Security Target

for the

netfence firewall Version 3.0-2

of

phion IT GmbH

Date	**25.1.2007**
Version No.:	**1.9**
Author:	**phion IT GmbH**

Table of Contents

1 SECURITY TARGET INTRODUCTION ... 4
1.1 ST IDENTIFICATION ... 4
1.2 CONVENTIONS, TERMINOLOGY, AND ACRONYMS ... 4
1.2.1 Conventions ... 4
1.2.2 Terminology ... 4
1.2.3 Acronyms ... 5
1.3 SECURITY TARGET OVERVIEW ... 6
1.4 COMMON CRITERIA CONFORMANCE ... 7
2 TOE DESCRIPTION ... 8
2.1 PRODUCT TYPE ... 8
2.2 SCOPE AND BOUNDARIES OF THE EVALUATED CONFIGURATION ... 10
2.2.1 Physical Scope and Boundary ... 10
Included in delivery: ... 10
2.2.2 Outside Scope ... 10
2.2.3 Security Features ... 11
3 TOE SECURITY ENVIRONMENT ... 12
3.1 THREATS ... 12
3.1.1 Threats addressed by the TOE ... 12
3.1.2 Threats addressed by the Operational Environment ... 13
3.2 ORGANISATIONAL SECURITY POLICIES ... 13
3.3 ASSUMPTIONS ... 13
4 SECURITY OBJECTIVES ... 14
4.1 SECURITY OBJECTIVES FOR THE TOE ... 14
4.2 SECURITY OBJECTIVES FOR THE ENVIRONMENT ... 15
5 IT SECURITY REQUIREMENTS ... 16
5.1 TOE SECURITY REQUIREMENTS ... 16
5.1.1 TOE Security Functional Requirements ... 16
5.1.2 TOE Security Assurance Requirements ... 25
5.2 SECURITY REQUIREMENTS FOR THE IT ENVIRONMENT ... 37
6 TOE SUMMARY SPECIFICATION ... 38
6.1 TOE SECURITY FUNCTIONS ... 38
6.1.1 Security Administration ... 38
6.1.2 Identification and Authentication ... 42
6.1.3 Information Flow Control ... 43
6.1.4 Privacy ... 45
6.1.5 Security Audit ... 46
6.2 ASSURANCE MEASURES ... 48
7 PROTECTION PROFILE CLAIMS ... 50
8 RATIONALE ... 51
8.1 SECURITY OBJECTIVES RATIONALE ... 51
8.2 SECURITY REQUIREMENTS RATIONALE ... 53
8.2.1 Traceability and Suitability of Functional Requirements ... 53
8.2.2 Rationale for Assurance Requirements ... 57
8.2.3 Rationale for Strength of Function ... 57
8.2.4 Rationale for Mutual Support of Security Requirements ... 58
8.3 TOE SUMMARY SPECIFICATION RATIONALE ... 61
8.3.1 Mapping between TOE Security Functions and SFRs ... 61
8.3.2 Mapping between Security Measures and Assurance Requirements ... 65
8.4 PROTECTION PROFILE CLAIMS RATIONALE ... 65

Revisions to Document

Version	Date	Changes made
1.0	24.01.2003	Initial Version
1.1	06.06.2003	Changes according to Review Protocol 1
1.2	25.08.2003	Changes according to Review Protocol 2
1.3	23.09.2003	Changes according to discussion with BSI (16.09.03) and discussion with ITSEF
1.4	02.06.2004	Changes according to discussion with BSI (16.02.04) and discussion with ITSEF
1.5	11.10.2004	Changes according to discussion with BSI (19.08.04) and discussion with ITSEF
1.6	15.01.2005	Changes according to discussion with BSI (02.12.04) and discussion with ITSEF
1.7	28.09.2005	Changes according to discussion with ITSEF
1.8	10.1.2007	Adjust final netfence CD Version
1.9	25.1.2007	Change document title

1 Security Target Introduction

1.1 ST Identification

Title: Security Target for the Phion netfence firewall Version 3.0-2 of phion IT GmbH

Assurance Level:	EAL4, augmented by AVA_VLA.3 and ALC_FLR.1
CC Version:	2.1

Note that all references to the netfence firewall version 3.0 imply references to the software version 3.0-2 which is deployed on the CD labeled “**netfence firewall 3.0 Version 3.0-2**”

1.2 Conventions, Terminology, and Acronyms

This section identifies the formatting conventions used to convey additional information, specific terminology and acronyms used throughout the remainder of the document.

1.2.1 Conventions

This section describes the conventions used in chapter 5 to denote CC operations on security requirements. The CC allows several operations to be performed on functional requirements; *assignment*, *iteration*, *refinement*, and *selection* are defined in paragraph 2.1.4 of Part 2 of the CC.

- The selection operation is used to select one ore more options provided by the CC in stating a statement. Selections are denoted by *underlined italicised text*.
- The assignment operation is used to assign a specific value to an unspecified parameter, such as the length of a password. An assignment is indicated by showing the value in square brackets [assignment value(s)].
- The refinement operation is used to add detail to a requirement, and thus further restricts a requirement. Refinement of security requirements is denoted by **bold text**.
- Iteration of a component is used when a component is repeated more than once with varying operations. Iterated components are given unique identifiers by an iteration number in parenthesis appended to the component and element identifiers.

1.2.2 Terminology

Session

An allowed flow of IP datagrams between two hosts (IP addresses) through the netfence firewall or from a management workstation to the netfence firewall constitutes a session. The state of a session comprises information about the initiator’s IP and port (if applicable) and the target’s IP and port (if applicable), protocol, and the remaining lifetime.

Pseudo Session

In order to keep track of the state of allowed connections between hosts (IP addresses) communicating through a connectionless protocol (all except TCP; UDP or ICMP in particular) the netfence firewall assigns each such allowed flow of IP datagrams a pseudo session with limited lifetime.

Pending Session

Refers to sessions which have been initiated but have not yet been successfully established. As there are timeouts associated with session establishment a pending state is assigned to the session.

Proxy

Refers to the ability of dynamic network address translation of an IP datagram forwarding device.

Statefulness, Stateful

Refers to the ability to keep track of the state of a flow of IP datagrams across an IP datagram forwarding device. The state comprises information such as initiator IP and port (if applicable), target IP and port (if applicable) or protocol based information such as IP address or port mappings/changes as required by certain protocols (e.g., active FTP). This information is taken into account in the decision process as to whether or not to forward arriving IP datagrams.

User

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

Human User

Any person who interacts with the TOE.

External IT Entity

Any IT product or system, untrusted or trusted, outside of the TOE that interacts with the TOE.

Authorised User

A user who, in accordance with the TSP, performs an operation.

Unauthorised User

A user who attempts to perform an operation which is not in accordance with the TSP. This attempt might occur unintentionally (for example by mistyping an IP address) or intentionally.

Attacker

An unauthorised user who attempts to violate the TSP.

Demilitarized Zone

A protected network which is separated from the internal and external networks by the TOE since it contains network hosts providing network services to network hosts outside the internal networks as, e.g., a web server run by company for its customers.

1.2.3 Acronyms

The following abbreviations are used in this Security Target:

ACL	Access Control List
ACPF	Application Controlled Packet Forwarding
ARP	Address Resolution Protocol
CC	Common Criteria for Information Technology Security Evaluation
CM	Configuration Management
EAL	Evaluation Assurance Levels
GUI	Graphical User Interface

HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
IP	Internet Protocol
IT	Information Technology
MAC	Media Access Control
NAT	Network Address Translation
NIC	Network Interface Card
POP	Post Office Protocol
PP	Protection Profile
SMTP	Simple Mail Transfer Protocol
ST	Security Target
SFP	Security Function Policy
TAP	Transparent Application Proxy
TCP	Transport Control Protocol
TOE	Target of Evaluation
TSC	TSF Scope of Control
TSF	TOE Security Functions
TSP	TOE Security Policy
UDP	Universal Datagram Protocol

1.3 Security Target Overview

The netfence firewall system allows to control IP traffic between different networks and in particular from and to the internet. The system allows traffic propagation either by stateful packet forwarding or by transparent application proxying. Traffic control is configured through a manageable firewall rule set based on which decisions are made whether traffic should be propagated or blocked.

The phion netfence firewall system comprises the following components:

Netfence firewall service – Software consisting of a collection of daemon processes that control packet forwarding and transparent proxying.

Netfence firewall kernel extensions – Linux kernel modules that implement Application Controlled Packet Forwarding (ACPF) and support transparent proxying with additional security features (SYN Protection). The netfence firewall kernel extension is a loadable kernel module that adds firewalling functionality, used by the netfence firewall system, to the standard linux kernel.

Netfence firewall base system – A collection of software modules allowing the administrator to control and analyse the status of the netfence system as well as that of the underlying Linux system. The netfence firewall base system also provides the interfaces for authorised rule set and system attribute management.

Phiona – Software running on a Windows NT/2000/XP system allowing remote management of the netfence firewall. This involves firewall rule management, status visualisation as well as security audit evaluation.

Phioni – Software running on a Windows NT/2000/XP system allowing to preconfigure a netfence system for installation.

1.4 Common Criteria Conformance

The TOE is

- Part 2 conformant
- Part 3 conformant, with a claimed Evaluation Assurance Level of EAL4 augmented by ALC_FLR.1 and AVA_VLA.3.

2 TOE Description

This section provides a product description in order to point out its purpose and possible fields of application. Furthermore the scope of the evaluated configuration is defined.

