password known only to authorized administrators.

Related security objective:

O.AUDIT

FAU_SAR.3 Selectable Audit Review

Dependencies:

FAU_SAR.1 Restricted Audit Review

Elements:

FAU_SAR.3.1 The TSF shall provide audit review tools with the ability to perform searches and sorting of audit data based on:

- ***time and date;***
- ***subject identity;***
- ***object identity;***
- ***event type; and***
- ***success or failure.***

The Black Hole process `bhstatsd` monitors the audit trail by looking for instances of specific messages, and uses them to populate a database. The administrator can use pre-built reports or Structured Query Language (SQL) commands to examine the collected information.

The reporting function gives an authorized administrator the ability to design custom reports focusing on any one (or combination of): time and date; subject identity; object identity; event type; and success or failure.

To search the audit log for messages that are not filtered by `bhstatsd`, the administrator can use non-destructive shell prompt utilities such as `sort`, `more`, `cat`, and `grep`. These commands allow

for searches based on one field, or many (through Boolean combinations).

Related security objective:

O.AUDIT

FAU_STG.3 Prevention of Audit Data Loss

Dependencies:

FAU_GEN.1 Audit Data Generation

Elements:

FAU_STG.3.1 The TSF shall store generated audit records in a permanent audit trail.

Process `syslogd` collects audit messages from all Black Hole subsystems, and stores them in the file system of the underlying operating system.

FAU_STG.3.2 The TSF shall limit the number of audit records lost due to ***system audit storage exhaustion or audit failure.***

Audit system failure is defined as the termination, or indefinite blocking, of `syslogd`. Audit system exhaustion is a form of audit failure, because when `syslogd` detects a full file system, it terminates, and logs the error condition to the console (note: the message that caused the full file system is not guaranteed to be logged).

No new connections can be established with `syslogd` in a failure state. Currently active connections will be killed within a one minute time period, by a background `cron` job that continually monitors the status of `syslogd` for failure conditions.

FAU_STG.3.3 In the event of audit storage exhaustion, the TSF shall be capable of ***preventing*** the occurrence of auditable actions, except those taken by the authorized administrator.

Audit system exhaustion is a form of audit failure. No new connections can be established without the audit subsystem, and all active connections will be killed within a one minute time frame.

Related security objective:

O.AUDIT

4.1.2 Class FDP User Data Protection

User data protection requirements are concerned with access control mediation when resources on an external network attempt to access resources on an internal network (or vice versa) through a firewall. The resources that are attempting access are: the remote process; the originating host; or the originating network (that is, subjects). The resources that are the target of access are the destination host, and the destination network (that is, objects). Hosts are identified by their IP addresses; networks are identified by their IP subnets; and processes are identified by the associated user identification (ID).

FDP_ACC.2 Complete Object Access Control

Dependencies:

FDP_ACF.1 Single Security Attribute Access Control

Elements:

FDP_ACC.2.1 The TSF shall enforce the ***untrusted user access control policy*** on ***subjects and objects***, and all operations among subjects and objects covered by the SFP.

Subjects are identified as: remote user processes; source hosts; or source networks.

Objects are identified as: destination hosts, or destination networks.

The possible interactions between subjects and objects are either to allow a connection (that is, to allow communication between a subject and an object), or to deny a connection. The firewall access control policy covers both interactions, through the creation of rules. These rules define what connections will be allowed, based on security attributes (source and destination IP address; service type; time of day; and user ID with authentication); everything else is denied. The policy therefore can be summarized as, “that which is not explicitly allowed, will be denied” with the following two exceptions. One, mail can be explicitly denied by a rule; but by default, all mail connections will be allowed. Two, the FTP protocol commands can be allowed and denied, but should there be conflicting rules, the first rule in the database (not the most restrictive) is applied.

Interactions between hosts on a VPN are not covered by the access control policy. Such hosts are considered trusted, and are treated by

Black Hole as if they reside on the same network. Hence, the access control policy described here does not apply to VPN. The firewall does not claim to have total control over those entities which it cannot control (that is, hosts). It claims only to control what it passes from an input physical connection (representing the subject network) to an output physical connection (representing the object network), based on certain access requests and a specified rule set. (For more information on subjects and objects, see: Appendix Appendix A - , Black Hole Security Policy Paradigm; and Appendix Appendix B - , Black Hole Security Policies.)

FDP_ACC.2.2 The TSF shall ensure that all operations between any subject in the TSC and any object within the TSC are covered by the SFP.

The possible interactions between a subject and an object are either to allow a connection (that is, to allow communication between a subject and an object), or to deny a connection. The firewall access control policy therefore covers all possible interactions, through the creation of rules.

Related security objective:

O.ACCESS

FDP_ACF.2 Multiple Security Attribute Access Control

Dependencies:

FDP_ACC.1 Subset Object Access Control

Elements:

FDP_ACF.2.1 The TSF shall enforce the ***untrusted user access control policy*** to objects based on:

- ***time of day;***
- ***subject identity;***
- ***object identity;***
- ***service type; and***
- ***if the service type is FTP or mail, the command requested.***

Subjects are defined as: remote user processes; source hosts; or source networks. In order to gain access to an object, a subject must

provide: a source IP address; a destination IP address; and a destination port. (The destination IP address is the security attribute of the desired object.) The time of day is tracked internally, and associated with the access request, by Black Hole. Depending upon the rule associated with the access request, further attributes may be required, in the form of the subject's user ID and authenticator.

The mail protocol performs command filtering in the form of "traps" for well-known mail security holes (such as debug or wiz). This functionality is "hardwired" into the Black Hole software, and cannot be changed.,

The FTP protocol allows an administrator to explicitly deny or allow four groups of FTP commands. The four command groups are: deposit commands (`put,` etc.); retrieval commands (`get,` etc.); destructive commands (`delete`, `rename`, `chmod,` etc.); and non-destructive commands (`cd`, `dir`, `ls,` etc.).

FDP_ACF.2.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed.

No rules shall apply to VPN, as it is not covered by the untrusted user access control policy.

For all other services, the following configurable rules shall apply.

- ***if the subject is permitted to use the requested service type;***
- ***if the access request is within the permitted range of access times for that subject (except for mail);***
- ***if the subject is permitted to communicate with the specified object; and***
- ***if the subject is permitted to use the requested command (only for mail and FTP),***

then access is granted to the requested object.

The following fixed, kernel-leve rules for packet-filtering shall apply.

- ***A service request by a subject from the external network may not have a source IP address which is known to belong to the internal network (or vice versa).***
- ***A service request by a subject may not specify source routing.***
- ***A service request by a subject may not specify ICMP Redirection.***

- ***A service request by a subject may not be directed to a destination port of 514 (syslog port) on the Black Hole firewall.***

In order to gain access to an object, a subject must provide: a source IP address; a destination IP address; and a destination port. (The destination IP address is the security attribute of the desired object.) The time of day is tracked internally, and associated with the access request, by Black Hole. Depending on the rule associated with the access request, further attributes may be required, in the form of the subject's user ID and authenticator. If there is no rule for a connection request, that request is denied.

An exception to this underlying "most-restrictive" principle is mail, which is not user-session-based. Mail rules are enforced on a least-restrictive basis, and must be explicitly prohibited, per IP address.

Mail and FTP perform application-level command filtering. The filtering of mail commands is hardwired, and cannot be changed. The filtering of FTP commands is least-restrictive, but only by default; administrators may change the rules to more-restrictive, if desired.

The Black Hole kernel performs fixed, packet-level filtering.

Related security objective:

O.ACCESS

FDP_ACF.4 **Access Authorization and Denial**

Dependencies:

FDP_ACC.1 Subset Object Access Control

Develop action elements:

FDP_ACF.4.1 The TSF shall enforce the ***untrusted user access control policy*** to provide the ability to explicitly grant access based on the value of security attributes of subjects and objects.

Access requests are governed on the basis of the security attributes that are provided by subjects and objects. Access requests are associated with rules. Rules can either explicitly permit access (allow), or explicitly deny access (disallow). If there is no rule for an access request, access is denied.

FDP_ACF.4.2 The TSF shall enforce the ***untrusted user access control policy*** to provide the ability to explicitly deny access based on the value of security attributes of subjects and objects.

Access requests are governed on the basis of the security attributes that are provided by subjects and objects. Access requests are associated with rules. Rules can either explicitly permit access (allow), or explicitly deny access (disallow). If there is no rule for an access request, access is denied.

Related security objective:

O.ACCESS

FDP_RIP.3 Full Residual Information Protection on Allocation

Dependencies:

No dependencies.

Elements:

FDP_RIP.3.1 The TSF shall ensure that upon the allocation of a resource to all objects any previous information content is unavailable.

All storage objects for network data are handled by the kernel internally, such that no other process has access to them. The objects are fixed-length memory structures, called `mbufs`. When an `mbuf` is allocated, it is filled with data, thereby overwriting any previous information. During allocation, an `mbuf` cannot be read, only filled, so there is no chance to read previous information. Should data not completely fill an `mbuf`, the remaining portion is padded with zeros.

Related security objective:

O.ACCESS

FDP_SAM.1 Administrator Attribute Modification

Dependencies:

FPT_TSA.1 Basic Security Administration

[FDP_ACC.1 Subset Object Access Control, or

FDP_IFC.1 Subset Information Flow Control]

Elements:

FDP_SAM.1.1 The TSF shall enforce the ***trusted administrator access control policy*** to provide authorized administrators with the ability to modify ***the following security attributes, that are associated with subjects and objects:***

- ***user ID;***
- ***source IP address;***
- ***destination IP address;***
- ***service type;***
- ***time of day; and***
- ***for FTP, the command requested.***

The required GUI commands are only accessible from the console. The console is protected by a login/password mechanism, with the password known only to authorized administrators.

Related security objective:

O.ADMIN.

FDP_SAQ.1 Administrator Attribute Query

Dependencies:

FPT_TSA.1 Basic Security Administration

[FDP_ACC.1 Subset Object Access Control or

FDP_IFC.1 Subset Information Control]

Elements:

FDP_SAQ.1.1 The TSF shall enforce the ***trusted administrator access control policy*** to provide the authorized administrator with the ability to query ***the following security attributes associated with subjects and objects***:

- ***user ID;***

- ***source IP address;***
- ***destination IP address;***
- ***service type;***
- ***time of day; and***
- ***for FTP, the command requested.***

The required GUI commands are only accessible from the console. The console is protected by a login/password mechanism, with the password known only to authorized administrators.

Related security objective:

O.ADMIN

4.1.3 Class FIA Identification and Authentication

The purpose of I&A in a firewall is to provide the proof of identity needed to verify that a user may make use of, or enable, a service or services between hosts (where such identity is required).

I&A in Black Hole can be considered as two main categories: administrators, and network users. Administrators are authenticated at the local console, through the standard UNIX login/password mechanism, before progressing. Network users interact with Black Hole proxies, which may or may not require user authentication, depending on the type of proxy and how the rules have been configured. Specifically, only the `Telnet`, `FTP`, `Gopher`, and `HTTP` *authenticating* proxies have the capability to require user-level authentication. *Non-authenticating* proxies (`SSL; TCP; UDP; RealAudio;` and `mail`) do not support user-level authentication. As a result, some I&A requirements (specifically, FIA_UAU.1; FIA_UAU.2; and FIA_UID.2) apply only to authenticating proxies).

FIA_ADA.3 Expanded User Authentication Data Administration

Dependencies:

FPT_TSA.1	Basic Security Administration
FIA_ADP.1	Basic User Authentication Data Protection
FIA_UAU.1	Basic User Authentication

Elements:

FIA_ADA.3.1 The TSF shall provide functions for initializing and modifying user authentication data related to ***reusable and one-time password mechanisms.***

The GUI contains functions that allow an administrator to initialize or modify the user password (or the S/Key value) for any user within the rules database.

FIA_ADA.3.2 The TSF shall restrict the use of these functions on the user authentication data for any user to the authorized administrator.

The user authentication data functions are accessible only from the console. The console is protected by a login/password mechanism, with the password known only to authorized administrators.

FIA_ADA.3.3 The TSF shall allow authorized users to use these functions to modify their own authentication data in accordance with the TSP.

The gateway mode supported by authenticating proxies allows users to change their passwords. The proxy is trusted code running within the TSF, and users are permitted to modify only their own passwords.

Related security objective:

O.ADMIN

FIA_ADP.1 Basic User Authentication Data Protection

Dependencies:

FIA_UAU.1 Basic User Authentication

Elements:

FIA_ADP.1.1 The TSF shall protect from unauthorized observation, modification, and destruction authentication data that is stored in the TOE.