2.1 Product Type

The netfence firewall system controls IP traffic between network nodes located in separated networks. The firewall system acts as an IP datagram router that controls datagram flow according to a configurable security policy which allows regulation of all IP protocols.
To this end the product provides an ***Application Controlled Packet Forwarder (ACPF)*** as well as a ***Transparent Application Proxy (TAP)***. To clarify the difference between these coexisting methods a brief characterisation of the two is given below:

- **Application Controlled Packet Forwarder**
  This method acts on IP packets (datagrams). For each datagram a decision based upon the configured firewall rules is made to control traffic between nodes. If the carried network protocol allows assignment of datagrams to sessions (e.g. TCP or UDP pseudo sessions) a state of these sessions is kept (*stateful*) and is taken into account in the decision process. This way the decision process is divided into two types: A slow decision for packets not belonging to an established session by consulting the firewall ruleset and a fast decision based upon a lookup in a table of allowed established sessions.
- **Transparent Application Proxy**
  This method controls data stream s (TCP) from one network node to another. Based on the address (IP-Address and Port Number) of the initiating node (source) and the responding node (destination) establishments of such data streams can be allowed or denied as seen fit by the firewall ruleset. The system acts as an endpoint for the source node and as an initiator for the destination, controlling and analysing the flow and its content. This way even if the stream between source and destination is separated by the intervening firewall the communicating nodes do not have to have knowledge of the firewall in-between (t*ransparency*) to perform an authorised data exchange.

For each of these two transport methods (ACPF and TAP) two operation modes, inbound mode and outbound mode, are provided. The inbound method is provided to shield protected network nodes from TCP-SYN attacks performed across the TOE which aim at resource exhaustion of the protected node or nodes. To this end the TOE will first expect the three-way TCP handshake with the inititiating source node to be completed before attempting to connect the protected target network node. In outbound mode an incoming TCP-SYN packet is immediately passed on to the target network node.

The security policy as well as the system configuration can be maintained by three different kinds of administrators.This allows to choose an appropriate role for the indended tasks of the administrator.

- The **“root” administrator** has unlimited access to the system and may grant access to the system to two kind of “named” adminstrators. The “root” administrator is the primary maintainer of the system. There is only one instance.
- The **read/write administrator** may visualize and modify the security policy and the system configuration but may not maintain other administrators. Read/write administrators maintain the security and the system configuration and visualize the system status for analysis or trouble shooting purposes.

- The **read-only administrator** may only visualize system data with no permission to modify any data. Its typical task is to visualize the firewall real time status for analysis or trouble shooting purposes.

The TOE is built of the following components

- Netfence firewall service (TAP, ACPF)
- Netfence kernel extensions (ACPF and Application Protection)
- Netfence firewall base system (Visualization and Configuration)
- Phiona (Remote administration client application)
- Phioni (Preinstallation setup)

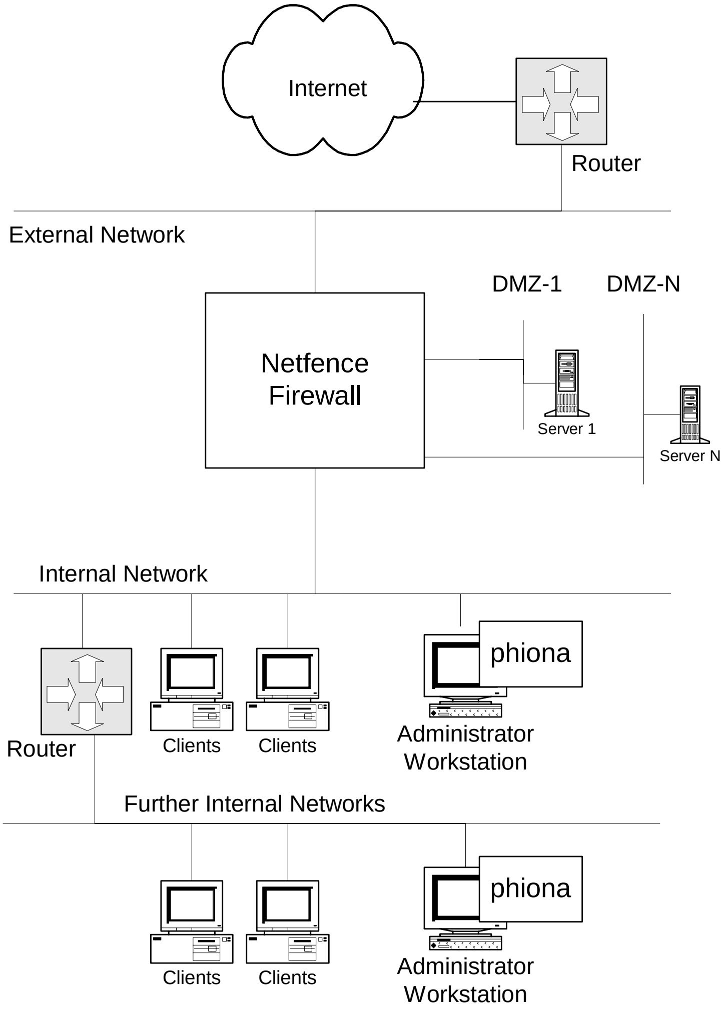

Fig. 1 Schematic diagram of a typical way in which a netfence firewall can be used and administered. The firewall separates internal network, demilitarised zone networks, and external networks from each other. Administration tasks are carried out with GUI-tool phiona from Windows®-based secure administrator workstations on the internal networks.

2.2 Scope and Boundaries of the Evaluated Configuration

2.2.1 Physical Scope and Boundary

For using the netfence firewall system the following physical components are needed:

- An Intel x686-based PC acting as hardware platform for the firewall. The operating system used is Linux with Kernel Version 2.4.

Hardware	Intel x686 compatible PC with at least 128 MB of memory 4 GB Hard Disk 1 CDROM for installation 1 1.44 MB Floppy disk drive for installation 2 Network interfaces
Software	Unix System (Linux with kernel 2.4) Netfence firewall version 3.0-2

- An Intel x86 compatible PC with Windows NT4.0, 2000 or XP installed acting as workstation for remote management. On the workstation the administrative tools phiona (remote management tool) and phioni (preinstallation setup) must be present.

Hardware	An Intel x86 compatible PC with 128 MB of memory 4 GB Hard Disk 1 CDROM for software installation 1 1.44 MB Floppy disk drive for preparation of firewall installation 1 Network interface
Software	Windows NT4.0 (Service Pack 6a or newer), 2000 or XP Phioni (for installation or recovery purposes only) Phiona remote administration software

Included in delivery:

Phion Netfence Version 3.0-2 CD ROM	Linux Operating System Netfence Firewall Version 3.0-2 User Documentation and Administrator Guidance (PDF Format)

2.2.2 Outside Scope

- Operating System (Linux)
- Netfence firewalls managed by the phion management centre
- High Availability
- Advanced Routing (Policy Routing, Redundant Routes)

2.2.3 Security Features

Feature	Comment
Security Audit	Complete audit trail
Information Flow Control	Is enforced by a security policy which is defined by the firewall ruleset. With this ruleset the firewall administrator can decide to allow specific network traffic from and to certain nodes (IP addresses or networks) while blocking others. The ruleset allows the administrator to decide on a per firewall rule basis whether TCP network traffic is to be handled via application controlled packet forwarding (ACPF) or transparent application proxying (TAP). All other IP traffic is handled via the ACPF method. For both transport methods two operation modes (inbound and outbound) are provided by the security policy again on a per firewall rule basis. The inbound mode can be used to counteract SYN attacks on protected network hosts across the TOE.
Identification and Authentication	Phion netfence firewall restricts administrative access for identification, authentication and security management purposes already on the IP address level based on an access control list containing network addresses from which such access is allowed. No administrative action is possible prior to successful administrator authentication. Role identification after successful authentication effects a separation of root administrator, read-only and read-write enabled administrators.
Privacy	NAT and transparent proxying hide internal addresses.
Security Management	Provides access to firewall status and audit data. Allows management of the firewall rule set. The scope of access to the security management for each administrator is defined by the security management role assigned to the administrator. Three different roles (root administrator, read-only and read-write enabled administrators) exist.

3 TOE Security environment

The TOE is meant to protect data (IT assets) residing on network entities in the protected network against unauthorized access. The IT assets requiring protection are for example the services provided by, and data accessible via, hosts on the demilitarized zone and internal network (or networks if there are multiple network interfaces on the TOE being attached to one of these types of networks).This involves network entities which are used to gain access to the internet or an unprotected network as well as network entities that serve the purpose to supply information to the internet (usually placed in a demilitarized zone network).

3.1 Threats

Threats may be addressed either by the TOE or by its intended environment (for example, using personnel, physical, or administrative safeguards). These two classes of threats are discussed separately. The related attack potential is described in Assumption A.MEDEXP.

3.1.1 Threats addressed by the TOE

The following table identifies the threats which are addressed by the TOE:

Table 1: Threats addressed by the TOE

Name	**Description**
T.NOAUTH	An unauthorised human user may attempt to bypass the security of the TOE so as to access and use security functions provided by the TOE.
T.ASPOOF	An unauthorised user may carry out spoofing in which information flows through the TOE into the connected network by using a spoofed source address for TCP connections. An unauthorised user may carry out spoofing in which information flows through the TOE into the connected network by using a spoofed source address for all IP protocols for which a reverse routing path check from the TOE back to the source address yields a network device of the TOE other than the one the request from the source arrived on .
T.MEDTF	An unauthorised user may send impermissible network information through the TOE which results in the exploitation of resources on a protected network.
T.PRIVACY	A user may send information to the TOE and may analyse information received from the TOE to determine real IP addresses of external IT entities (network nodes such as hosts providing services or access to other networks) on the internal and demilitarized zone networks based on information extracted from received IP protocol headers. He may gain information about the IP addresses or TCP stacks used by the network nodes on the internal or demilitarized zone networks or about the topology of the protected networks. Retrieved information could be used by the user to optimise an attack strategy on network nodes within the protected networks.
T.NODETECT	An unauthorised user may continually attempt to bypass the TSP without detection in order to successfully send data through the TOE.

3.1.2 Threats addressed by the Operational Environment

There are no threats to be addressed by the operational environment.

3.2 Organisational Security Policies

The following table identifies the organisational security policy which has to be met by the TOE:

Table 2: Policies to be met by the TOE

Name	Description
P.ROLE	The TOE must be able to distinguish between a root administrator with unrestricted management access, administrators with read/write permissions and administrators with read-only permissions.
P.AUDACC	Users must be accountable for the actions that they conduct.

3.3 Assumptions

The following table identifies the assumptions about the intended usage of the TOE and about the environment of use of the TOE:

Table 3: Assumptions

Name	Description
A.MEDEXP	Potential threat agents attempting to attack the TOE are considered to be of a moderate attack potential. This incorporates familiarity with internet protocols, firewall principles and design, information published about the TOE, as well as tools and techniques for firewall penetration testing.
A.NOEVIL	Administrators are non-hostile, competent, trained, and follow all administrator guidance.
A.ONEWAY	Information cannot flow between networks connected to the netfence firewall unless it passes through the netfence firewall.
A.PHIONA	After the netfence firewall has been installed, administrators use a Management Workstation to administrate it, not the system console.
A.PHYSEC	The netfence firewall is operated in a physically secure environment which prevents access from unauthorised users.
A.WSSEC	The Management Workstation is operated in an environment which is free of malicious software (trojan horses, etc.)
A.TIME	The underlying operating system provides reliable time information to the TOE.

4 Security Objectives

4.1 Security Objectives for the TOE

The following table identifies the security objectives to address security concerns that are directly addressed by the TOE:

Table 4: Security Objectives for the TOE

Name	Description
O.ADMIN	The TOE will provide functionality to allow an authorised administrator to manage access and use of security functions, and will ensure that only authorised administrators are able to access such functionality.
O.IDENTIFY	The TOE will uniquely identify all human users, before granting a user access to TOE functions.
O.AUTHEN	The TOE will uniquely authenticate the claimed identity of all human users, before granting a user access to TOE functions.
O.MEDTF	The TOE will mediate the flow of all information from users connected on another network based on network layer and transport layer information as configured by an administrator.
O.DETSPOOF	The TOE will detect source address spoofing attacks and will prevent related unwanted information flows into a protected network.
O.PRIVACY	The TOE will ensure that users on the external network cannot determine the addresses of the users on the internal network as specified by an authorised administrator.
O.AUDGEN	The TOE will ensure that users are accountable for the actions that they conduct by logging security-relevant events.
O.AUDREV	The TOE will provide functionality to support administrators in reviewing the logged information about security relevant events.
O.IDS	The TOE will provide functionality to detect specific attack patterns which indicate a potential security violation.
O.ROLE	The TOE will provide functionality to distinguish between a root administrator with unrestricted management access, administrators with read/write permissions and administrators with read-only permissions.

4.2 Security Objectives for the Environment

The following table identifies security objectives to address security concerns that are addressed by TOE environment:

Table 5: Security Objectives for the Environment

Name	Description
OE.MEDEXP	Those responsible for the TOE must use it in an environment in which the threat of malicious attacks at discovering exploitable vulnerabilities is considered moderate.
OE.NOEVIL	Those responsible for the TOE must assign trustworthy, competent, and trained personnel to the administration of the TOE which follows all administrator guidance.
OE.ONEWAY	Those responsible for the TOE must ensure that information cannot flow between networks connected to the netfence firewall unless it passes through the netfence firewall.
OE.PHIONA	Those responsible for the TOE must ensure that, after the netfence firewall has been installed, administrators use a Management Workstation to administer it, not the system console.
OE.PHYSEC	Those responsible for the TOE must ensure that the netfence firewall is operated in a physically secure environment which prevents access from unauthorised users.
OE.WSSEC	Those responsible for the TOE must ensure that the Management Workstation is operated in an environment which is free of malicious software (trojan horses, etc.)
OE.TIME	The underlying operating system will provide reliable time information to the TOE.

5 IT Security Requirements

5.1 TOE Security Requirements

5.1.1 TOE Security Functional Requirements

The following table identifies the selected TOE security functional requirements. All are drawn from Part 2 of the CC.

Table 6: Security Functional Requirements Overview

Component	Component Name
FAU_GEN.1	Audit data generation
FAU_SAA.1	Potential violation analysis
FAU_SAR.1	Audit review
FAU_SAR.3	Selectable audit review
FDP_IFC.1	Subset information flow control
FDP_IFF.1	Simple security attributes
FIA_AFL.1	Authentication failure handling
FIA_ATD.1	User attribute definition
FIA_SOS.1	Verification of secrets
FIA_UAU.2	User authentication before any action
FIA_UAU.5	Multiple authentication mechanisms
FIA_UAU.7	Protected authentication feedback
FIA_UID.2	User identification before any action
FMT_MOF.1	Management of security functions behaviour
FMT_MSA.1	Management of security attributes
FMT_MSA.3	Static attribute initialisation
FMT_MTD.1	Management of TSF data
FMT_SMF.1	Specification of Management Functions
FMT_SMR.2	Restrictions on Security Roles
FPR_PSE.1	Pseudonymity

Family FAU_GEN Security audit data generation

FAU_GEN.1 Audit data generation

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

a) Start-up and shutdown of the audit functions;
b) All auditable events for the *not specified* level of audit; and
c) [list of events specified in left column of the table below].

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

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

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

Event	Additional Information provided
Administrator identification and authentication	Login name, Peer IP Address
Modifying operations on system attributes	Login name, Peer IP Address, Attribute name
Modifying operations on security attributes	Login name, Peer IP Address, Attribute name
Requests for information flow	Source and Destination Address
Status changes for information flow	Source and Destination Address
IP Spoofing attacks	Source and Destination Address
Port Scans	Source and Destination Address
Address Range Scans	Source and Destination Address
Malformed Datagrams	Source and Destination Address

Family FAU_SAA Security audit analysis

FAU_SAA.1 Potential violation analysis

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

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

a) Accumulation or combination of [requests for information flow] known to indicate a potential security violation;

b) [spoofing attempts, port scans, address range scans].

Family FAU_SAR Security audit review

FAU_SAR.1 Audit review

FAU_SAR.1.1 The TSF shall provide [all administrators] with the capability to read [all audit information] from the audit records.

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

FAU_SAR.3 Selectable audit review

FAU_SAR.3.1 The TSF shall provide the ability to perform *searches*, *sorting* of audit data based on [regular expressions, time intervals].

Family FDP_IFC Information flow control policy

FDP_IFC.1 Subset information flow control

FDP_IFC.1.1 The TSF shall enforce the [Stateful Packet Filter SFP] on [

a) subjects: external IT entities that send and receive information through the TOE to one another;

b) information: packets;

c) operation: pass information

].

Family FDP_IFF Information flow control functions

FDP_IFF.1 Simple security attributes

FDP_IFF.1.1 The TSF shall enforce the [Stateful Packet Filter SFP] based on the following types of subject and information security attributes: [source IP address, destination IP address, transport layer protocol, destination port, network interface the packet arrived from, total number sessions for the source IP address, total number of sessions matched by rule, total number of sessions matched by rule and source IP, tuning parameters].

FDP_IFF.1.2 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: [all the information security attribute values source IP address, destination IP address, transport layer protocol, destination port, are unambiguously permitted by an active rule within the information flow security policy ruleset created by an administrator with full read/write access permission.]. **Rules might be enabled on selected weekdays and day hours, only. The TSF itself can generate dynamic rules with limited activation duration upon administrator request.**

FDP_IFF.1.3 The TSF shall enforce [no additional information flow control SFP rules].

FDP_IFF.1.4 The TSF shall provide the following [action capabilities specified in the ruleset:

- block (ignore the request)
- deny (deny the request and inform the requestor)
- pass (process the request without changing the destination address)
  - pass without changing the source address
  - pass with changing the source address (dynamic or static NAT)
- redirect (process the request with changing the destination address)
  - redirect without changing the source address
  - redirect with changing the source address (dynamic or static NAT)

].

FDP_IFF.1.5 The TSF shall explicitly authorise an information flow based on the following rules: [none].

FDP_IFF.1.6 The TSF shall explicitly deny an information flow based on the following rules: [

- the network interface the packet arrived from mismatches the interface a reply packet would leave from to reach the same source IP address (interface mismatch for reversed path),
- maximum number of sessions for the source IP address is reached,
- maximum number of sessions matched by rule is reached,
- maximum number of sessions matched by rule and source IP is reached,
- packet is malformed,
- maximum number of pending sessions for the source IP is reached

].

Family FIA_AFL Authentication failures

FIA_AFL.1 Authentication failure handling

FIA_AFL.1.1 The TSF shall detect when [a configurable limit (>=3) of] unsuccessful authentication attempts occur related to [one network session is reached]. **A network peer (source IP address) may not exceed a configurable rate of network session initiations. The identification and authentication process may not exceed a configurable time limit.**

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall [terminate the network session].