There are no user accounts or untrusted processes on Black Hole. Consequently, processes that access authentication data stored on Black Hole are trusted not to disclose or modify data in an unknown or unauthorized manner.

Related security objective:

O.PROTECT

FIA_ATA.1 User Attribute Initialization

Dependencies:

FIA_ATD.1 User Attribute Definition

FPT_TSA.1 Basic Security Administration

Elements:

FIA_ATA.1.1 The TSF shall provide the ability to initialize user attributes with provided default values.

Note: User attributes are interpreted as all user attributes *except user authentication data* (which is covered elsewhere, such as under FIA_ADA.3).

New records will be accepted, as long as the specified user ID is unique. If an administrator specifies only a user ID (and no other information), the default setting will be for the new user to be disabled; consequently, the user will have no defined password mechanism. For the rule to work in this case, the user must be explicitly enabled, and an appropriate password mechanism must be specified.

An administrator can also set up a “default” user record, and copy the attributes of that record into each new user record when it is created.

Related security objective:

O.ADMIN

FIA_ATA.2 Basic User Attribute Administration

Dependencies:

FIA_ATD.1 User Attribute Definition

FPT_TSA.1 Basic Security Administration

Developer actin elements:

FIA_ATA.2.1 The TSF shall provide the ability to ***display and modify*** user attributes.

Note that user attributes are interpreted as all user attributes *except user authentication data* (which is covered elsewhere, such as under FIA_ADA.3).

An administrator is able to view and change the attributes associated with each user through the GUI (as discussed in the *Black Hole Administration Guide* [MIL96a]).

FIA_ATA.2.2 The TSF shall limit the ability to modify user attributes to only the authorized administrator.

User attribute modification functions are accessible only from the console. The console is protected by a login/password mechanism, with the password known only to authorized administrators.

Related security objective:

O.ADMIN

FIA_ATD.2 Unique User Attribute Definition

Dependencies:

ADV_FSP.1 TOE and security policy

Elements:

FIA_ATD.2.1 The TSF shall provide, for each user, a unique set of security attributes necessary to enforce the TSP.

This requirement is concerned with achieving a one-to-one association between users and their security attributes. That is, each user must have his or her own set of security attributes, which do not necessarily need to have unique values.

The rules database consists of several types of records.

- A *user record* contains (among other things): a user's full name; authentication information; and the user identifier (used for challenges). There is a unique user record for each user.
- A *service record* determines which proxy will be started for a particular address/port combination.
- An *address record* specifies a single address, or a range of addresses (subnet).
- A *time record* specifies a range of times.
- A *rule record* includes a reference to associated records, for: time; source address; destination address; service; and, if the rule requires authentication, the associated user record. Each rule record also includes an action: drop; disallow; challenge; or allow.

Thus, each rule record can be associated with a unique user record, which in turn corresponds to a unique user. It is possible that different rule records would point to the same service, time or address records, but this would only occur if the values were the same for multiple rules.

Related security objective:

O.ACCESS

FIA_UAU.1 **Basic User Authentication**

Dependencies:

FIA_UID.1 Basic User Identification

FIA_ADA.1 User Authentication Data Initialization

Elements:

FIA_UAU.1.1 The TSF shall authenticate any user's claimed identity prior to performing any functions for the user.

This requirement applies only to administrators at the local console, and to *authenticating* proxies.

An administrator at the console is treated like a user on a UNIX system: authentication via password is required before progressing.

An authenticating proxy can be configured to require user authentication before any connection request is allowed to complete.

Administrators should be aware that, while the use of non-authenticating proxies does not interfere with the functionality of authenticating proxies, **non-authenticating proxies do not meet this requirement**. In addition, **authenticating proxies must be configured properly (that is, to use challenge mode, and a fixed password mechanism) in order to meet this requirement**.

It is possible, though cumbersome, to force users to authenticate even when using non-authenticating proxies. This can be accomplished by configuring the rules for non-authenticating proxies to use `challenge` mode, and setting the `transparent promote` flag. The effect of this configuration is that a non-authenticating proxy will start only if `transparent mode` has already been enabled. (This behavior can be understood by noting that if `transparent` mode were not enabled, a non-authenticating proxy would attempt to

challenge.) As non-authenticating proxies do not support `challenge` mode, the proxy will terminate and drop the connection. `Transparent promote` must be set in order for the proxy not to challenge every connection attempt. To enable `transparent` mode, a user must first interact with Black Hole in `gateway` mode, and thus be forced to authenticate himself/herself.

Not all non-authenticating proxies will work in this manner (`mail` and DNS are two examples). Also, once `transparent` mode is enabled, no additional users from the enabling IP address need to be authenticated. **The evaluated configuration mandates that transparent mode only be used from single-user workstations.**

Related Security Objective:

O.IDENT

FIA_UAU.2 **Single-use Authentication Mechanisms**

Dependencies:

FIA_UID.1 Basic User Identification

FIA_ADA.1 User Authentication Data Initialization

Elements:

FIA_UAU.2.1 The TSF shall authenticate any user's claimed identity prior to performing any functions for the user.

This requirement applies only to *authenticating* proxies.

Authenticating proxies can be configured to require user authentication before any connection requests are allowed to complete.

Administrators should be aware that, while the use of non-authenticating proxies does not interfere with the functionality of the authenticating proxies, **non-authenticating proxies do not meet this requirement**. In addition, **authenticating proxies must be configured properly (challenge mode, and the S/Key password mechanism) in order to meet this requirement**.

It is possible, though cumbersome, to force users to authenticate even when using non-authenticating proxies. This can be accomplished by configuring the rules for non-authenticating proxies

to use `challenge` mode, and setting the `transparent promote` flag. The effect of this configuration is that a non-authenticating proxy will start only if `transparent` mode has already been enabled. (This behavior can be understood by noting that if `transparent` mode were not enabled, a non-authenticating proxy would attempt to challenge.) As non-authenticating proxies do not support `challenge` mode, the proxy will terminate, and drop the connection. `Transparent promote` must be set in order for the proxy not to challenge every connection attempt. To enable `transparent` mode, a user must first interact with Black Hole in `gateway` mode, and thus be forced to authenticate himself/herself.

Not all non-authenticating proxies will work in this manner (`mail` and `DNS` are two examples). Also, once `transparent` mode is enabled, no additional users from the enabling IP address need to be authenticated. **The evaluated configuration mandates that transparent mode only be used from single-user workstations.**

FIA_UAU.2.2 The TSF shall prevent reuse of authentication data related to ***passwords***.

Black Hole uses S/Key for one-time authentication. An administrator can require the use of S/Key for authentication, through appropriate configuration of the rules database. S/Key uses the MD5 one-way hash algorithm to generate a new password for each authentication attempt. Passwords from previous authentication attempts cannot be used for new authentication attempts.

Related security objective:

O.IDENT

FIA_UID.2 Unique Identification of Users

Dependencies:

FIA_ATD.2 Unique User Attribute Definition

Elements:

FIA_UID.2.1 The TSF shall uniquely identify each user before performing any actions requested by the user.

This requirement applies only to *authenticating* proxies.

Authenticating proxies can be configured to require user identification before any connection requests are allowed to complete. Note that when user identification is requested by Black Hole, the supplied identification will always be authenticated (that is, if Black Hole asks for a user name, it will also ask for a password), so identification as discussed here is in fact equivalent to I&A.

Administrators should be aware that, while the use of non-authenticating proxies does not interfere with the functionality of the authenticating proxies, **non-authenticating proxies do not meet this requirement**. In addition, **authenticating proxies must be configured properly (that is, to use challenge mode, and a password mechanism) in order to meet this requirement.**

It is possible, though cumbersome, to force users to identify themselves even when using non-authenticating proxies. This can be accomplished by configuring the rules for non-authenticating proxies to use `challenge` mode, and setting the `transparent promote` flag. The effect of this configuration is that a non-authenticating proxy will start only if `transparent` mode has already been enabled. (This behavior can be understood by noting that if `transparent` mode were not enabled, a non-authenticating proxy would attempt to challenge.) As non-authenticating proxies do not support `challenge` mode, the proxy will terminate, and drop the connection. `Transparent promote` must be set in order for the proxy not to challenge every connection attempt. To enable `transparent` mode, a user must first interact with Black Hole in `gateway` mode, and thus be forced to identify himself/herself.

Not all non-authenticating proxies will work in this manner (`mail` and `DNS` are two examples). Also, once `transparent` mode is enabled, no additional users from the enabling IP address need to be authenticated. **The evaluated configuration mandates that transparent mode only be used from single-user workstations.**

Related security objective:

O.IDENT

4.1.4 Class FPT Protection of the Trusted Security Functions

The components of this class address the means by which the TOE protects itself from unauthorized tampering and mechanism failure.

FPT_AMT.2 **Abstract Machine Testing During Start-Up**

Dependencies:

No dependencies

Elements:

FPT_AMT.2.1 The TSF shall run a suite of self tests during initial start-up, in order to demonstrate the correct operation of the functions provided by the TSF's underlying abstract machine.

The evaluation team interprets the abstract machine for Black Hole as being the underlying hardware platform (SPARCstation 5, 10, or 20). Correct operation of the abstract machine is validated by self-tests performed by the hardware upon system startup (the POST routine). The extent of the self-testing is not exhaustive, but appropriately covers the security-relevant functionality.

Related security objective:

O.PROTECT

FPT_REV.1 **Basic Revocation**

Dependencies:

No dependencies

Elements:

FPT_REV.1.1 The TSF shall provide a capability for revocation of security attributes associated with ***subjects and objects*** within the TSC.

The revocation of security attributes is interpreted as the ability of the TSF to take away privileges that have been granted. When a change is made to the rules database, the permissions for the subjects and objects associated with the updated rules are changed. Lost permissions will be reflected in a denial, the next time that access is requested.

FPT_REV.1.2 The TSF shall enforce revocation upon access attempt ***for which permissions have been revoked***.

When a change is made to the rules database, the permissions for the subjects and objects associated with the updated rules are

changed. Lost permissions will be reflected in a denial, the next time that access is requested.

Related security objective:

O.ACCESS

FPT_RVM.1 **Non-Bypassability of the TSP**

Dependencies:

No dependencies

Elements:

FPT_RVM.1.1 The TSF shall ensure that TSP enforcement functions are invoked and succeed before any security-related operation is allowed to proceed.

The environmental assumption that the firewall is the only node common to both the source network and the destination network ensures that no communication can flow around Black Hole. All communication through Black Hole, from the initial connection request to the data flow through an established connection, is performed by trusted code.

Communication to Black Hole (for purposes such as changing a password, or toggling `transparent` mode) requires that the user first be authenticated. Again, this is performed by trusted code.

Related security objective:

O.PROTECT

FPT_SEP.1 **TSF Domain Separation**

Dependencies:

No dependencies

Elements:

FPT_SEP.1.1 The TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

All code on Black Hole is trusted. The only interfaces to Black Hole are through the local administrative console, and through connected networks. The administrative console is protected by a password mechanism, and the machine itself is to be stored in a protected physical area. All network access to Black Hole is mediated by: Guardian; The Oracle; and the proxies.

The effect of: all code being trusted on Black Hole; the physical isolation of the machine; and the mediation of all possible access points to the TSF, is to achieve the appropriate isolation of the TSF.

FPT_SEP.1.2 The TSF shall enforce separation between the security domains of subjects in the TSC.

Remote hosts and processes are outside Black Hole's scope of control. Hence, Black Hole cannot enforce domain separation between remote hosts, or between remote processes; this must be accomplished by the environment. However, Black Hole **can** enforce separation at the network level, and also achieves separation **within** the bounds of the TSF. Each stream of network traffic can be considered as being associated with a remote subject. All network traffic passing though Black Hole is mediated by trusted code, which ensures the integrity of the individual streams.

This viewpoint is analogous with the fact that Black Hole forms only one part of the overall network architecture. Black Hole can only mediate based on the information provided by remote subjects, and can only enforce non-interference within its own bounds of control.

Related security objective:

O.PROTECT

FPT_TSA.2 Separate Security Administrative Role

Dependencies:

FIA_UID.1 Basic User Identification

FIA_ATD.1 User Attribute Definition

FIA_ATA.1 User Attribute Initialization

AGD_ADM.1 Administrator guidance

Elements:

FPT_TSA.2.1 The TSF shall distinguish security-relevant administrative functions from other functions.

Administration can be performed only from the local console (all remote administration is disabled as part of the evaluated configuration).