Family FIA_ATD User attribute definition

FIA_ATD.1 User attribute definition

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual **human** users: [role, passphrase, public RSA key].

Family FIA_SOS Specification of secrets

FIA_SOS.1 Verification of secrets

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet [the following conditions: the passphrase can contain any character except blank, TAB (tab stop), LF (line feed) or CR (carriage return), a minimum length of 6 characters and a maximum of 16 characters is required, at least 1 of them has to be non-alphabetical].

Family FIA_UAU User authentication

FIA_UAU.2 User authentication before any action

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

FIA_UAU.5 Multiple authentication mechanisms

FIA_UAU.5.1 The TSF shall provide [mechanisms: passphrase, RSA private key stored in registry and protected by passphrase, private RSA key stored on smartcard] to support user authentication.

FIA_UAU.5.2 The TSF shall authenticate any user's claimed identity according to the [correct passphrase specified and/or successful RSA key based challenge response check].

FIA_UAU.7 Protected authentication feedback

FIA_UAU.7.1 The TSF shall provide [no information] to the user while the authentication is in progress.

Family FIA_UID User identification

FIA_UID.2 User identification before any action

FIA_UID.2.1 The TSF shall require each **human** user to identify itself before allowing any other TSF-mediated actions on behalf of that user.

Family FMT_MOF Management of functions in TSF

FMT_MOF.1 (1) Management of security functions behaviour

FMT_MOF.1.1 The TSF shall restrict the ability to *disable*, *enable*, *modify the behaviour of* the functions [firewall service, log service and box access service] to [the "root" administrator and a named administrator with read/write permission].

FMT_MOF.1 (2) Management of security functions behaviour

FMT_MOF.1.1 The TSF shall restrict the ability to *determine the behaviour of* the functions [firewall service, log service and box access service] to [all administrators].

Family FMT_MSA Management of security attributes

FMT_MSA.1 (1) Management of security attributes

FMT_MSA.1.1 The TSF shall enforce the [Stateful Packet Filter SFP] to restrict the ability to *add*, *delete*, *modify* the security attributes [source IP address, destination IP address, transport layer protocol, destination port, network interface the packet arrived from, total number sessions for the source IP address, total number of sessions matched by rule, total number of sessions matched by rule and source IP, tuning parameters] to [the “root” administrator and a named administrator with read/write permission].

FMT_MSA.1 (2) Management of security attributes

FMT_MSA.1.1 The TSF shall enforce the [Stateful Packet Filter SFP] to restrict the ability to *query* the security attributes [source IP address, destination IP address, transport layer protocol, destination port, network interface the packet arrived from, total number sessions for the source IP address, total number of sessions matched by rule, total number of sessions matched by rule and source IP, tuning parameters] to [all administrators].

FMT_MSA.3 Static attribute initialisation

FMT_MSA.3.1 The TSF shall enforce the [Stateful Packet Filter SFP] to provide *restrictive* default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow [nobody] to specify alternative initial values to override the default values when an object or information is created.

Family FMT_MTD Management of TSF data

FMT_MTD.1 (1) Management of TSF data

FMT_MTD.1.1 The TSF shall restrict the ability to, *delete*, *modify*, [add] the [information about attached networks, routes, used network device drivers, management IP address, administrative access control lists, log size, log lifetime limits] to [the “root” administrator and a named administrator with read/write permission].

FMT_MTD.1 (2) Management of TSF data

FMT_MTD.1.1 The TSF shall restrict the ability to *clear* the [log files, firewall access cache] to [the “root” administrator and a named administrator with read/write permission].

FMT_MTD.1 (3) Management of TSF data

FMT_MTD.1.1 The TSF shall restrict the ability to [terminate] [administrative sessions, active connections] to [the “root” administrator and a named administrator with read/write permission].

FMT_MTD.1 (4) Management of TSF data

FMT_MTD.1.1 The TSF shall restrict the ability to [activate, deactivate] [dynamic rules] to [the “root” administrator and a named administrator with read/write permission].

FMT_MTD.1 (5) Management of TSF data

FMT_MTD.1.1 The TSF shall restrict the ability to [initiate, schedule, terminate] [traffic traces] to [the “root” administrator and a named administrator with read/write permission].

FMT_MTD.1 (6) Management of TSF data

FMT_MTD.1.1 The TSF shall restrict the ability to *query* the [firewall real time status, firewall access cache, firewall traffic traces, dynamic firewall rule states, list of IP addresses protected by the firewall, log files, subsystem status, network status, process information, memory information, disk status information, administrative sessions, license status, software version, information about attached networks, routes, used network device drivers, management IP address, administrative access control lists, log size, log lifetime limits] to [all administrators].

FMT_MTD.1 (7) Management of TSF data

FMT_MTD.1.1 The TSF shall restrict the ability to [set, change] [any administrator passphrase and/or public key] to [the “root” administrator].

FMT_MTD.1 (8) Management of TSF data

FMT_MTD.1.1 The TSF shall restrict the ability to [change] the [own passphrase and/or public key] to [all administrators].

FMT_MTD.1 (9) Management of TSF data

FMT_MTD.1.1 The TSF shall restrict the ability to [create] a [new administrator] to [the “root” administrator].

FMT_MTD.1 (10) Management of TSF data

FMT_MTD.1.1 The TSF shall restrict the ability to [assign] an [administrator role] to [the “root” administrator].

Family FMT_SMF Specification of Management Functions

FMT_SMF.1 Specification of Management Functions

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

- Configuration of
  - Management IP address
  - Used network device drivers
  - Networks, IP addresses and routes
  - Administrative ACLs
  - Administrative accounts (login name, role)
  - Administrator passphrases and public keys
  - Firewall tuning parameters
  - Firewall ruleset
  - Log size and lifetime limits

- Manage
  - Backup and restore configuration
  - Start and stop subsystems
  - Reboot and halt the system
  - Import license key
  - Perform software updates
  - Terminate administrative sessions
  - Empty log files
  - Reset Firewall Access Cache
  - Terminate active connections
  - Activate or deactivate dynamic rules
  - Initiate or schedule traffic traces
- Visualise
  - Firewall real time status
  - Firewall access cache
  - Firewall traffic traces
  - Dynamic firewall rule states
  - List of IP addresses protected by the firewall
  - Configuration
  - Log Files
  - Subsystem status
  - Network status
  - Process, memory and disk status information
  - Administrative Sessions
  - License status
  - Software Version

].

Family FMT_SMR Security management roles

FMT_SMR.2 Restrictions on security roles

FMT_SMR.2.1 The TSF shall maintain the roles: [
"root" administrator,
named administrator with read/write permission,
named administrator with read-only permission
].

FMT_SMR.2.2 The TSF shall be able to associate users with roles.

FMT_SMR.2.3 The TSF shall ensure that the condition [source IP address is admitted by an administrative ACL] is satisfied.

Family FPR_PSE Pseudonymity

FPR_PSE.1 Pseudonymity

FPR_PSE.1.1 The TSF shall ensure that [external IT entities on the external network] are unable to determine the real user name bound to [external IT entities on the internal network that generate connections to external IT entities on the external network]. **The user name bound to an external IT entity is its IP address.**

FPR_PSE.1.2 The TSF shall be able to provide [unlimited] aliases of the real user name to [external IT entities on the internal network].

FPR_PSE.1.3 The TSF shall *determine an alias for a user* and verify that it conforms to the [range of IP addresses specified by the “root” administrator and a named administrator with read/write permission].

The overall Strength of function claim for the TOE is SOF-high. This claim is applicable to FIA_UAU.2 and FIA_UAU.5.

5.1.2 TOE Security Assurance Requirements

Table 7: TOE Security Assurance Requirements

Component	**Component Name**
ACM_AUT.1	Partial CM automation
ACM_CAP.4	Generation support and acceptance procedures
ACM_SCP.2	Problem tracking CM coverage
ADO_DEL.2	Detection of modification
ADO_IGS.1	Installation, generation, and start-up procedures
ADV_FSP.2	Fully defined external interfaces
ADV_HLD.2	Security enforcing high-level design
ADV_IMP.1	Subset of the implementation of the TSF
ADV_LLD.1	Descriptive low-level design
ADV_RCR.1	Informal correspondence demonstration
ADV_SPM.1	Informal TOE security policy model
AGD_ADM.1	Administrator Guidance
AGD_USR.1	User guidance
ALC_DVS.1	Identification of security measures
ALC_FLR.1	**Basic flaw remediation**
ALC_LCD.1	Developer defined life-cycle model
ALC_TAT.1	Well-defined development tools
ATE_COV.2	Analysis of coverage
ATE_DPT.1	Testing: high-level design
ATE_FUN.1	Functional testing
ATE_IND.2	Independent testing - sample
AVA_MSU.2	Validation of analysis
AVA_SOF.1	Strength of TOE security function evaluation
AVA_VLA.3	**Moderately resistant**

Class ACM Configuration management

ACM_AUT.1 Partial CM automation

Developer action elements:

ACM_AUT.1.1D The developer shall use a CM system.
ACM_AUT.1.2D The developer shall provide a CM plan.

Content and presentation of evidence elements:

ACM_AUT.1.1C The CM system shall provide an automated means by which only authorised changes are made to the TOE implementation representation.

ACM_AUT.1.2C The CM system shall provide an automated means to support the generation of the TOE.

ACM_AUT.1.3C The CM plan shall describe the automated tools used in the CM system.

ACM_AUT.1.4C The CM plan shall describe how the automated tools are used in the CM system.

Evaluator action elements:

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

ACM_CAP.4 Generation support and acceptance procedures

Developer action elements:

ACM_CAP.4.1D The developer shall provide a reference for the TOE.