FPT_TSA.2.2 The TSF's set of security-relevant administrative functions shall include all functions necessary to install, configure, and manage the TSF; minimally, this set shall include the following administrative capabilities:

- ***startup and shutdown of the audit subsystem;***
- ***management of the audit trail;***
- ***provision of audit review tools for searching and sorting;***
- ***configuration of the access permissions of subjects to objects;***
- ***initialization and modification of user authentication data; and***
- ***initialization, display and modification of user attributes.***

The *Black Hole Administration Guide* [MIL 96a] describes the purposes and the correct uses of all security-relevant administrative functions, including: management of the audit subsystem; archiving the audit logs; use of the Postgres95 audit reduction tool; and maintaining the rules database. The *Black Hole Installation Guide* [MIL96c] details the procedures necessary to install Black Hole.

The GUI contains functions that allow an administrator to initialize or modify: user passwords; S/Key seed values; and other attributes, for any user within the rules database.

FPT_TSA.2.3 The TSF shall restrict the ability to perform security-relevant administrative functions to a security administrative role that has a specific set of authorized functions and responsibilities.

Administrators at the local console must log in with root privilege, using the standard UNIX login/password mechanism.

FPT_TSA.2.4 The TSF shall be capable of distinguishing the set of users authorized for administrative functions from the set of all users of the TOE.

Only administrators can have local accounts on Black Hole itself, and each administrator account must have a unique user ID. Network users do not have user accounts (they exist only as rules in The Oracle database).

FPT_TSA.2.5 The TSF shall allow only specifically authorized users to assume the security administrative role.

An administrator must have local access to the console, and must supply the correct login/password combination, in order to perform any administrative function.

FPT_TSA.2.6 The TSF shall require an explicit request to be made in order for an authorized user to assume the security administrative role.

An administrator must have local access to the console, and must supply the correct login/password combination, in order to perform any administrative function.

Related security objective:

O.ADMIN

FPT_TSM.1 Management Functions

Dependencies:

FPT_TSA.1 Basic Security Administration

Elements:

FPT_TSM.1.1 The TSF shall provide the authorized administrator with the ability to set and update the following configuration parameters:

- ***functions that enable and that specify audit recording and alarms;***
- ***functions that enable/disable protocol-based communications;***
- ***functions that enable/disable mail services;***
- ***functions that control audit record storage; and***
- ***functions that control and maintain the system configuration files.***

A security administrator can set all of the security parameters through the use of configuration files, which are modified through the administrator's GUI interface. Such configuration files include, for example, those that prescribe the behavior of the audit and alarm processes. Security administrators also define the services (including the enabling and disabling of each, and the rules database.

FPT_TSM.1.2 The TSF shall provide the authorized administrator with the ability to:

- ***perform the initial product installation;***
- ***perform backup and/or recovery of the firewall system; and***
- ***add or configure peripheral device, such as printers and drives.***

The *Black Hole Installation Guide (*with its included release notes) explains the installation procedure. The installation program guides the administrator through the process, querying for information when necessary.

The security administrator has full control over the firewall configuration, including its: operation; backup; recovery; and start-up/shut-down. (Black Hole's backup facility also allows backing up of the configuration files.)

The *Black Hole Administration Guide* explains the backup/recovery procedures, as well as the internal rules database recovery system (required in the event of an inelegant shut-down). The backup of configuration files provides system object backup; the backup of the policy database provides backup of network objects. The policy database is digitally signed using MD5 when initially backed up, in order to allow for the verification of integrity, if and when it is restored.

Peripheral devices (for example, tape or CD-ROM drives) are configured by the underlying UNIX kernel (that is, the kernel has the required drivers for the devices, as well as the associated programs and utilities to supply the human interface).

Related security objective:

O.ADMIN

4.2 IT Assurance Requirements

Each of the following assurance requirements includes a description of the evaluated firewall's assurance measures that are necessary to meet the requirement. While most of these are EAL3 assurance requirements, there are also EAL4-level requirements (ACM_CAP.3; ACM_SCP.2; and ALC_LCD.1) and EAL7-level requirements (ATE_IND.3) listed that are met by the product. This combination of all of the assurance requirements that are being met earns Black Hole an *EAL3-Augmented* rating. The assurance requirements that make up this rating are described in Table III below.

Table III - Security Assurance Requirements

Configuration Management	
ACM_CAP.3	Generation Support and Acceptance Procedures
ACM_SCP.2	Problem Tracking CM Coverage
Delivery and Operation	
ADO_IGS.1	Installation, Generation, and Start-up
Development	
ADV_FSP.1	TOE and Security Policy
ADV_HLD.2	Security Enforcing High-Level Design
ADV_RCR.1	Informal Correspondence Demonstration
Guidance Documents	
AGD_ADM.1	Administrator Guidance
AGD_USR.1	User Guidance
Life Cycle Support	
ALC_DVS.1	Identification of Security Measures
ALC_LCD.1	Developer Defined Life-Cycle Model
Testing	
ATE_COV.2	Complete Coverage - Rigorous
ATE_DPT.2	Testing - High Level Design
ATE_FUN.1	Functional Testing
ATE_IND.3	Independent Testing - Complete
Vulnerabilites	
AVA_MSU.1	Misuse Analysis - Obvious Flaws
AVA_SOF.1	Strength of TOE Security Function Evaluation
AVA_VLA.1	Developer Vulnerability Analysis

4.2.1 Class ACM Configuration Management

This class addresses the means of assuring that the functional requirements are realized in the TOE.

ACM_CAP.3 Generation support and acceptance procedures

Dependencies:

ACM_SCP.1 Minimal CM coverage

ALC_DVS.1 Identification of Security Measures

Developer action elements:

ACM_CAP.3.1D The developer shall use a CM system.

Milkyway Networks has a configuration management (CM) system in place for Black Hole and associated configuration items (CIs) (for example, supporting product documentation; test cases; and security flaws).

Milkyway Networks uses a freeware product, Concurrent Versioning System (CVS), which is based on the Revision Control System (RCS) freeware product. CVS is the tool used to support CM activities on the product, as well as aid the software development team's acceptance procedures. (For details, see *CVS User's Manual* [MIL97g].)

The supporting product documentation (such as *Black Hole Administration Guide,* and *Black Hole User Guide* [MIL96b]) are tracked by an online document-repository system that is administered by the Milkyway Networks technical writing department.

Product flaws (including security flaws) are tracked by the Distributed Defect Tracking System (DDTS).

The quality assurance (QA) test case suite for the product is run under a Microsoft application suite, that includes Microsoft Excel spreadsheets and Microsoft Access databases.

All of these distributed CM systems use the Black Hole product version as a common reference base.

ACM_CAP.3.2D The developer shall provide CM documentation.

The CM documentation includes: a configuration list; a CM plan; and acceptance procedures, which are all found in the *Configuration Management Plan* [MIL97e] and *Engineering Procedures* [MIL97f] documents.

Content and presentation of evidence elements:

ACM_CAP.3.1C The CM documentation shall include a configuration list, a CM plan, and an acceptance plan.

The CM documentation includes a configuration list, a CM plan, and acceptance procedures, which are all found in the *Configuration Management Plan* and *Engineering Procedures* documents.

ACM_CAP.3.2C The configuration list shall describe the configuration items that comprise the TOE.

The configuration list that is produced by CVS; the document repository system; and the QA test case system, together comprise the elements that make up the TOE. The CIs that are tracked are: the individual software modules of the product; the modules' associated MAKE files; the individual documents; the individual security flaws (as tracked by DDTS); and individual test cases.

ACM_CAP.3.3C The CM documentation shall describe the method used to uniquely identify the TOE configuration items.

The CVS manual describes how CIs are uniquely identified, created and tracked. CIs in the document repository system and in the QA test case system have unique tracking identifiers.

ACM_CAP.3.4C The CM plan shall describe how the CM system is used.

The CM plan, with support from the CVS manual, describes how the CM systems are used, and provides examples of the CM systems being used.

ACM_CAP.3.5C The CM documentation shall provide evidence that the CM system is working properly.

The CM plan (with support from the CVS manual) describes how the CM systems are used, and provides examples of the CM systems working properly.

ACM_CAP.3.6C The CM documentation shall provide evidence that all configuration items have been and are being effectively maintained under the CM system.

The CM plan (with support from the CVS manual) describes how the CM systems are used, and provides examples of all CIs being maintained.

ACM_CAP.3.7C The CM system shall ensure that only authorized changes are made to the TOE configuration items.

The CM system (in conjunction with the acceptance procedures practiced) ensures that only authorized changes are made to any of

the TOE elements, whether the elements are software modules, document contents, or test cases.

ACM_CAP.3.8C The CM system shall support the generation of all supported versions of the TOE.

The CM system allows for the generation of any previous version (including the compiler tools that were used).

ACM_CAP.3.9C The acceptance plan shall describe the procedures used to accept modified or newly created TSF configuration items as part of the TOE.

The acceptance test plan and CM practices allow the QA department to test and accept new or updated modules (CIs) into the CM system, and to incorporate them into the TOE.

Evaluator action elements:

ACM_CAP.3.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

The provided documents and examples were reviewed and accepted. Milkyway Networks has provided a complete listing of all CIs associated with the TOE. The evaluation team has determined that the granularity of CIs is acceptable (for a description, see ACM_CAP.3.2C).

ACM_SCP.2 Problem tracking CM coverage

Dependencies:

ACM_CAP.2 Authorization controls

Developer action elements:

ACM_SCP.2.1D The developer shall provide CM documentation.

The CM documentation and the CVS Manual describe how CIs are tracked by the: CVS; DDTS; document repository; and QA CM systems, which comprise the Milkyway Networks CM system.

Content and presentation of evidence elements:

ACM_SCP.2.1C As a minimum, the following shall be tracked by the CM system: the TOE implementation representation, design documentation, test documentation, user documentation, administrator documentation, CM documentation, and security flaws.

Milkyway Networks uses the CVS tool to track the TOE implementation representation (that is, the software product).

The document repository system administered by the Milkyway Networks technical writers tracks the TOE's: design documentation; user documentation; and administrator documentation, as well as the CM documentation.

Product flaws (including security flaws) are tracked by the DDTS tool.

The test documentation associated with the TOE is maintained by the Milkyway Networks QA department. All test cases are stored in Microsoft Access databases. Each test case has fields for: a unique testcase identifying number; the version of Black Hole to which the test case applies; the revision number of the test case; the author of the test case; the module to which the test case applies (GUI, guardian, etc.); and the relevant DDTS number (if the test case was generated in response to a bug report). There is no automated maintenance of the database. QA personnel make changes by hand, when necessary.

ACM_SCP.2.2C The CM documentation shall describe how configuration items are tracked by the CM system.

The CVS manual describes how CIs are uniquely identified, created and tracked. CIs in the document repository system and in the QA test case system have unique tracking identifiers associated with each CI.

Evaluator action elements:

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

The provided documents and examples were reviewed and accepted by the evaluation team.

4.2.2 Class ADO Delivery and Operation

This class provides requirements for the: delivery; installation; generation; and start-up of the TOE.

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

Dependencies:

AGD_ADM.1 Administrator guidance

Developer action elements:

ADO_IGS.1.1D The developer shall document procedures to be used for the secure installation, generation, and start-up of the TOE.

Milkyway Networks has produced the *Black Hole Installation Guide* and the *Black Hole Administration Guide,* in order to provide administrators with a complete, step-by-step approach to installing Black Hole.

Content and presentation of evidence elements:

ADO_IGS.1.1C The documentation shall describe the steps necessary for secure installation, generation, and start-up of the TOE.

Milkyway Networks has produced the *Black Hole Installation Guide* and the *Black Hole Administration Guide,* in order to provide administrators with a complete, step-by-step approach to installing Black Hole. A form is included (at the end of the installation guide), to help an administrator collect important information about a system prior to installation. A section on rules configuration explains the procedures for creating rules in each of the four action modes (**drop**, **disallow**, **challenge**, **allow**).

Evaluator action elements:

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

The provided documentation was reviewed and accepted by the evaluation team.

4.2.3 Class ADV Development

This class addresses the means of assuring, through graduated levels of abstraction, that the TOE's functional interface corresponds to its implementation.

ADV_FSP.1 TOE and security policy

Dependencies:

ASE_TSS.1 Security Target, TOE Summary Specification, Evaluation Requirements

ADV_RCR.1 Informal correspondence demonstration

Developer action elements:

ADV_FSP.1.1D The developer shall provide a functional specification.

Milkyway Networks has supplied the evaluation team with the *Black Hole Administration Guide*, the *Black Hole User Guide*, and the *Black Hole Functional Specifications* [MIL97b]. The totality of these documents provides the functional specification.

ADV_FSP.1.2D The developer shall provide a TSP.

The Black Hole TSP is included with the *Black Hole Functional Specification*.

Content and presentation of evidence elements:

ADV_FSP.1.1C The functional specification shall describe the TSF using an informal style.

The supplied Milkyway Networks documentation is expressed in a natural language (English), which qualifies as informal.

ADV_FSP.1.2C The functional specification shall include an informal presentation of syntax and semantics of all external TSF interfaces.

The functional specification documentation provides an informal presentation of all external TSF interfaces. This was verified by determining what the external interfaces are (through inspection), and verifying that the documentation supports those interfaces.

ADV_FSP.1.3C The functional specification shall include evidence that demonstrates that the TSF is completely represented.

The evaluation team verified that Black Hole was completely represented, by informally mapping all known Black Hole external interfaces to the information presented in the functional specification.

Evaluator action elements:

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

The provided documentation was reviewed and accepted by the evaluation team.

ADV_FSP.1.2E The evaluator shall determine that the functional specification is consistent with the TSP.

By informal inspection, the functional specification documentation was determined to be consistent with the TSP.

ADV_FSP.1.3E The evaluator shall determine if the functional requirements in the ST are addressed by the representation of the TSF.

The evaluation team determined that the representation of the TSF (in this case, the functional specification) addressed all of the requirements in the ST. This was accomplished by cross-checking each ST requirement against the functional specification documentation, and ensuring that the description(s) for each requirement demonstrated, at this level of design, that the requirement was satisfied.

ADV_HLD.2 Security enforcing high-level design

Dependencies:

ADV_FSP.1 TOE and security policy

ADV_RCR.1 Informal correspondence demonstration

Developer action elements:

ADV_HLD.2.1D The developer shall provide the high-level design of the TSF.

Milkyway Networks supplied the evaluation team with high-level design documentation for Black Hole, in the form of the *Black Hole Functional Specifications* and the *Black Hole Module and Interface Specifications* [MIL97c].

Content and presentation of evidence elements:

ADV_HLD.2.1C The presentation of the high-level design shall be informal.

The supplied Milkyway Networks documentation is expressed in a natural language (English), which qualifies as informal.

ADV_HLD.2.2C The high-level design shall describe the structure of the TSF in terms of subsystems.

The high-level design partitions Black Hole into appropriate subsystems, as determined by the evaluation team.

ADV_HLD.2.3C The high-level design shall describe the security functionality provided by each subsystem of the TSF.

Each identified subsystem is accompanied by a description of related security functionality.

ADV_HLD.2.4C The high-level design shall identify the interfaces of the subsystems of the TSF.

Each identified subsystem is described in terms of its interfaces with other Black Hole subsystems.

ADV_HLD.2.5C The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.

The evaluation team interpreted the purpose of this requirement as being the identification of the protection mechanisms, supplied by an underlying abstract machine, which would be used by Black Hole in enforcing both TSF self-protection, and subject non-interference properties. Traditionally for an operating system, such protection mechanisms are (for example) rings or segment limits, provided by the hardware and used by the OS to protect itself (and other processes) from damage by other processes. In the case of Black Hole, the boundary of trust is the network interface, and the issue of such protection mechanisms is therefore moot - all code on Black Hole is trusted. No underlying support for protection mechanisms is required for Black Hole to correctly implement its security functionality.

ADV_HLD.2.6C The high-level design shall describe the separation of the TSF into TSP enforcing and other subsystems.

The high-level design explicitly identifies the subsystems involved with TSP enforcement. This was verified by the evaluation team.

Evaluator action elements:

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

The provided documentation was reviewed and accepted by the evaluation team.

ADV_HLD.2.2E The evaluator shall determine if the functional requirements in the ST are addressed by the representation of the TSF.

The evaluation team determined that the representation of the TSF (in this case, the high-level design) addressed all the requirements in the ST. This was accomplished by cross-checking each ST requirement against the high-level design documentation, and ensuring that the description(s) for each requirement demonstrates, at this level of design, that the requirement is satisfied.

ADV_RCR.1 Informal correspondence demonstration

Dependencies:

No dependencies.

Developer action elements:

ADV_RCR.1.1D The developer shall provide evidence that the least abstract TSF representation provided is an accurate, consistent, and complete instantiation of the functional requirements expressed in the ST.

Milkyway Networks supplied representation correspondence documentation (in the form of a traceability table), from the ST requirement level, to the functional specification, to the high-level design.

Content and presentation of evidence elements:

ADV_RCR.1.1C For each adjacent pair of TSF representations, the evidence shall demonstrate that all parts of the more abstract representation are refined in the less abstract representation.

The evaluation team verified, by informal inspection, that each more abstract representation is a refinement of the parent representation (for example, the high-level design is a refinement of the functional specification).

ADV_RCR.1.2C For each adjacent pair of TSF representations, the demonstration of correspondence between the representations may be informal.

The supplied Milkyway Networks documentation is expressed in a natural language (English), which qualifies as informal.

Evaluator action elements:

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

The provided documentation was reviewed and accepted by the evaluation team.

ADV_RCR.1.2E The evaluator shall analyze the correspondence between the functional requirements expressed in the ST and the least abstract representation provided to ensure accuracy, consistency, and completeness.

The evaluation team used the traceability table to verify that the high-level design completely expresses the requirements of the ST.

4.2.4 Class AGD Guidance Documents

This class provides the requirements for user documentation, and for administrator documentation.

AGD_ADM.1 Administrator guidance

Dependencies:

ADV_FSP.1 TOE and security policy

Developer action elements:

AGD_ADM.1.1D The developer shall provide administrator guidance addressed to system administrative personnel.

Milkyway Networks has produced the *Black Hole Administration Guide*, which provides instructions and guidance for the secure administration of Black Hole.

Content and presentation of evidence elements:

AGD_ADM.1.1C The administrator guidance shall describe how to administer the TOE in a secure manner.

Milkyway Networks has produced the *Black Hole Administration Guide*, which provides instructions and guidance for the secure administration of Black Hole

AGD_ADM.1.2C The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

The *Black Hole Administration Guide* contains warnings about the functions and privileges that should be controlled in a secure processing environment.

AGD_ADM.1.3C The administrator guidance shall contain guidelines on the consistent and effective use of the security functions within the TSF.

The *Black Hole Administration Guide* contains guidelines on the consistent and effective use of the security functions within the TSF.

AGD_ADM.1.4C The administrator guidance shall describe the difference between two types of functions: those which allow an administrator to control security parameters, and those which allow the administrator to obtain information only.

The *Black Hole Administration Guide* contains guidance about each type of administrator function.

AGD_ADM.1.5C The administrator guidance shall describe all security parameters under the administrator's control.

The *Black Hole Administration Guide* contains the guidance necessary to describe all security parameters under administrator control.

AGD_ADM.1.6C The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

The *Black Hole Administration Guide* contains an appendix, that describes each audit event that relates to particular parts of the system.

AGD_ADM.1.7C The administrator guidance shall contain guidelines on how the security functions interact.

The *Black Hole Administration Guide* contains guidelines on how the security functions interact with each other.

AGD_ADM.1.8C The administrator guidance shall contain instructions regarding how to configure the TOE.

The *Black Hole Administration Guide* is divided into several chapters, describing:

- the policy database;
- secure installation;
- user maintenance;
- service maintenance;
- rules maintenance;
- configuration of applications;
- virtual private networking;
- auditing and alarms; and
- administrative tasks (such as backups and rolling of the log files).

Each of these chapters describes the functions that an administrator may use, and how to go about using them in a secure manner. The chapters on user maintenance and service maintenance describe how to define the user security attributes and the service security attributes, respectively. The chapter on rules maintenance describes how to implement the desired security policy.

AGD_ADM.1.9C The administrator guidance shall describe all configuration options that may be used during secure installation of the TOE.

The *Black Hole Administration Guide* describes all configuration options that may be used during secure installation of the TOE.

AGD_ADM.1.10C The administrator guidance shall describe details, sufficient for use, of procedures relevant to the administration of security.

The *Black Hole Administration Guide* is divided into several chapters, describing:

- the policy database;
- secure installation;
- user maintenance;
- service maintenance;
- rules maintenance;
- configuration of applications;
- virtual private networking;
- auditing and alarms; and
- administrative tasks (such as backups and rolling of the log files).

Each of these chapters describes the functions that the administrator may use, and how to go about using them in a secure manner. The chapters on user maintenance and service maintenance describe how to define the user security attributes and the service security attributes, respectively. The chapter on rules maintenance describes how to implement the desired security policy.

AGD_ADM.1.11C The administrator guidance shall be consistent with all other documents supplied for evaluation.

The *Black Hole Administration Guide* is consistent with all other documents supplied for evaluation.

Evaluator action elements:

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

The evaluation team verified that the *Black Hole Administration Guide* meets all of the requirements for content and presentation of evidence.

AGD_ADM.1.2E The evaluator shall confirm that the installation procedures result in a secure configuration.

The evaluation team verified that the *Black Hole Administration Guide* contains all of the guidance and instructions necessary for administration of the Black Hole in a secure manner.

AGD_USR.1 User guidance

Dependencies:

ADV_FSP.1 TOE and security policy

Developer action elements:

AGD_USR.1.1D The developer shall provide user guidance.

Milkyway Networks has produced the *Black Hole User Guide*, which provides instructions to the user on how to correctly and securely use Black Hole services and applications.

Content and presentation of evidence elements:

AGD_USR.1.1C The user guidance shall describe the TSF and interfaces available to the user.

The *Black Hole User Guide* includes sections on how to securely use the following services: World Wide Web (WWW); Telnet; FTP; Gopher; News; Archie; Wide Area Information Service (WAIS), and X-Windows. There is also a section on how to correctly make use of `transparent` mode.

AGD_USR.1.2C The user guidance shall contain guidelines on the use of security functions provided by the TOE.

The *Black Hole User Guide* provides guidance on the secure use of the product's security-relevant functions.

AGD_USR.1.3C The user guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

The *Black Hole User Guide* provides warnings about functions and privileges that should be controlled in a secure processing environment.

AGD_USR.1.4C The user guidance shall describe the interaction between user-visible security functions.

The *Black Hole User Guide* describes the interaction between user-visible security functions.

AGD_USR.1.5C The user guidance shall be consistent with all other documentation delivered for evaluation.

The *Black Hole User Guide* is consistent with all other documentation delivered for this evaluation.

Evaluator action elements:

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

The evaluation team verified that the *Black Hole User Guide* contains all of the guidance and instructions necessary to use Black Hole in a secure manner.

4.2.5 Class ALC Life-cycle Support

This class addresses control in changes to the TOE during its development and maintenance.

ALC_DVS.1 Identification of security measures

Dependencies:

No dependencies.

Developer action elements:

ALC_DVS.1.1D The developer shall produce development security documentation.

The Milkyway Networks document, *Security Policy and Procedures* [MIL97a], provides information on how Milkyway Networks provides the physical, personnel, and other security measures that are used to protect the confidentiality and integrity of the TOE during its development.

Content and presentation of evidence elements:

ALC_DVS.1.1C The development security documentation shall describe the physical, procedural, personnel, and other security measures that are used to protect the confidentiality and integrity of the TOE during its development.

The Milkyway Networks document, *Security Policy and Procedures,* provides information on how Milkyway Networks provides the physical, personnel, and other security measures that are used to protect the confidentiality and integrity of the TOE during its development.

ALC_DVS.1.2C The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.

An example access control log is included in the documentation, showing the access granted to Milkyway Networks personnel as they move from one physical security zone to another within the Milkyway Networks office complex.

Evaluator action elements:

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

The evaluation team verified that the *Security Policy and Procedures* document contains all of the required content and presentation elements.

ALC_DVS.1.2E The evaluator shall check whether the security measures are being applied.

The security measures documented in the *Security Policy and Procedures* document are being applied at the Milkyway Networks development site in Ottawa. This has been evident during: the Trusted Product Evaluation Program (TPEP) Preliminary Technical Review; TPEP Advice & Guidance meetings; and during the Design Analysis Phase, both in on-site meetings, and in conversations with Milkyway Networks staff.

ALC_LCD.1 Developer defined life-cycle model

Dependencies:

No dependencies.

Developer action elements:

ALC_LCD.1.1D The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.

Milkyway Networks has developed and documented a software development life-cycle model, which is described in the company's *Engineering Procedure* [MIL97f] document.

ALC_LCD.1.2D The developer shall produce life-cycle definition documentation.

Milkyway Networks has developed and documented a software development life-cycle model, which is described in the *Engineering Procedures* document.

Content and presentation of evidence elements:

ALC_LCD.1.1C The life-cycle definition documentation shall describe the model used to develop and maintain the TOE.

Milkyway Networks has developed and documented a software development life-cycle model, which is described in the *Engineering Procedures* document. This document provides guidance to Milkyway Networks software developers, on the life-cycle model used through the: conceptual; developmental; testing; and launch phases of the product.