ACM_CAP.4.2D The developer shall use a CM system.

ACM_CAP.4.3D The developer shall provide CM documentation.

Content and presentation of evidence elements:

ACM_CAP.4.1C The reference for the TOE shall be unique to each version of the TOE.

ACM_CAP.4.2C The TOE shall be labelled with its reference.

ACM_CAP.4.3C The CM documentation shall include a configuration list, a CM plan, and an acceptance plan.
The configuration list shall uniquely identify all configuration items that comprise the TOE.[1]

ACM_CAP.4.4C The configuration list shall describe the configuration items that comprise the TOE.

ACM_CAP.4.5C The CM documentation shall describe the method used to uniquely identify the configuration items.

ACM_CAP.4.6C The CM system shall uniquely identify all configuration items.

ACM_CAP.4.7C The CM plan shall describe how the CM system is used.

ACM_CAP.4.8C The evidence shall demonstrate that the CM system is operating in accordance with the CM plan.

ACM_CAP.4.9C The CM documentation shall provide evidence that all configuration items have been and are being effectively maintained under the CM system.

ACM_CAP.4.10C The CM system shall provide measures such that only authorised changes are made to the configuration items.

ACM_CAP.4.11C The CM system shall support the generation of the TOE.

ACM_CAP.4.12C The acceptance plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.

[1] This element is added as a result of Interpretation 003.

Evaluator action elements:

ACM_CAP.4.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ACM_SCP.2 Problem tracking CM coverage

Developer action elements:

ACM_SCP.2.1D The developer shall provide a list of configuration items for the TOE.[2]

Content and presentation of evidence elements:

ACM_SCP.2.1C The list of configuration items shall include the following: implementation representation; security flaws; and the evaluation evidence required by the assurance components in the ST.[3]

Evaluator action elements:

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

Class ADO Delivery and operation

ADO_DEL.2 Detection of modification

Developer action elements:

ADO_DEL.2.1D The developer shall document procedures for delivery of the TOE or parts of it to the user.

ADO_DEL.2.2D The developer shall use the delivery procedures.

Content and presentation of evidence elements:

ADO_DEL.2.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user’s site.

ADO_DEL.2.2C The delivery documentation shall describe how the various procedures and technical measures provide for the detection of modifications, or any discrepancy between the developer’s master copy and the version received at the user site.

ADO_DEL.2.3C The delivery documentation shall describe how the various procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user’s site.

Evaluator action elements:

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

[2] This element is changed as a result of Interpretation 004.

[3] The content and presentation of evidence elements are replaced as a result of Interpretations 004 and 038.

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

Developer action elements:

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

Content and presentation of evidence elements:

ADO_IGS.1.1C The installation, generation and start-up documentation shall describe the steps necessary for secure installation, generation, and start-up of the TOE.[4]

Evaluator action elements:

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

ADO_IGS.1.2E The evaluator shall determine that the installation, generation, and start-up procedures result in a secure configuration.

Class ADV Development

ADV_FSP.2 Fully defined external interfaces

Developer action elements:

ADV_FSP.2.1D The developer shall provide a functional specification.

Content and presentation of evidence elements:

ADV_FSP.2.1C The functional specification shall describe the TSF and its external interfaces using an informal style.

ADV_FSP.2.2C The functional specification shall be internally consistent.

ADV_FSP.2.3C The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing complete details of all effects, exceptions and error messages.

ADV_FSP.2.4C The functional specification shall completely represent the TSF.

ADV_FSP.2.5C The functional specification shall include rationale that the TSF is completely represented.

Evaluator action elements:

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

ADV_FSP.2.2E The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

ADV_HLD.2 Security enforcing high-level design

Developer action elements:

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

Content and presentation of evidence elements:

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

ADV_HLD.2.2C The high-level design shall be internally consistent.

[4] This element is changed as a result of Interpretation 051.

ADV_HLD.2.3C The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV_HLD.2.4C The high-level design shall describe the security functionality provided by each subsystem of the TSF.

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.

ADV_HLD.2.6C The high-level design shall identify all interfaces to the subsystems of the TSF.

ADV_HLD.2.7C The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.

ADV_HLD.2.8C The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV_HLD.2.9C The high-level design shall describe the separation of the TOE into TSP-enforcing and other subsystems.

Evaluator action elements:

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

ADV_HLD.2.2E The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

ADV_IMP.1 Subset of the implementation of the TSF

Developer action elements:

ADV_IMP.1.1D The developer shall provide the implementation representation for a selected subset of the TSF.

Content and presentation of evidence elements:

ADV_IMP.1.1C The implementation representation shall unambiguously define the TSF to a level of detail such that the TSF can be generated without further design decisions.

ADV_IMP.1.2C The implementation representation shall be internally consistent.

Evaluator action elements:

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

ADV_IMP.1.2E The evaluator shall determine that the least abstract TSF representation provided is an accurate and complete instantiation of the TOE security functional requirements.

ADV_LLD.1 Descriptive low-level design

Developer action elements:

ADV_LLD.1.1D The developer shall provide the low-level design of the TSF.

Content and presentation of evidence elements:

ADV_LLD.1.1C The presentation of the low-level design shall be informal.

ADV_LLD.1.2C The low-level design shall be internally consistent.

ADV_LLD.1.3C The low-level design shall describe the TSF in terms of modules.

ADV_LLD.1.4C The low-level design shall describe the purpose of each module.

ADV_LLD.1.5C The low-level design shall define the interrelationships between the modules in terms of provided security functionality and dependencies on other modules.

ADV_LLD.1.6C The low-level design shall describe how each TSP-enforcing function is provided.

ADV_LLD.1.7C The low-level design shall identify all interfaces to the modules of the TSF.

ADV_LLD.1.8C The low-level design shall identify which of the interfaces to the modules of the TSF are externally visible.

ADV_LLD.1.9C The low-level design shall describe the purpose and method of use of all interfaces to the modules of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV_LLD.1.10C The low-level design shall describe the separation of the TOE into TSP-enforcing and other modules.

Evaluator action elements:

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

ADV_LLD.1.2E The evaluator shall determine that the low-level design is an accurate and complete instantiation of the TOE security functional requirements.

ADV_RCR.1 Informal correspondence demonstration

Developer action elements:

ADV_RCR.1.1D The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided.

Content and presentation of evidence elements:

ADV_RCR.1.1C For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

Evaluator action elements:

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

ADV_SPM.1 Informal TOE security policy model

Developer action elements:

ADV_SPM.1.1D The developer shall provide a TSP model.

ADV_SPM.1.2D The developer shall demonstrate correspondence between the functional specification and the TSP model.

Content and presentation of evidence elements:

ADV_SPM.1.1C The TSP model shall be informal.

ADV_SPM.1.2C The TSP model shall describe the rules and characteristics of all policies of the TSP that can be modelled.

ADV_SPM.1.3C The TSP model shall include a rationale that demonstrates that it is consistent and complete with respect to all policies of the TSP that can be modelled.

ADV_SPM.1.4C The demonstration of correspondence between the TSP model and the functional specification shall show that all of the security functions in the functional specification are consistent and complete with respect to the TSP model.

Evaluator action elements:

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

Class AGD Guidance documents

AGD_ADM.1 Administrator guidance

Developer action elements:

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

Content and presentation of evidence elements:

AGD_ADM.1.1C The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.

AGD_ADM.1.2C The administrator guidance shall describe how to administer the TOE in a secure manner.

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

AGD_ADM.1.4C The administrator guidance shall describe all assumptions regarding user behaviour that are relevant to secure operation of the TOE.

AGD_ADM.1.5C The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.

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

AGD_ADM.1.7C The administrator guidance shall be consistent with all other documentation supplied for evaluation.

AGD_ADM.1.8C The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.

Evaluator action elements:

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

AGD_USR.1 User guidance

Developer action elements:

AGD_USR.1.1D The developer shall provide user guidance.

Content and presentation of evidence elements:

AGD_USR.1.1C The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.

AGD_USR.1.2C The user guidance shall describe the use of user-accessible security functions provided by the TOE.

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

AGD_USR.1.4C The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behaviour found in the statement of TOE security environment.

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

AGD_USR.1.6C The user guidance shall describe all security requirements for the IT environment that are relevant to the user.

Evaluator action elements:

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

Class ALC Life cycle support

ALC_DVS.1 Identification of security measures

Developer action elements:

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

Content and presentation of evidence elements:

ALC_DVS.1.1C The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

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

Evaluator action elements:

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

ALC_DVS.1.2E The evaluator shall confirm that the security measures are being applied.

ALC_FLR.1 Basic flaw remediation

Developer action elements:

ALC_FLR.1.1D The developer shall provide flaw remediation procedures addressed to TOE developers.[5]

[5] This element is modified as a result of Interpretation 094.

Content and presentation of evidence elements:

ALC_FLR.1.1C The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

ALC_FLR.1.2C The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC_FLR.1.3C The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC_FLR.1.4C The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.

Evaluator action elements:

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

ALC_LCD.1 Developer defined life-cycle model

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.

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

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.

ALC_LCD.1.2C The life-cycle model shall provide for the necessary control over the development and maintenance of the TOE.

Evaluator action elements:

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

ALC_TAT.1 Well-defined development tools

Developer action elements:

ALC_TAT.1.1D The developer shall identify the development tools being used for the TOE.

ALC_TAT.1.2D The developer shall document the selected implementation-dependent options of the development tools.

Content and presentation of evidence elements:

ALC_TAT.1.1C All development tools used for implementation shall be well-defined.