Evaluator action elements:

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

The life-cycle model is what would be expected of a medium-sized company such as Milkyway Networks. This model is linked closely with the company's CM plan. The procedures and steps have been demonstrated to be in use, by the implementation and use of the CM systems.

4.2.6 Class ATE Testing

This class addresses the means of assuring that the TOE satisfies at least the security functional requirements of the ST.

ATE_COV.2 Complete coverage - rigorous

Dependencies:

ADV_FSP.1 TOE and security policy

ATE_FUN.1 Functional testing

Developer action elements:

ATE_COV.2.1D The developer shall provide an analysis of the test coverage.

Milkyway Networks has provided an analysis of the test coverage.

Content and presentation of evidence elements:

ATE_COV.2.1C The analysis of the test coverage shall demonstrate that the tests identified in the test documentation cover the TSF.

Milkyway Networks has provided a table, mapping each ST security functional element to test cases identified in the supplied documentation.

ATE_COV.2.2C The analysis of the test coverage shall demonstrate the correspondence between the security functions and the tests identified in the test documentation.

Milkyway Networks has provided a table, mapping each ST security functional element to test cases identified in the supplied documentation.

Evaluator action elements:

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

The evaluation team ensured that each ST functional element was exercised by at least one vendor test, and that the tests mapped to each requirement were sufficient to demonstrate correct functioning. The only exception to this was object reuse, which was verified through design analysis.

ATE_DPT.2 Testing - high level design

Dependencies:

ADV_FSP.1 TOE and security policy

ADV_HLD.1 Descriptive high-level design

ATE_FUN.1 Functional testing

Developer action elements:

ATE_DPT.2.1D The developer shall provide the analysis of the depth of testing.

Milkyway Networks has provided an analysis of the depth of testing.

Content and presentation of evidence elements:

ATE_DPT.2.1C The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TOE operates in accordance with the functional specification, and high level design of the TSF.

The functional specification and high-level design were each mapped to the test cases. The external interfaces of the functional specification and the subsystem interfaces of the high-level design were shown to be sufficiently exercised and in accordance with the test cases.

Evaluator action elements:

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

The evaluation team verified that the tests corresponding to each interface (external or subsystem level) were sufficient to demonstrate correct functioning. Each interface was tested in every manner in which it could be used, as indicated by the supplied design documentation.

ATE_FUN.1 **Functional testing**

Dependencies:

ATE_COV.1 Complete coverage - informal

ATE_DPT.1 Testing - functional specification

Developer action elements:

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

Milkyway Networks performed testing and generated documentation in the form of test cases. The test cases describe test procedures, as well as the expected and actual results.

ATE_FUN.1.2D The developer shall provide test documentation.

Milkyway Networks supplied documentation in the form of a test plan, test cases, and a table mapping functions to test cases.

Content and presentation of evidence elements:

ATE_FUN.1.1C The test documentation shall consist of test plans, test procedure descriptions, and test results.

Milkyway Networks has presented a test plan and test cases. Each test case includes the functionality being tested. It also includes: the setup; the procedure; the expected results; and the actual results.

ATE_FUN.1.2C The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.

The test plan contains the testing philosophy and procedures. It also maps each test case to Black Hole's: security functions; functional specifications; and high-level design. Testing is broken up into stages, which have clearly defined goals and criteria. The stages are:

- **Functional level testing.** Programs and libraries are tested by the programmers, with the goal of having each individual program working correctly on its own before Alpha testing.
- **Alpha testing.** System integration testing is done by the QA department, in a special test lab. All components must work together correctly, and the product must pass all tests, before Beta testing may begin.

- **Beta testing.** Testing of the "finished" product is performed by selected end-users of the firewall. The goal is to find any bugs in the product that have been missed during Alpha testing, but that would be detected during "real world" use.

ATE_FUN.1.3C The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function.

Each test case includes the functionality being tested. It also includes: the setup; the procedure; the expected results; and the actual results.

ATE_FUN.1.4C The test results in the test documentation shall show the expected results of each test.

Each test case includes the expected results, and the actual results.

ATE_FUN.1.5C The test results from the developer execution of the tests shall demonstrate that each security function operates as specified.

Each test case includes the expected results, and the actual results for confirmation.

Evaluator action elements:

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

The evaluation team examined the documentation, and performed all of the vendor's security-relevant tests for verification.

ATE_IND.3 Independent testing - complete

Dependencies:

ADV_FSP.1 TOE and security policy

AGD_USR.1 User guidance

AGD_ADM.1 Administrator guidance

ATE_FUN.1 Functional testing

Developer action elements:

ATE_IND.3.1D The developer shall provide the TOE for testing.

Milkyway Networks provided the installation CD of version 3.01E2 of Black Hole, as well as its accompanying documentation (installation, user, and administrative guidance). Milkyway Networks also verified

the team's pre-installation checklist, and supplied the necessary product-activation codes.

Content and presentation of evidence elements:

ATE_IND.3.1C The TOE shall be suitable for testing.

The TOE was a full working version, which matched the evaluated configuration.

Evaluator action elements:

ATE_IND.3.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

The evaluation team examined the documentation, and found that it met the requirements.

ATE_IND.3.2E The evaluator shall test the TSF to confirm that the TSF operates as specified.

The evaluation team performed all of the vendor's tests (for verification), and performed some additional tests which had been created by the team. The TSF operates as specified.

ATE_IND.3.3E The evaluator shall execute all tests in the test documentation to verify the developer test results.

The evaluation team performed all of the vendor's security-relevant tests, for verification.

4.2.7 Class AVA Vulnerability Assessment

This class addresses weaknesses resulting from: the existence of covert channels; improper configuration of the TOE; or penetration attacks.

AVA_MSU.1 Misuse analysis - obvious flaws

Dependencies:

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

AGD_ADM.1 Administrator guidance

AGD_USR.1 User guidance

Developer action elements:

AVA_MSU.1.1D The developer shall document an analysis of the guidance documentation for conflicting and incomplete guidance.

Milkyway Networks has produced a *Vulnerability Assessment* document [MIL97d], which describes the process that the development team took to verify the correctness and the validity of the guidance documentation.

On several occasions, the guidance documentation (administrative, user, and installation) was reviewed by senior people from Milkyway Networks support staff (other than the original writers), in order to ensure that no obvious mistakes existed.

AVA_MSU.1.2D The developer shall ensure that the guidance documentation contains no misleading or unreasonable guidance.

On several occasions, the guidance documentation (administrative, user, and installation) was reviewed by senior people from Milkyway Networks support staff (other than the original writers), in order to ensure that no obvious mistakes existed.

Content and presentation of evidence element:

AVA_MSU.1.1C The analysis documentation shall provide a rationale that demonstrates that the guidance is not conflicting and is complete.

The *Vulnerability Assessment* document provides an analysis that shows that the guidance documentation is not conflicting, and is complete.

Evaluator action elements:

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

The evaluation team has confirmed that the information provided meets all requirements for content and presentation of evidence.

AVA_MSU.1.2E The evaluator shall determine that there is no misleading or unreasonable guidance in the guidance documentation.

The evaluation team verified the procedures outlined in the guidance documentation, in order to ensure that there is no misleading or unreasonable guidance within that documentation.

AVA_MSU.1.3E The evaluator shall repeat any procedures in the guidance documentation to ensure that they produce the documented results.

The evaluation team repeated the procedures outlined in the guidance documentation, in order to ensure that the procedures produce the documented results.

AVA_SOF.1 Strength of TOE security function evaluation

Dependencies:

ADV_FSP.1 TOE and security policy

ADV_HLD.1 Descriptive high-level design

Developer action elements:

AVA_SOF.1.1D The developer shall identify all TOE security mechanisms for which a strength of TOE security function analysis is appropriate.

Milkyway Networks has identified all security mechanisms for which a strength of TOE security function analysis is appropriate. The password mechanisms for UNIX and S/Key were identified as such.

AVA_SOF.1.2D The developer shall perform a strength of TOE security function analysis for each identified mechanism.

Milkyway Networks performed a TOE security function analysis for each identified mechanism (UNIX-like and S/Key passwords). For UNIX-like passwords, the analysis consisted of a discussion on the amount of time required to guess a password of varying length (for example, at 1 guess per second, a 16 bit password might be found in a day). For S/Key, the analysis consisted of a comparison, of the strength of the MD5 hashing algorithm (upon which S/Key is based) versus the standard UNIX password mechanism. UNIX-like passwords were found to be of *basic* strength. S/Key passwords were found to be of *medium* strength.

Content and presentation of evidence elements:

AVA_SOF.1.1C The strength of TOE security function analysis shall determine the impact of the identified TOE security mechanisms on the ability of the TOE security functions to counter the threats.

The analysis of the security functions determined that the impact of the basic/medium classification for reusable/one-time passwords (respectively) is minimal, in terms of Black Hole being able to counter the specified threats in the identified environment.

AVA_SOF.1.2C The strength of TOE security function analysis shall demonstrate that the identified strength of the security functions is consistent with the security objectives of the TOE.

The evaluation team verified that the strength of the identified security functions (basic and medium) is consistent with the target environment (threats, policies, assumptions and objectives).

AVA_SOF.1.3C Each strength claim shall be either basic, medium, or high.

Milkyway Networks has documented its analysis by using a "basic, medium or high" notation. The results of the developer's TOE security function analysis are:

- UNIX: basic to medium strength; and
- S/key: medium strength.

Evaluator action elements:

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

The evaluation team confirmed that the information provided meets all requirements for content, and for the presentation of evidence.

AVA_SOF.1.2E The evaluator shall confirm that all TOE security mechanisms requiring a strength analysis have been identified.

The evaluation team confirmed that the password mechanisms were the only TOE security mechanisms that required strength analyses.

AVA_SOF.1.3E The evaluator shall confirm that the strength claims are correct.

The evaluation team confirmed that the strength claims identified by the vendor are correct.

AVA_VLA.1 Developer vulnerability analysis

Dependencies:

ADV_FSP.1 TOE and security policy

ADV_HLD.1 Descriptive high-level design

AGD_ADM.1 Administrator guidance

AGD_USR.1 User guidance

Developer action elements:

AVA_VLA.1.1D The developer shall perform and document an analysis of the TOE deliverables searching for obvious ways in which a user can violate the TSP.

Milkyway Networks has performed and documented a vulnerability analysis [MIL97d] of the TOE deliverables. This analysis is based on the National Computer Security Association (NCSA) certification of Black Hole [NCS97a] against the following categories of vulnerabilities (all of which are tested for by the `Internet Security Scanner` [ISS]):

- well-known vulnerabilities in the following applications: `rlogin; rsh; HTTP; X-Windows; NFS; sendmail; FTP; TFTP; and finger;`
- IP spoofing;
- TCP sequence number prediction;
- source routing; and
- denial-of-service attacks (for example, repeated sending of packets with malformed flags, UDP bombing).

For more specific technical information, readers should see the security library on UNIX system vulnerabilities (found at [ISS97a]), or contact Milkyway Networks.

AVA_VLA.1.2D The developer shall document the disposition of identified vulnerabilities.

The vulnerability assessment documentation lists each vulnerability, and whether or not Black Hole is vulnerable.

Content and presentation of evidence elements:

AVA_VLA.1.1C The evidence shall show, for each vulnerability, that the vulnerability cannot be exploited in the intended environment for the TOE.

The vulnerability assessment documentation shows that the listed vulnerabilities are not exploitable in the intended environment for Black Hole.

Evaluator action elements:

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

The evaluation team confirmed that the information provided meets all requirements for content, and for the presentation of evidence.

AVA_VLA.1.2E The evaluator shall conduct penetration testing, based on the developer vulnerability analysis, to ensure obvious vulnerabilities have been addressed.

The evaluation team conducted penetration testing based on the developer's vulnerability analysis, in order to ensure that obvious vulnerabilities had been addressed. The evaluation team repeated the NCSA ISS test suite, and performed the following additional testing:

- testing of the disabled ports (6000, 2000 and 514) for `X-Windows` and `syslogd`, to ensure that they could not be connected to;
- sending oversized PING packets to Black Hole;
- SYN flooding;
- verifying that VPN traffic is not plaintext, and that VPN traffic stalls if the tunnel is shut down (that is, open connections will hang);
- attempting connections to the `Netacl` port;
- running `SATAN` and `strobe` against Black Hole; and
- testing the behavior of Black Hole when probing the `FTP` data and DNS return ports.

The testing verified that these vulnerabilities were addressed.

5. Rationale

This section presents the rationale which demonstrates: that this ST states a complete and cohesive set of requirements; that Black Hole constitutes an effective set of IT security countermeasures within the stated security environment; and that the summary specification addresses the requirements.

5.1 Security Objectives Rationale

This section shows the mapping between threats, objectives and assumptions (identified in Chapter 2, Security Environment). Specifically, each threat is shown to be countered by at least one objective, or else is accepted as a potential system risk. Each objective is shown to be necessary in order to counter at least one threat.

5.1.1 Threats Countered by the TOE

T.ACORRUPT

O.PROTECT counters this threat, as it requires that the firewall protect the audit trail from tampering, destruction or modification. O.AUDIT counters this threat by requiring that the firewall suspend the occurrence of auditable events in case of audit storage exhaustion or audit system failure.

T.DCORRUPT

O.PROTECT counters this threat by requiring that the firewall protect from tampering, destruction or modification, all firewall: data structures; configuration information; and executables responsible for enforcing the firewall security policy. This objective also requires a separate domain of execution for the firewall.

T.PROBE

O.AUDIT counters this threat by requiring that the firewall create an audit trail for all security-relevant events.

T.REPLAY

O.IDENT counters this threat by requiring that the firewall have the capability to perform one-time authentication of users (but not administrators).

T.SPOOF

O.ACCESS counters this threat by ensuring that all accesses between subjects and objects be mediated by the firewall. The firewall can therefore examine the security attributes (including the subnet and the IP address) of external entities, and thus determine if a packet is being spoofed.

T.SYSACC

O.IDENT counters this threat by requiring that each administrator be uniquely identified and authenticated. O.ADMIN counters this threat by requiring separate roles for users and administrators. O.LOCAL counters this threat by allowing administration from the local console only.

T.SACCESS

O.ACCESS counters this threat by ensuring that all accesses between subjects and objects be mediated by the firewall. O.ADMIN counters this threat by providing the administrative capability to configure access control rules.

T.FLAW

O.ASSURE counters this threat by requiring that the firewall have sufficient assurance to ensure that the probability of implementation flaws being exploited is minimized. The level of assurance for this ST is EAL3-Augmented.

Table IV - Mapping of IT Objectives vs. Threats

Threat	**Associated Objectives**
T.ACORRUPT	O.PROTECT O.AUDIT
T.DCORRUPT	O.PROTECT
T.PROBE	O.AUDIT
T.REPLAY	O.IDENT
T.SPOOF	O.ACCESS
T.SYSACC	O.IDENT O.ADMIN O.LOCAL
T.SACCESS	O.ACCESS O.ADMIN
T.FLAW	O.ASSURE

5.1.2 Threats not countered by the TOE

T.DENIAL

This threat is accepted as a potential system risk.

T.ABUSE

This threat is partially countered by O.TRAIN and O.MANAGE, but is still a potential system risk.

T.ACCIDENT

This threat is partially countered by O.TRAIN and O.MANAGE, but is still a potential system risk.

T.SNIFF

This threat is partially countered by O.LOCAL and O.MANAGE, but is still a potential system risk.

T.SESSION

This threat is accepted as a potential system risk.

T.TROJAN

This threat is partially countered by O.TRAIN and O.MANAGE, but is still a potential system risk.

T.ADMIN

This threat is partially countered by O.TRAIN, O.MANAGE and O.REVIEW, but is still a potential system risk.

T.PHYSICAL

This threat is countered by O.PHYSICAL and O.CONNECT.

Table V - Mapping of Non-IT Objectives vs. Threats

Threat	Objective
T.DENIAL	---
T.ABUSE	O.TRAIN O.MANAGE
T.ACCIDENT	O.TRAIN O.MANAGE
T.SNIFF	O.LOCAL O.MANAGE
T.SESSION	---
T.TROJAN	O.TRAIN O.MANAGE
T.ADMIN	O.TRAIN O.MANAGE O.REVIEW
T.PHYSICAL	O.PHYSICAL O.CONNECT

5.1.3 Completeness of the Objectives

Table VI shows the mapping of security objectives to threats. It shows that the ST includes no unnecessary objectives, since each of the objectives counters at least one threat.

Table VI - Completeness of Objectives

Objective	Threat
O.ACCESS	T.SPOOF T.SACCESS
O.ADMIN	T.SYSACC T.SACCESS
O.AUDIT	T.ACORRUPT T.PROBE
O.IDENT	T.REPLAY T.SYSACC

O.PROTECT	T.ACORRUPT T.DCORRUPT
O.ASSURE	T.FLAW
O.MANAGE	T.ABUSE T.ACCIDENT T.SNIFF T.TROJAN T.ADMIN
O.PHYSICAL	T.PHYSICAL
O.CONNECT	T.PHYSICAL
O.LOCAL	T.SNIFF T.SYSACC
O.TRAIN	T.ABUSE T.ACCIDENT T.TROJAN T.ADMIN
O.REVIEW	T.ADMIN

5.2 Assumptions Rationale

This section provides mapping between the secure usage assumptions, and the non-IT security objectives. The purpose is to justify the selection of each assumption.

A.COMMS

This assumption partially supports the non-IT security objective O.MANAGE.

A.SECURE

This assumption supports the non-IT security objective O.PHYSICAL.

A.CUSTOMIZE

This assumption partially supports the non-IT security objective O.MANAGE.

A.LOCAL

This assumption supports the non-IT security objective O.LOCAL.

A.NOEVIL

This assumption supports the non-IT security objectives O.MANAGE, O.REVIEW and O.TRAIN.

A.SINGL_PT

This assumption supports the non-IT security objective O.CONNECT.

A.CASCADE

This assumption supports the non-IT security objective O.MANAGE.

Table VII - Mapping of Assumptions vs. Non-IT Security Objectives

Assumption	**Non-IT Objective**
A.COMMS	O.MANAGE
A.SECURE	O.PHYSICAL
A.CUSTOMIZE	O.MANAGE
A.LOCAL	O.LOCAL
A.NOEVIL	O.MANAGE O.REVIEW O.TRAIN
A.SINGL_PT	O.CONNECT
A.CASCADE	O.MANAGE

5.3 Security Requirements Rationale

This section shows: that the selected CC requirements support the defined objectives; that all dependencies of all requirements have been satisfied; and that the assurance level (with any augmentations) has been appropriately selected.

5.3.1 Mapping from CC requirements to IT security objectives

Table VIII - CC Requirements

Component	**Name**
FAU_GEN.1	Audit Data Generation
FAU_MGT.1	Audit Trail Management
FAU_POP.1	Human Understandable Format
FAU_PRO.1	Restricted Audit Trail Access
FAU_SAR.1	Restricted Audit Review

FAU_SAR.3	Selectable Audit Review
FAU_STG.3	Prevention of Audit Data Loss
FDP_ACC.2	Complete Object Access Control
FDP_ACF.2	Multiple Security Attribute Access Control
FDP_ACF.4	Access Authorisation and Denial
FDP_RIP.3	Full Residual Information Protection on Allocation
FDP_SAM.1	Administrator Attribute Modification
FDP_SAQ.1	Administrator Attribute Query
FIA_ADA.3	Extended User Authentication Data Administration
FIA_ADP.1	Basic User Authentication Data Protection
FIA_ATA.1	User Attribute Initialisation
FIA_ATA.2	Basic User Attribute Administration
FIA_ATD.2	Unique User Attribute Definition
FIA_UAU.2	Single-use Authentication Mechanisms
FIA_UID.2	Unique Identification of Users
FPT_AMT.2	Abstract Machine Testing During Start-Up
FPT_REV.1	Basic Revocation
FPT_RVM.1	Non-Bypassability of the TSP
FPT_SEP.1	TSF Domain Separation
FPT_TSA.2	Separate Security Administrative Role
FPT_TSM.1	Management Functions

FAU_GEN.1 Audit Data Generation

This component is included to support O.AUDIT, as it specifies the particular types of events that the firewall should audit, and also the minimum amount of information that is to be included in the audit records.

FAU_MGT.1 Audit Trail Management

This component is included to directly support O.ADMIN, as it ensures that the audit trail will be manageable by authorized administration personnel.

FAU_POP.1 Human Understandable Format

This component is included to support O.AUDIT, as it requires that the audit logs be human-readable, and understandable to administrative personnel.

FAU_PRO.1 Restricted Audit Trail Access

This component is included to support O.PROTECT, by protecting the audit trail from unauthorized modification.

FAU_SAR.1 Restricted Audit Review

This component supports O.AUDIT, as it specifies that audit review tools are to be provided.

FAU_SAR.3 Selectable Audit Review

This component is included to support O.AUDIT, as it requires that audit review tools be capable of using multiple criteria to select audit data for review. This reduces the time required to review the audit data, yet enables the administrator to search for security-relevant events.

FAU_STG.3 Prevention of Audit Data Loss

This component is included to support O.AUDIT, by preventing the loss of audit data through storage exhaustion or audit system failure.

FDP_ACC.2 Complete Object Access Control

This component is included to support O.ACCESS, as it requires that the firewall control all operations on all subjects and objects in the firewall.

FDP_ACF.2 Multiple Security Attribute Access Control

This component is included to support O.ACCESS, as it enforces access control based on multiple security attributes.

FDP_ACF.4 Access Authorization and Denial

This component is included to support O.ACCESS, as it provides the ability to explicitly grant or deny access.

FDP_RIP.3 Full Residual Information Protection on Allocation

This component is included to support O.ACCESS, by ensuring that no residual information remains when a user's access is revoked.

FDP_SAM.1 Administrator Attribute Modification

This supports O.ADMIN, by allowing an administrator to modify security attributes.

FDP_SAQ.1 Administrator Attribute Query

This supports O.ADMIN, by allowing authorized administrators to query security attributes.

FIA_ADA.3 Extended User Authentication Data Administration

This supports O.IDENT, by allowing authorized administrators to initialize or modify user authentication data.

FIA_ADP.1 Basic User Authentication Data Protection

This component is included to support O.PROTECT, as it provides protection of the authentication data that is permanently stored in the firewall

FIA_ATA.1 User Attribute Initialization

This component supports O.ADMIN, by ensuring that the ability to initialize user attributes is provided.

FIA_ATA.2 Basic User Attribute Administration

This component supports O.ADMIN, by ensuring that the ability to display or modify user attributes is provided.

FIA_ATD.2 Unique User Attribute Definition

This component supports O.ACCESS, as it requires that user security attributes be uniquely associated with each user.

FIA_UAU.1 Basic User Authentication

This component supports the O.IDENT objective, by requiring reusable passwords.

FIA_UAU.2 Single-use Authentication Mechanisms

This component supports the O.IDENT objective, by requiring one-time authentication for users.

FIA_UID.2 Unique Identification of Users

This component supports O.IDENT, by requiring the unique identification of each user.

FPT_AMT.2 Abstract Machine Testing During Start-Up

This component is included to support O.PROTECT, as it provides for TSF-invoked tests during start-up in order to ensure the correct operation of the firewall.

FPT_REV.1 Basic Revocation

This supports O.ACCESS, by revoking security attributes when a user session ends.

FPT_RVM.1 Non-Bypassability of the TSP

This supports O.PROTECT, by ensuring that the security functions cannot be circumvented.

FPT_SEP.1 TSF Domain Separation

This supports O.PROTECT, by preventing tampering with the internal structures and the security-enforcing mechanisms of the firewall.

FPT_TSA.2 Separate Security Administrative Role

This component is included to support O.ADMIN, as it requires that sufficient functions be provided to securely manage the firewall, and that these functions be restricted to authorized administrators. It provides a means to administer the firewall.

FPT_TSM.1 Management Functions

This component is included to support O.ADMIN, as it specifies the management functions required in order for the firewall to be properly and securely administered.

Table IX shows the relationship between the included functional requirements, and the objectives that they are intended to satisfy. Every IT security objective is shown to be met by at least one functional requirement (Note: O.ASSURE is addressed in the next section).

Table IX - Mapping Objectives to Functional Requirements

Security Objective	Functional Requirements
O.IDENT	FIA_UAU.1, FIA_UAU.2, FIA_UID.2,
O.ACCESS	FDP_ACC.2, FDP_ACF.2, FDP_ACF.4, FDP_RIP.3, FIA_ATD.2, FPT_REV.1
O.ADMIN	FPT_TSA.2, FPT_TSM.1, FAU_MGT.1 FDP_SAM.1, FDP_SAQ.1, FIA_ADA.3, FIA_ATA.1, FIA_ATA.2
O.PROTECT	FIA_ADP.1, FPT_AMT.2, FPT_SEP.1, FPT_RVM.1, FAU_PRO.1
O.AUDIT	FAU_GEN.1, FAU_POP.1, FAU_SAR.1, FAU_SAR.3, FAU_STG.3

5.3.2 Rationale for EAL3-Augmented Assurance

Table X describes the relationship between the evaluation assurance levels, and the assurance classes, families and components. Each assurance class and family is listed along the left side of the table, and each evaluation assurance level is listed along the top. The numbers in the boxes represent the component level that each family achieves for a given assurance level. The shaded boxes represent the assurance requirements which are listed in this profile.