ALC_TAT.1.2C The documentation of the development tools shall unambiguously define the meaning of all statements used in the implementation.

ALC_TAT.1.3C The documentation of the development tools shall unambiguously define the meaning of all implementation-dependent options.

Evaluator action elements:

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

Class ATE Tests

ATE_COV.2 Analysis of coverage

Developer action elements:

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

Content and presentation of evidence elements:

ATE_COV.2.1C The analysis of the test coverage shall demonstrate the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.

ATE_COV.2.2C The analysis of the test coverage shall demonstrate that the correspondence between the TSF as described in the functional specification and the tests identified in the test documentation is complete.

Evaluator action elements:

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

ATE_DPT.1 Testing: high-level design

Developer action elements:

ATE_DPT.1.1D The developer shall provide the analysis of the depth of testing.

Content and presentation of evidence elements:

ATE_DPT.1.1C The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design.

Evaluator action elements:

ATE_DPT.1.2E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_FUN.1 Functional testing

Developer action elements:

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

ATE_FUN.1.2D The developer shall provide test documentation.

Content and presentation of evidence elements:

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

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

ATE_FUN.1.3C The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.4C The expected test results shall show the anticipated outputs from a successful execution of the tests.

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

Evaluator action elements:

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

ATE_IND.2 Independent testing - sample

Developer action elements:

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

Content and presentation of evidence elements:

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

ATE_IND.2.2C The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

Evaluator action elements:

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

ATE_IND.2.2E The evaluator shall test a subset of the TSF as appropriate to confirm that the TOE operates as specified.

ATE_IND.2.3E The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

Class AVA Vulnerability assessment

AVA_MSU.2 Validation of analysis

Developer action elements:

AVA_MSU.2.1D The developer shall provide guidance documentation.

AVA_MSU.2.2D The developer shall document an analysis of the guidance documentation.

Content and presentation of evidence elements:

AVA_MSU.2.1C The guidance documentation shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AVA_MSU.2.2C The guidance documentation shall be complete, clear, consistent and reasonable.

AVA_MSU.2.3C The guidance documentation shall list all assumptions about the intended environment.

AVA_MSU.2.4C The guidance documentation shall list all requirements for external security measures (including external procedural, physical and personnel controls).

AVA_MSU.2.5C The analysis documentation shall demonstrate that the guidance documentation is complete.

Evaluator action elements:

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

AVA_MSU.2.2E The evaluator shall repeat all configuration and installation procedures, and other procedures selectively, to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.

AVA_MSU.2.3E The evaluator shall determine that the use of the guidance documentation allows all insecure states to be detected.

AVA_MSU.2.4E The evaluator shall confirm that the analysis documentation shows that guidance is provided for secure operation in all modes of operation of the TOE.

AVA_SOF.1 Strength of TOE security function evaluation

Developer action elements:

AVA_SOF.1.1D The developer shall perform a strength of TOE security function analysis for each mechanism identified in the ST as having a strength of TOE security function claim.

Content and presentation of evidence elements:

AVA_SOF.1.1C For each mechanism with a strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the minimum strength level defined in the PP/ST.

AVA_SOF.1.2C For each mechanism with a specific strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the specific strength of function metric defined in the PP/ST.

Evaluator action elements:

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

AVA_SOF.1.2E The evaluator shall confirm that the strength claims are correct.

AVA_VLA.3 Moderately resistant

Developer action elements:

AVA_VLA.3.1D The developer shall perform a vulnerability analysis.

AVA_VLA.3.2D The developer shall provide vulnerability analysis documentation.[6]

Content and presentation of evidence elements:

AVA_VLA.3.1C The vulnerability analysis documentation shall describe the analysis of the TOE deliverables performed to search for ways in which a user can violate the TSP.

AVA_VLA.3.2C The vulnerability analysis documentation shall describe the disposition of identified vulnerabilities.

AVA_VLA.3.3C The vulnerability analysis documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.

AVA_VLA.3.4C The vulnerability analysis documentation shall justify that the TOE, with the identified vulnerabilities, is resistant to obvious penetration attacks.

AVA_VLA.3.5C The vulnerability analysis documentation shall show that the search for vulnerabilities is systematic.[7]

[6] The original two elements are changed as a result of Interpretation 051.

[7] The original three elements are replaced by five as a result of Interpretation 051.

Evaluator action elements:

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

AVA_VLA.3.2E The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure the identified vulnerabilities have been addressed.

AVA_VLA.3.3E The evaluator shall perform an independent vulnerability analysis.

AVA_VLA.3.4E The evaluator shall perform independent penetration testing, based on the independent vulnerability analysis, to determine the exploitability of additional identified vulnerabilities in the intended environment.

AVA_VLA.3.5E The evaluator shall determine that the TOE is resistant to penetration attacks performed by an attacker possessing a moderate attack potential.

5.2 Security Requirements for the IT Environment

Family FPT_STM Time stamps

FPT_STM.1 Reliable time stamps

FPT_STM.1.1 The **IT environment** shall be able to provide **reliable time stamps**.

6 TOE Summary Specification

6.1 TOE Security Functions

6.1.1 Security Administration

The netfence firewall system is managed remotely with a windows application called **phiona**, which is available for Windows NT 4.0, Windows 2000 and Windows XP. For exchanging data between the firewall system and the phiona a socket connection is established on port 800. When connecting to the system the administrator can either choose the “root” administrator with unlimited permissions or a named administrator with restricted access. Named administrators can have either read/write permission or read-only permission. Once an administrator has gained access he/she can perform configurative actions or visualise the present status. Configurative actions are protected by a locking mechanism which prevents simultaneous write operations on attributes. The administrative GUI is divided into four topics:

- **System Configuration**
  Management of system attributes
- **Firewall Control**
  Management and configuration of the security attributes. Visualisation of the firewall status.
- **System Control**
  Management and visualisation of the system status.
- **Log Viewer**
  Management and visualisation of log data.

The phiona is used by the administrator to perform the following tasks:

- **Configuration of**
  - **Management IP address**
    This IP address is used to gain access to the security administration. The system will create a listening TCP socket on port 800 for this address.
  - **Used network device drivers**
    Depending on the used network interface card (NIC) network drivers may be chosen.
  - **Networks, IP addresses and routes**
    The firewall needs information about the networks attached to the system in order to perform controlled propagation between them. In particular a so called “default route” is required when information flow from and to the internet is required. The firewall itself may be a “next hop” for other network components. Therefore a list of IP addresses in addition to the management IP can be configured to be activated on the system.
  - **Administrative ACLs**
    In order to restrict administration to a controlled number of network nodes (IP addresses) ACLs can be used to specify a list of IP addresses and networks which may gain access to the administrative functions.

- **Named administrators, administrator passphrases and public keys**

  Before gaining access to the system the administrators have to authenticate themselves by providing a login name and a passphrase or/and passing a challenge response scheme. The passphrases and the public RSA keys for the administrators can be configured.

- **Firewall tuning parameters**

  The firewall needs a set of parameters which will affect the performance and certain security policies of the firewall.

- **Firewall ruleset**

  The firewall rule set defines the security policy when controlling the information flow through the firewall.

- **Log size and lifetime limits**

  Log files can be limited to hold event data for a specified period of time and/or up to a specified maximum file size.

- **Manage**

  - **Backup and restore configuration**

    The entire configuration including security and system attributes can be saved into a file which can be used for later recovery.

  - **Start and stop subsystems**

    The netfence firewall system contains three subsystems (firewall service, log service and box access service) which can be started, stopped and restarted through the phiona**.**

  - **Reboot and halt the system**

  - **Import license key**

  - **Perform software updates**

  - **Terminate administrative sessions**

    Since the system allows simultaneous access for multiple administrative sessions, which protect their actions through a locking mechanism, the phiona provides the possibility to terminate active session in order to break locks.

  - **Empty log files**

  - **Reset Firewall Access Cache**

    The firewall access cache contains cumulative information about the latest information flow activity. The administrator can decide to restart the information gathering, in order to focus on imminent events.

  - **Terminate active connections**

    If an information flow can be assigned to a session (TCP) the administrator can actively terminate the established session between to network nodes.

  - **Activate or deactivate dynamic rules**

    The firewall ruleset allows to create dynamic rules which can be dynamically activated without making a change to the contents of the ruleset. Furthermore these dynamic rules can have a limited duration of activation. Through the phiona these rules can be activated and when doing so the duration of the activation can be specified.

- **Initiate or schedule traffic traces**

  The firewall system allows activation of network traces as a helpful tool for troubleshooting. Traces can be initiated by toggling the trace mode on an active network session or by specifying trace conditions which will turn on the trace mode on further traffic meeting the conditions.

- **Visualise**

  - **Firewall real time status**

    A real time status of all information flow activity is displayed. The status shown includes information like source and destination IP addresses, network ports used and time stamps of initiation and last activity.

  - **Firewall access cache**

    A cumulative list of the recent network activities is shown. The list has a limited number of entries and recycles older entries for newer ones if required.

  - **Firewall traffic traces**

    If a network connection was selected for a trace the data gathered by the trace can be viewed.

  - **Dynamic firewall rule states**

    Dynamic firewall rules are stateful. They can either be inactive or active for a limited period of time. The state and the expiration time of these rule is displayed.

  - **List of IP addresses protected by the firewall**

    The firewall keeps a list of protected IP addresses for licensing purposes.