Table X - Assurance Requirements

Assurance Class	**Assurance Family**	**Assurance Components by Evaluation Assurance Level**						
		EAL1	**EAL2**	**EAL3**	**EAL4**	**EAL5**	**EAL6**	**EAL7**
ConfigurationManagement	ACM_AUT				1	1	2	2
	ACM_CAP	1	1	2	3	3	4	4
	ACM_SCP			1	2	3	3	3
Delivery and Operation	ADO_DEL							
	ADO_IGS		1	1	1	1	1	1
Development	ADV_FSP	1	1	1	2	4	5	6
	ADV_HLD		1	2	2	3	4	5
	ADV_IMP				1	2	3	3
	ADV_INT					1	2	3
	ADV_LLD				1	1	2	2
	ADV_RCR	1	1	1	1	2	2	3
Guidance Documents	AGD_ADM	1	1	1	1	1	1	1
	AGD_USR	1	1	1	1	1	1	1
Life-cycle Support	ALC_DVS			1	1	1	2	2
	ALC_FLR							
	ALC_LCD				1	2	2	3
	ALC_TAT				1	2	3	3
Tests	ATE_COV		1	2	2	2	3	3
	ATE_DPT		1	2	2	3	3	4
	ATE_FUN		1	1	1	1	1	1
	ATE_IND	1	1	2	2	2	2	3
Vulnerability Assessment	AVA_CCA					1	2	2
	AVA_MSU			1	2	2	2	2
	AVA_SOF		1	1	1	1	1	1

	AVA_VLA		1	1	2	3	4	4

EAL3 Methodically Tested and Checked

CSE personnel believe that EAL3 represents a minimum acceptable level of assurance for firewalls used in most government environments. EAL3 assurance can be met by vendors practicing quality software engineering and development methodologies.

ACM_CAP.3 Generation support and acceptance procedures

The differences between ACM_CAP.2 and ACM_CAP.3 are that the latter requires an acceptance plan, and requires that the CM system support the generation of all supported versions of the TOE.

The added assurance acquired by having an acceptance plan in place is not only good product development practice, but a requirement for the type of firewall product envisioned for this environment. Vendors should have the capability to produce not only the current version of the product under development, but the historical, or other versions, as well.

Both of these requirements are met at the ACM_CAP.3 level.

ACM_SCP.2 Problem tracking CM coverage

The only difference between the EAL3 requirement of ACM_SCP.1 and the augmentation required for ACM_SCP.2 is the latter's inclusion of security flaws in the list of configuration items. CSE personnel believe that for products such as firewalls, whose sole purpose is the prevention of unauthorized access to networks, all discovered means of circumventing this protection must be tracked by the developer. The tracking of discovered security flaws, along with the product changes resulting from their discovery (see ACM_CAP.3), provides a linking of the two, thereby achieving the assurance required of a product for such an intended environment.

ALC_LCD.1 Developer defined life-cycle model

The procedures used in the development and maintenance of a firewall are inextricably linked to the tracking of the product's revisions. Consideration of its development procedures provides additional assurance that the elimination of a discovered flaw does not result in the introduction of a new flaw. It is therefore imperative that a description of the model used for life-cycle development be included in the assurance documentation.

ATE_IND.3 Independent testing - complete

The difference between ATE_IND.2 and ATE_IND.3 is the latter's requirement for running all of the tests in the test documentation, rather than merely "a subset of

tests," as is required by the former. Evaluators must perform testing of all security-critical functions (that is, those functions that enforce one or more security policies). Security-relevant functions (that is, those that do not enforce a security policy, but must operate correctly because security-relevant functions rely upon them) need not be completely tested; a subset will suffice. This approach is consistent with past and current TPEP practice.

Because the test documentation identifies those security functions to be tested (ATE_FUN.1) which are the security-critical functions, there should be no tests of non-security-critical functions in the test documentation. Therefore, all of the tests in the documentation must be tested.

EAL3-Augmented assurance satisfies the IT objective O.ASSURE.

5.3.3 Dependency Analysis

Table XI shows the dependency analysis for the functional requirements in this ST. Every requirement is cross-referenced with dependencies, showing that all dependencies are satisfied.

Table XI - Dependency Analysis

Table Line Number	Component	Dependencies	Reference to Table Line Number
1	FAU_GEN.1	FIA_UID.1	22 (h)[1]
2	FAU_MGT.1	FAU_STG.1	7 (h)
3	FAU_POP.1	FAU_STG.1	7 (h)
4	FAU_PRO.1	FAU_STG.1 FPT_TSA.1	7 (h) 27 (h)
5	FAU_SAR.1	FAU_STG.1 FPT_TSA.1 FAU_PRO.1	7 (h) 27 (h) 4
6	FAU_SAR.3	FAU_SAR.1	5
7	FAU_STG.3	FAU_GEN.1	1
8	FDP_ACC.2	FDP_ACF.1	9 (h)
9	FDP_ACF.2	FDP_ACC.1	8 (h)
10	FDP_ACF.4	FDP_ACC.1	8 (h)

[1] (h) implies that the dependency is satisfied hierarchically. Because these are hierarchies of function, rather than of security, these components were analyzed to ensure that the security required of each dependency was met.

11	FDP_RIP.3	-	-
12	FDP_SAM.1	FPT_TSA.1 FDP_ACC.1	27 (h) 8 (h)
13	FDP_SAQ.1	FPT_TSA.1 FDP_ACC.1	27 (h) 8 (h)
14	FIA_ADA.3	FPT_TSA.1 FIA_ADP.1 FIA_UAU.1	27 (h) 15 21 (h)
15	FIA_ADP.1	FIA_UAU.1	21 (h)
17	FIA_ATA.1	FIA_ATD.1 FPT_TSA.1	19 (h) 27 (h)
18	FIA_ATA.2	FIA_ATD.1 FPT_TSA.1	19 (h) 27 (h)
19	FIA_ATD.2	ADV_FSP.1	assurance
20	FIA_UAU.1	FIA_UID.1 FIA_ADA.1	22 (h) 14 (h)
21	FIA_UAU.2	FIA_UID.1 FIA_ADA.1	22 (h) 14 (h)
22	FIA_UID.2	FIA_ATD.2	19 (h)
23	FPT_AMT.2	-	-
24	FPT_REV.1	-	-
25	FPT_RVM.1	-	-
26	FPT_SEP.1	-	-
27	FPT_TSA.2	FIA_UID.1 FIA_ATD.1 FIA_ATA.1 AGD_ADM.1	22 (h) 19 (h) 17 assurance
28	FPT_TSM.1	FPT_TSA.1	27 (h)

5.4 Rationale for TOE IT Security Functions

The purpose of this section is to show that the security functions provided by Black Hole meet the security objectives for this ST, and to show that the strength-of-function claims made for Black Hole are valid.

The reader is referred to Chapter 4, IT Security Requirements, for a mapping from the Black Hole security functions to each security objective, and a discussion of the strength-of-function claims for Black Hole.

Appendix A - Black Hole Security Policy Paradigm

This appendix informally describes the security policy of Black Hole. Like many firewalls, Black Hole enforces a security policy derived primarily from rules that are set and modified by the administrator. As a result, there is no fixed set of accesses, which means that the security policy must be stated in its most basic terms: a rule must explicitly allow access permission in order for that permission to be granted.

Within the bounds of this underlying enforcement principle, there are further refinements, based upon the constraints that are enforced by those rules; while it is true to say that there must be a rule, it is more useful to say what that rule governs. Access mediation security policy models are traditionally expressed in terms of: entities to which access is attempted (objects); active entities that attempt such access (subjects); the operations performed upon objects by subjects (access methods); and rules that govern whether or not access requests are granted (security policy).

A.1 Traditional Security Policy Modeling Paradigms

Generally, subjects are allowed to access objects in a variety of ways. For example, in the UNIX multi-user paradigm, processes (subjects) can access files (objects) in three distinct ways: subjects can read; write; and execute files. Whether or not a subject is granted a requested access is based on: the subject's security attributes; the object's security attributes; and the policy rules.

Subject security attributes contain information that identifies the entity making the request, and are based on certain pieces of known (or verified) information from the requester (for example, a user's identity or security clearance level). Once a subject is instantiated, its attributes are not modifiable. Object security attributes contain security-relevant information about the object (for example, an access control list, or a sensitivity label). This information is often inherited from the subject creating the object, and can contain access methods which the creator has assigned to the object. For example, a UNIX file system assigns an owner's user ID to an object upon its creation; in addition, protection bits can be set by the owner.

Policy rules can be described as the triple cross-product of all subjects, all objects and all access method combinations. For example, the UNIX protection rule set could be characterized using the discretionary access control matrix, as defined in the Bell/Lapadula security policy model [BEL76], where a (subject $\times$ object $\times$ access method) triple would describe whether the subject had a particular access method to the object. A null access method could be looked upon as an inactive rule, because no action is taken to grant access (which signifies denied access, by default). A non-null access method could be described as an activated rule, because action must be taken to grant a specific type of access to a subject, which has the appropriate subject protection attributes, in order to access the object.

It is worth noting the underlying assumption in the foregoing modeling paradigm: that all objects and subjects are entirely under the control of the entity mediating the request. That is, the creation and deletion of subjects and objects, as well as all their interactions, are performed entirely by the product.

A.2 Firewall Security Policies

Traditionally, subjects are internal, untrusted, active entities, running on behalf of untrusted users. A proxy-based firewall such as Black Hole has no such entities. The only internal active entities running on behalf of remote users are the proxies, which are trusted; the only untrusted active entities running on behalf of the users are the remote processes, which are not internal.

There is similar difficulty in applying the traditional view of objects to firewalls such as Black Hole, in which users do not attempt access to *data*, but to *communication pathways through the firewall*. There are no resources on the firewall that are created or destroyed by users. As such, the resources being protected are not entirely under the control of the product.

Black Hole responds to a service requests from an entity on an external network seeking access to an entity residing on the internal network, or vice versa. It is the responsibility of the firewall to grant or deny this request. The firewall paradigm therefore calls for a non-traditional view of subjects and objects.

A.3 Black Hole Security Policy

The subjects, objects, and operations of a firewall such as Black Hole are defined as follows.

- A firewall subject is an active entity whose behavior through the firewall is controlled by the TSF.
- A firewall object is a remote resource to which access through the firewall is controlled by the TSF.
- A firewall operation is the means by which a subject accesses an object.

Black Hole's subjects and objects were determined by identifying the entities whose attributes were used in the mediation being performed. The subjects are: the remote user processes; the remote (source) hosts; and the (source) networks. The Black Hole objects (upon which the access performed) are: the destination hosts, and the destination networks. The operations mediated by Black Hole (which define the ways in which subjects may communicate with objects) are the services (Telnet, FTP, UDP, HTTP, etc.) and (for FTP and mail) the service commands. (Note: each variation of a service resulting from the setting of protocol option flags constitutes a distinct service.)

The firewall does not claim to have total control over those entities which it cannot control (for example, hosts). It claims only to control what it passes from an input physical connection (representing the subject network) to an output physical connection (representing the object network), based on certain types of access requests and a specified rule set.

A.4 Black Hole Rule Set

The network security policy is instantiated by sets of rules defined by an administrator. Each rule defines an allowed operation between subjects and objects. Given: the subject (identified by source IP address and, for remote users, the authenticated user ID); the object (identified by destination IP address); and the operation, the rule set must contain a rule permitting the subject to perform the operation upon the object.

No rules shall apply to VPN, as it is not covered by the untrusted-user access control policy.

For all other services, the following configurable rules shall apply. If:

- the subject is permitted to use the requested service type;
- the access request is within the permitted range of access times for that subject (except for mail);
- the subject is permitted to communicate with the specified object;
- the subject is permitted to use the requested command (only for mail and FTP); and
- the subject is permitted to use the requested service,

then access is granted to the requested object.

In addition to its configurable rule set, Black Hole also has the following fixed kernel-level rules, which define the filtering of packets.

- A service request by a subject from the external network may not have a source IP address which is known to belong to the internal network (or vice versa).
- A service request by a subject may not specify source routing.
- A service request by a subject may not specify ICMP Redirection.
- A service request by a subject may not be directed to a destination port of 514 (`syslog` port) on the Black Hole Firewall.

Appendix B - Black Hole Security Policies

This appendix describes the security policies enforced by Black Hole: identification and authentication (I&A); access mediation; and audit. (Note: these descriptions are of the *policies* being enforced within the evaluated configuration, rather than of the *mechanisms* that enforce those policies. Descriptions of the mechanisms are presented in the Black Hole final evaluation report [CSE97a], in Section 4.2, Software Architecture).

B.1 Identification & Authentication Policy

The concept of identification (who you claim to be) and authentication (confirming that you are who you claim to be) is a concept fundamental to most IT security architectures. The I&A policy enforced by Black Hole is integral to the overall security policy that the product is designed to enforce, and can be considered separately for network users, for administrators, and for packets.

Black Hole allows for the unique identification of all network users. Each user is assigned (by the administrator) a unique identifier, to be used to identify himself/herself to Black Hole upon initiating a connection request. The authentication of the user identity is performed either by a reusable password, or by a single-use authentication mechanism. The choice of either a reusable or a single-use authenticator is made by the administrator, when creating the user's entry in the policy database. (The assignment of the authentication mechanisms to be used must be decided by the accreditor, the certifier, or other security personnel.)

Only firewall administrators are allowed to log in at a Black Hole console. Each administrator must have a unique account name, for identification purposes. For an administrator to log in at the console, he or she must provide the correct UNIX user name and password.

Black Hole also performs source authentication on all IP packets entering the firewall. The claimed source address within the packet is verified against the address of the network interface card (NIC) on which it was received. This check ensures that any packet received from the external side of the firewall, but which appears to have originated from the internal side (or vice versa), will be disallowed.

Some Black Hole connection proxies (TCP, UDP, SSL, RealAudio and mail) operate only at the circuit level, and do not have the ability to request a user to provide I&A data. These non-authenticating proxies can be configured in the rules database, such that they can be used only after a user has enabled `transparent` mode. To enable `transparent` mode, a user must undergo an I&A challenge (via a service with authentication capability).

Black Hole also allows some "user-less" requests to be initiated, in order to support the collection and subsequent forwarding of electronic mail, and for communication through a VPN tunnel between two firewalls. There is no user-based authentication provided on these requests.

B.2 Access Control Policies

An administrator's role and responsibilities differ greatly from those of an ordinary network user, so separate access control policies are required for each . Consequently, Black Hole's access

control policies cover two types of users: trusted (administrators), and untrusted (users). Administrators can access Black Hole only from the local console. Users can only request services, from either the internal or the external networks.

The access control policy for users defines how untrusted entities interact (for example, how untrusted processes get access to user data). The access control policy does not constrain trusted processes (they are allowed to circumvent the policy, because their behavior is known). Similarly, the policy does not extend to trusted data structures (such as user passwords). Untrusted processes do not have access to these structures; rather, they must communicate with trusted processes, which have direct access.

VPN links two networks logically, making both networks act as one large network. Therefore, the mediation which would normally occur between an internal and external network does not apply. Data transmission through a VPN tunnel is accomplished solely through routing information: the VPN tunnel is identified by a source/destination address pair. Any service that attempts to connect to a site at the other end of a tunnel will be routed through the VPN tunnel by the Black Hole kernel (based on the destination IP address and the routing table).

B.3 Trusted Administrator Access Control Policy

Administrators have standard `root` access. The default configuration of Black Hole is for one administrator; the creation of subsequent administrator accounts requires manually editing the password file. All administrators will have the same `root` access, so it is not possible to limit or separate duties between administrators.

Administrators access Black Hole through its GUI. They set the desired security policy by adding users and rules to Black Hole. All log files are available to administrators, who can read, create and empty the files as necessary. Each VPN must be configured and established by an administrator. As well, administrators have access to all configuration files, and are responsible for system backups.

B.4 Untrusted User Access Control Policy

All traffic is mediated as it crosses Black Hole, so that no packets are allowed to flow freely through. Users can interact with Black Hole only through proxies, to request services which allow for: the enabling and disabling of `transparent` mode; the changing of user passwords; and the establishment of connections through the firewall. It is assumed that only administrators will have physical access to the console; therefore, untrusted users will have no access to privileged files and processes.

Black Hole controls user access based on type of service and allowed commands, as listed in pre-defined rules. The rules are either *static* (that is, those that are unchangeable by administrators, such as kernel-level packet filtering), or *dynamic*. A dynamic rule is created by the administrator so that when a user makes a connection request, Black Hole decides whether to allow or deny the request based on its connection attributes (time of day; service type; source and destination IP addresses) and user authentication (user ID, and fixed or one-time password mechanism). Additionally, mail and FTP can be controlled by limiting their service commands. In the event of conflicting rules, Black Hole chooses the most restrictive rule.

The general policy of Black Hole is that any connection requests which are not covered by an explicit rule are denied by default. An exception to this underlying "most-restrictive" principle is for mail, because: mail is not user-session-based; and if mail is on a system, it is generally receivable by any user. Therefore, mail rules are enforced on a least-restrictive basis. One must be explicitly prohibited (on an IP address basis) from using mail.

Although the application-level filtering of FTP commands is similarly least-restrictive by default, administrators can change the rules to be more restrictive, if desired. (The rules governing access to an FTP service itself are, like those of all other services, most-restrictive.)

B.5 Audit Policy

Black Hole detects and records the occurrence of security-relevant events, as defined by the administrator. The events are detected by the relevant subsystems and forwarded to the logging daemon which writes them into the audit file. The audit records can then be sorted and reviewed. An administrator can set up an auxiliary alarming daemon, that scans the audit records (as they are written to the audit file) for specific types of events, and then alerts the administrator that such events have occurred.

Since Black Hole administrators are considered to be trusted personnel, they are trusted to not circumvent the security policy. As a result, the auditing of administrator events is not designed to prevent administrators from doing anything sinister, but merely as a means of housekeeping; an administrator can trace actions taken, in order to determine how a current system state was produced. Administrator actions that change the system state (logging in, changing databases, etc.) are security-relevant and, therefore, auditable. Administrator actions that do not change the state (such as the non-destructive read of a database) are not audited.

Black Hole's audit policy recognizes the following auditable events:

- system startup/shutdown;
- administrator logon/logoff;
- GUI startup/shutdown;
- subsystem (Guardian, The Oracle, `vpnd`, proxy, etc.) startup/shutdown;
- administrator changing a user record;
- administrator changing a rule;
- administrator editing or backing up a database or configuration file;
- administrator rolling over an audit data log;
- user authentication;
- user or administrator changing a user's password;
- toggling of `transparent` mode;
- kernel-level filtering detecting a prohibited packet (for example, an ICMP redirect);
- refusal of a connection;
- establishment or termination of a connection;
- attempted use of FTP commands;
- attempted use of prohibited mail commands (`wiz`, `debug`, etc.);

- file system filling up;
- `syslogd` startup/shutdown;
- alarms sent;
- changing of system time/date; and
- establishment or termination of a VPN connection.

Although both the creation and the termination of a VPN tunnel are audited, communication traversing a VPN tunnel is not audited. The tunnel is not audited because VPN connects two remote hosts so that they are for all intents and purposes one network, and the activities transpiring between them are completely internal.

Glossary

access	The ability, in a firewall, to enable a service.
Anonymous FTP	An FTP service wherein users are not authenticated before being granted access to read public files. Users typically login using "anonymous" as the user name, and without a password.
authorized administrator	A human user to whom the authorization has been granted to perform administrative operations which may affect the enforcement of the TSP.
cron	A background process, running on a UNIX system, which periodically executes commands defined in a configuration file.
Domain Name Service	The on-line distributed database system used to identify host IP addresses as human-readable machine names.
File Transfer Protocol (FTP)	An application used to transfer files from one site to another. Users normally use an FTP client program to access an FTP server.
Gopher	Both a protocol and an application. It is used as an information-browsing tool on the Internet. Users normally use a Gopher client program to browse through information stored on a Gopher server. Gopher clients and servers communicate using the Gopher protocol.
HyperText Transfer Protocol (HTTP)	The protocol used on the World Wide Web to retrieve pages.
Internet Control Message Protocol (ICMP)	Part of the Internet Protocol (IP) that handles error and control messages.
ICMP Ping	A program used with TCP/IP networks to test the reachability of destinations, by sending an ICMP echo request and waiting for the reply. Also known as PING (Packet InterNet Groper).
private network	An internal network segment that is to be protected from external or untrusted network segments.
proxy	A method whereby a process pretends to be the intended recipient host, thereby ensuring secure communication through a gateway.
public network	An external network segment, that is considered to have all, or mostly, untrusted users.
service	A third layer (within the TCP/IP communications protocol model)

	communications type. Also known as “service class.”
Simple Mail Transfer Protocol (SMTP)	The TCP/IP standard protocol for the electronic transfer of mail messages.
subnet	A portion of a network, identified by an extension of the IP-addressing scheme, that allows a single IP address (the subnet address) to be used for multiple physical devices or networks.
Transmission Control Protocol (TCP)	A connection-oriented protocol that provides reliable virtual circuits, running atop IP.
target of evaluation (TOE)	The part of an information technology system or product that is subjected to security evaluation.
TOE security functions (TSF)	All parts of the TOE which have to be relied upon for enforcement of the TOE Security Policy (TSP).
TOE security policy (TSP)	The totality of the rules and objectives that define the security behavior of a TOE.
Telnet	A remote terminal application that allows users to access a remote computer. The user normally uses a Telnet client program, and the remote computer must have a Telnet server running (usually telnet on UNIX).
User Datagram Protocol	A connectionless (unreliable) protocol that presents applications with IP services.
Wide Area Information Service (WAIS)	An Internet application that provides wide-area information searching on any subject. Users normally use a WAIS client program to search databases on the WAIS.
World Wide Web (WWW)	An application used as an information-browsing tool on the Internet. A WWW client program (browser), such as Netscape Navigator; Mosaic; Cello; or Viola, accesses information stored on servers. WWW clients and servers communicate primarily using the HyperText Transfer Protocol (HTTP); however, they can also communicate with Gopher servers, News servers, and FTP servers.

Bibliography

[BEL76] Bell, David E. And Leonard J. LaPadula, *Secure Computing Systems: Unified Exposition and Multics Interpretations*, MTR-2997, rev. 1, The MITRE Corporation, Bedford, Mass. , 1976.

[COM96] Common Criteria Editorial Board, *Common Criteria for Information Technology Security Evaluation*, Version 1.00, 1996.

[CSE97a] Communications Security Establishment, *Final Evaluation Report for Milkyway Networks Black Hole Firewall Version 3.01E2 for SPARCstations*, 1997.

[ISS97a] http://www.iss.net/vd/, then click the 'Internet Scanner UNIX checks' link.

[MIL96a] Milkyway Networks Corporation, *Black Hole Administration Guide*, version 3.01, 1996.

[MIL96b] Milkyway Networks Corporation, *Black Hole User Guide*, version 3.01, 1996, 44 pages.

[MIL96c] Milkyway Networks Corporation, *Black Hole Installation Guide*, version 3.01, 1996, 55 pages.

[MIL96d] Milkyway Networks Corporation, *Document Release Procedures*, Release 1, Issue 1, 1996.

[MIL96e] Milkyway Networks Corporation, *Milkyway New Product Development Cycle*, Release 1, 1996.

[MIL96f] Milkyway Networks Corporation, *Controlled Documents - Policy and Procedures*, Release 1, Issue 2, 1996.

[MIL96g] Milkyway Networks Corporation, *Orientation to Black Hole for System Administrators.*

[MIL96h] Milkyway Networks Corporation, *Program Description Document*, Release 1, Issue 1, 1996.

[MIL97a] Milkyway Networks Corporation, *Security Policy and Procedures*, Release 2, Issue 1, 1997.

[MIL97b] Milkyway Networks Corporation, *Black Hole Functional Specifications*, Release 4.0, Issue 1, 1997.

[MIL97c] Milkyway Networks Corporation, *Black Hole Module and Interface Specifications*, Release 2, Issue 1, 1997.

[MIL97d] Milkyway Networks Corporation, *Vulnerability Assessment - Black Hole v3.0*, Release 1, Issue 2, 1997.

[MIL97e] Milkyway Networks Corporation, *Configuration Management Plan*, Release 3, Issue 1, 1997.

[MIL97f] Milkyway Networks Corporation, *Engineering Procedures*, Release 2, 1997.

[MIL97g] Milkyway Networks Corporation, *CVS User's Manual*, 1997.

[NCS97a] National Computer Security Association home page, http://www.ncsa.com.

[UCB95] Regents of the University of California, *Postgres95 User Manual*, Version 1.0, 1995.