  - **Configuration**

    Shows the configured system and the security attributes

  - **Log Files**

    The different subsystems maintain a log file recording all relevant events. These log files can be reviewed. For reviewing, a filter tool allows to reduce the data shown to events containing a specified keyword. Furthermore a search tool provides quick navigation within a displayed data block.

  - **Subsystem status**

    The status of the three subsystems (firewall, log, box access) is visualised.

  - **Network status**

    Display of all network devices, active IP addresses, network routes, learned MAC addresses.

  - **Process, memory and disk status information**

  - **Administrative Sessions**

    All current administrative session are displayed. Information on the originating peer, the administrator, the functions used and the time stamp of initiation are displayed.

  - **License status**
  - **Software Version**

6.1.2 Identification and Authentication

Access to the security administration is only gained if the following conditions are met:

a) The administrator provides a login name to identify him/herself. The login name provided is contained in the configured list of known login names.

And

b1) The administrator provides the proper passphrase for the chosen login name.

And/Or

b2) The administrator is in possession of the private RSA key which matches the public key stored on the system. This is checked by a challenge response method: The system generates a random string which it encrypts with the public key stored on the system. Upon request this encrypted string is sent to the phiona.
If the administrator is in possession of the correct private RSA key, meaning that this private key has been imported into the applications registry, the client side application is able to decrypt the random string and can send back the original string to the system as proof of authentication.

Comment: Private RSA keys are not stored in clear text in the registry, but are passphrase protected. Private RSA key can also reside on smartcards.

And

c) The IP address used for the network connection matches a specified ACL. This limits access to selected workstations or networks.

And

d) The time the process for a) identification and b) authentication takes does not exceed a configurable time limit.

And

e) Since the last unsuccessful identification and authentication attempt from the peer IP address used a configurable amount of time has passed (limitation of the rate of identification and authentication attempts).

The following policies are enforced for the identification and authentication of administrators:

- login name selection policy:
  - a valid login name has at least 6 characters and at most 32 characters
  - a valid login name consists of alphabetic characters of the english alphabet and digits
- passphrase selection policy:
  - a valid passphrase has at least 6 characters and a maximum of 16 characters
  - a valid passphrase contains at least 1 non alphabetic character
  - the passphrase can contain any character except blank, TAB (tab stop), LF (line feed) or CR (carriage return)
- passphrase change policy:
  - an administrator must provide the old passphrase prior to the passphrase change.

  - the minimum duration between passphrase changes is zero
  - the maximum duration between passphrase changes is infinite
  - named administrators may only change their own passphrases
  - root may change passphrases of named administrators without providing the old passphrase

- public key change policy:
  - an administrator must pass a challenge response scheme for the old public key prior to the key change.
  - the minimum duration between key changes is zero
  - the maximum duration between key changes is infinite
  - named administrators can only change their key
  - root can change the public keys of named administrators without having possessing the respective previous private keys
- identification and authentication policies
  - no information about TOE attributes is provided prior to a successful identification and authentication.
  - the identification and authentication process may not exceed a configurable timelimit.
  - per administrative network session only a configurable number of identification and authentication attempts (>= 3) are allowed.
  - A Network Peer (Source IP address) may not exceed a configurable rate of network session initiations.

This security function is realised by probabilistic/permutational mechanisms (authentication secrets). The strength of this security function is SOF-High.

6.1.3 Information Flow Control

The aim of information flow control is to enforce a security policy which is defined by the firewall ruleset. With this ruleset the firewall administrator can decide to allow specific network traffic from and to certain nodes while blocking others.
The ruleset contains an ordered list of rules which define what action is to be taken if information flow is requested and certain conditions are met. The conditions for a rule to match and to trigger the specified action are built from the following information:

- Protocol (i.e. TCP, UDP, ICMP,GRE, ESP, etc.)
- Source IP address
- Destination IP address
- Destination Port (if applicable, Network Service: SMTP, POP, HTTP, etc.)

For each rule an action can be chosen which then controls how the information flow request will be treated: Possible actions are:

- Block (Ignore the request)
- Deny (Deny the request and inform the requestor)
- Pass (Process the request)
  - Pass without changing the source address
  - Pass with changing the source address (dynamic or static NAT)
- Redirect (Process the request and change the destination address)
  - Redirect without changing the source address
  - Redirect with changing the source address (dynamic or static NAT)

If a matching rule is found and the action still permits propagation of traffic (is not blocked) further checks based on the following information are performed:

- Network interface the datagram arrived from and network interface the datagram would have to leave from to reach its source (reverse path check as part of IP spoofing check)
- Total number of active sessions or pseudo sessions for the source IP address (Limit number of session per source IP).
- Total number of active sessions or pseudo sessions matched by the rule (Limit number of session per rule)
- Number of active sessions or pseudo sessions matched by the rule for the source IP address (Limit number of session per rule and source IP)
- Current system time (to see whether or not the connection initiation falls into the configured allowed time window, i.e. an access time restriction)

Information flow control is implemented by two coexisting software components:

- ***Application Controlled Packet Forwarder***

  This method acts on IP packets (datagrams). For each datagram a decision based upon the configured firewall rules is made to control traffic between nodes. If the carried network protocol allows assignment of datagrams to sessions (e.g. TCP) the state of these sessions is recorded (*stateful*) and is taken into account in the decision process. For connectionless protocols like UDP the concept of a *pseudo session* is established, where it is still possible to assign datagrams to pseudo sessions. Although no explicit session establishment nor termination can be tracked, the session start is implicitly defined as the start of network activity of a source IP address (initiator) to a destination IP (responder). The session end is defined as the end of a configurable period of network inactivity. This way the decision process is divided into two types: A slow decision for packets not belonging to an prevalidated session by consulting the firewall ruleset and a fast decision based upon a lookup in a table of all prevalidated sessions. The pseudo sessions are again split into two types: -*Proxy Capable Pseudo Sessions* that carry additional information about the initiator (e.g. a source port number) which allows to assign a datagram to a pseudo session without using the IP address and – *Non Proxy Capable Pseudo Sessions* which can be identified solely by the pair of source and destination IP address.

- ***Transparent Application Proxy***

  This method controls data streams (TCP) from one network node to another. Based on the address (IP-Address and Port Number) of the initiating node (source) and the responding node (destination) establishments of such data streams can be allowed or denied. The system acts as a session endpoint for the source node and as a session initiator for the destination, controlling and analysing the flow and its content. This way even if the stream between source and destination is separated by the intervening firewall the communicating nodes do not have to have knowledge of the firewall inbetween (t*ransparency*) to perform an authorised data exchange.

Each of these two software components implements a different transport method for IP network traffic. For each of these two transport methods (ACPF and TAP) two operation modes, inbound mode and outbound mode, are provided. The inbound method is provided to shield protected network nodes from TCP-SYN attacks performed across the TOE which aim at resource exhaustion of the protected node or nodes. To this end the TOE will first expect the three-way TCP handshake with the inititiating source node to be completed before attempting to connect the protected target network node. In outbound mode an incoming TCP-SYN packet is immediately passed on to the target network node.

Network Session Types:

Type	Method	Dynamic NAT	Protocols
Real Sessions	ACPF and TAP	Yes	TCP
Proxy Capable Pseudo Session	ACPF	Yes	UDP, ICMP-ECHO
Non Proxy Capable Pseudo Session	ACPF	No	all others

Functionalities of the information flow control modules:

- Application controlled packet forwarder
  - Stateful packet forwarding (Session aware)
  - Static NAT (1 to 1 source or destination address translation),
    Dynamic NAT (masquerading, n to 1 address translation for proxy capable sessions)
- Transparent Application Proxy
  - Address and Port Redirection
  - Proxying
  - Application plugins for (ftp, sqlnet, rsh)
- Common Security Features
  - Packet Analysis
    Check Packet for proper checksums, invalid TCP flag combinations, length encoding and misused IP options.
  - Fragmentation attack protection
  - IP Spoofing Protection
    Checks if incoming- (receive datagram) and outgoing interface (reverse path lookup) match. Furthermore, TCP connection requests can be pre-established locally by the firewall before any datagram is forwarded to the destination (inbound mode providing SYN-protection)
  - Access Time Restriction
    Enables firewall rules only on selected weekdays and day hours.
  - Dynamic Rules with limited activation duration.
  - Logging of network activities.

6.1.4 Privacy

To hide information about internal networks the firewall is capable of masquerading IP addresses when propagating traffic to other networks or to the internet. For the packet forwarder this is achieved by Dynamic NAT where the internal source IP addresses of a datagram is rewritten to a

common external address. In order to reassign reverse traffic for the session a random port number is chosen as source port, which uniquely defines the session.
When doing transparent proxying internal addresses are hidden by choosing an external address as source when connecting to the destination. To hide information about the TCP stacks used by hosts on the internal network the firewall is capable of using transparent application proxying (TAP) which will only expose the TCP stack used by the firewall system itself to both source and destination. Thus for TAP transport method no information about the TCP stack used by the destination is available to the source and vice versa.

6.1.5 Security Audit

Logging into files is performed for the following events:

- Administrative login attempts (successful and unsuccessful)
  Information provided: Login Name, Peer IP Address
- Modifying administrative actions
  Information provided: Login Name, Peer IP Address and Attribute name or Attribute group
- Allowed information flow
  Information provided: Source and Destination IP Address, Source Network Device, Protocol specific Information (TCP,UDP port numbers)
- Blocked information flow
  Information provided: Source and Destination IP Address, Source Network Device, Protocol specific Information (TCP,UDP port numbers) and a human readable reason why the flow request was blocked.
- Information flow state changes
  Information provided: Source and Destination IP Address, Source Network Device, Protocol specific Information (TCP,UDP port numbers) and the new state of the information flow (ie. Termination).
- IP Spoofing attempts
  Information provided: Source and Destination IP Address, Source Network Device, Protocol specific Information (TCP,UDP port numbers) and a human readable reason indicating the spoofing attempt.
- Limit exceeded events (too many connections per source)
  Information provided: Source and Destination IP Address, Source Network Device, Protocol specific Information (TCP,UDP port numbers) and the current number of connections.
- Port Scans
  Information provided: Source IP Address and the number of different blocked information flows within 10 seconds for this address.
- Address Range Scans
  Information provided: Source IP Address and the number of different blocked information flows within 10 seconds for this address.

- All subsystem startups and stops
  Information provided: Name of the started or stopped subsystem or subsystem component.

Each event is issued with a timestamp (year, month, day, hour, minute and second)

The log files can be reviewed using the Log facility of the phiona. Access to the log files is only granted for previously authenticated administrators.

6.2 Assurance Measures

Phion IT GmbH has appropriate procedures in place to meet the assurance requirements as specified in Chapter 5.1.2. The following table identifies the documentation which provides evidence that the requirements are met:

Table 8: TOE Security Assurance Requirements

Component	**Component Name**	**Measure described in document**
ACM_AUT.1	Partial CM automation	Netfence Configuration Management The CM System is designed to support users with an explicit role assignment that allows to limit the access to CM entities and controls the CM operations available to the user. These roles are enforced by a central command dispatching tool that also keeps a protocol of all CM activity. The CM system is documented and meets the requirements for ACM_AUT.1
ACM_CAP.4	Generation support and acceptance procedures	Netfence Configuration Management The CM system uses a versioning system to maintain a history of all CE entities and tags each entity with a version number. Tests are performed for known version tags and for a complete build of the software (ie. The CDROM) the version tags of each component is known. The generation support and the acceptance procedures of fhe CM system are documented and meet the requirements for ACM_CAP.4
ACM_SCP.2	Problem tracking CM coverage	Netfence Configuration Management The CM system holds CM entities that contain problem reports for testet software modules. The problem tracking ascpect of the CM system is documented and meets the requirements for ACM_SCP.2
ADO_DEL.2	Detection of modification	Netfence Delivery Procedures Phion distributes a checksum of the CDROM that is used to deliver the software. This allows the user to verify that the content of the CDROM (ie. the netfence software) is authentic. The netfence delivery procedure is documented and meets the requirement for ADO_DEL.2.
ADO_IGS.1	Installation, generation, and start-up procedures	Netfence User Guidance The netfence user guidance contains a part that describes how the TOE is securely installed from the delivered CDROM. This involes installation and initial setup. The netfence user guidance meets the requirements for ADO_IGS.1.
ADV_FSP.2	Fully defined external interfaces	Netfence Functional Specification The FSP document identifies all subsystems and describes the external and internal interfaces of the TOE. The document meets the requirements for ADV_FSP.2.
ADV_HLD.2	Security enforcing high-level design	Netfence High Level Design The HLD document describes the purpose and functionality of all subsystems of the TOE. All interfaces that used for the interoperability of the subsystem are documented. The document meets the requirements for ADV_HLD.2.
ADV_IMP.1	Subset of the implementation of the TSF	Netfence Source Code

Component	Component Name	Measure described in document
		Phion will deliver the netfence source code to the evaluator
ADV_LLD.1	Descriptive low-level design	Netfence Low Level Design The LLD document describes the software modules of the sub-system of the TOE. It provides all information that is needed to implement the software modules. The document meets the requirements for ADV_LLD.1.
ADV_RCR.1	Informal correspondence demonstration	Netfence Correspondence Documentation The documents show the correspondence between the following documents : ST with FSP FSP with HLD HLD with LLD LLD with IMP The documents show that the specification are mutually consistent and complete. The documents meet the requirement for ADV_RCR.1
ADV_SPM.1	Informal TOE security policy model	Netfence Security Policy Model The SPM document describes the security policies that are implied by the functional specification. The document meets the requirement for ADV_SPM.1
AGD_ADM.1	Administrator Guidance	Netfence User Guidance The netfence user guidance allows an administrator to securely install and maintain the TOE. It describes startup, TOE configuration and visualization issues used for maintaining the system securely. The document meets the requirements for AGD_ADM.1.
AGD_USR.1	User guidance	Netfence User Guidance The netfence user guidance allows an administrator to securely install and maintain the TOE. It describes startup, TOE configuration and visualization issues used for maintaining the system securely. The document meets the requirements for AGD_USR.1.
ALC_DVS.1	Identification of security measures	Netfence Security Measures The DVS document describes the security measures that are taken to provide a confidential and integer TOE design and implementation. The document meets the requirements for ALC_DVS.1.
ALC_FLR.1	Basic flaw remediation	Netfence Flaw Remediation The CM system supports a workflow that allows to track a software flaw by assigning the reported flaw to a software module referencing the flaw report. Fixes of the software module result into a new version of the module that can be delivered as a hotfix accompanied with documentation that describe the purpose of the hotfix. The CM system implementation meets the requirements for ALC_FLR.1.
ALC_LCD.1	Developer defined life-cycle model	Netfence Development Procedures The CM system supports a workflow that allows to change software modules. These changes result into new version numbers of the affected modules and are referenced against documents that

Component	Component Name	Measure described in document
		describe the reason for the change.The netfence development procedures meet the requirements for ALC_LCD.1.
ALC_TAT.1	Well-defined development tools	Netfence Development Procedures The CM system uses a well defined set of software development tools. For each tool documention is available to the developer. The netfence development procedures meet the requirements for ALC_TAT.1.
ATE_COV.2	Analysis of coverage	Netfence Test Documentation The netfence test documentation contains test results for all subsystems of the TOE as described in the HLD document. Each test is assigned to the subsystems and interfaces involved. Each subsystem and interface is at least covered by one test procedure. The document meets the requirements for ATE _COV.2
ATE_DPT.1	Testing: high-level design	Netfence Test Documentation The netfence test documentation contains test results for all subsystems of the TOE as described in the HLD document. Each test is assigned to the subsystems and interfaces involved. Each subsystem and interface is at least covered by one test procedure. The document meets the requirements for ATE _DPT.1
ATE_FUN.1	Functional testing	Netfence Test Documentation The netfence test documentation contains test results for all functions specified in the FSP and ST documents. Each function is at least covered by one test procedure. The document meets the requirements for ATE _FUN.1
ATE_IND.2	Independent testing - sample	Phion will deliver the TOE suitable for testing to the evaluator.
AVA_MSU.2	Validation of analysis	Netfence Misuse Analysis The analysis will show that the information provided in the user guidance document allows the administrator to securely install and maintain the TOE. It shows that the user guidance does not contain any misleading, unreasonable or conflicting statements that will lead to an unsecure operation. The analysis meets the requirements for AVA_MSU.2.
AVA_SOF.1	Strength of TOE security function evaluation	Netfence Strength of Function Analysis The analysis shows that the strength of function claim SOF-high is appropriate for this type of TOE and the selected assurance level. The analysis meets the requirements for AVA_SOF.1.
AVA_VLA.3	Moderately resistant	Netfence Vulnerability Analysis The document identifies all known vulnerabilities of the TOE and refers to the corresponding passages in the user guidance that will discourage the user to operate the TOE under these circumstances. The document meets the requirements for AVA_VLA.3.

7 Protection Profile Claims

There are no Protection Profile Claims.

8 Rationale

This section identifies the rationale for the adequacy of the security objectives to counter the identified threats and to meet the identified policies and assumptions. It identifies the rationale for the adequacy of the security functional requirements and the security assurance requirements in meeting these objectives. It identifies the rationale for the adequacy of the TOE realisation in meeting these requirements.

8.1 Security Objectives Rationale

The following table traces all security objectives for the TOE back to aspects of the threats to be countered by the TOE and to policies to be met by the TOE:

Table 9: Tracing TOE Objectives to Threats and Policies

Threats and Policies	Objectives
T.NOAUTH	O.ADMIN, O.IDENTIFY, O.AUTHEN
T.ASPOOF	O.DETSPOOF
T.MEDTF	O.MEDTF
T.PRIVACY	O.PRIVACY
T.NODETECT	O.IDS
P.AUDACC	O.AUDGEN, O.AUDREV
P.ROLE	O.ROLE

Countering T.NOAUTH

O.ADMIN	This security objective assures, that only authorised administrators are able to access the functionality to manage access and use of security functions. This objective contributes to counter threat T.NOAUTH.
O.IDENTIFY	This security objective is necessary to counter threat T.NOAUTH because it requires that human users are identified before they can access the TOE functions.
O.AUTHEN	This security objective is necessary to counter threat T.NOAUTH because it requires that the claimed identity of a human user is validated before access to TOE functions is granted.

This threat is fully countered by achieving these three objectives. They require that security functions of the TOE can only be accessed by authorised administrators (O.ADMIN) and that these human users are identified (O.IDENTIFY) and authenticated (A.AUTHEN) by the TOE before access to the security functions is granted.

Countering T.ASPOOF

O.DETSPOOF	This security objective is necessary and sufficient to counter threat T.ASPOOF by preventing unwanted information flow caused by an attacker using a spoofed source address.

Countering T.MEDTF

O.MEDTF This security objective is necessary and sufficient to counter threat T.MEDTF because it requires that all information flow through the TOE is mediated by the TOE.

Countering T.PRIVACY

O.PRIVACY This security objective is necessary and sufficient to counter threat T.PRIVACY because it requires that addresses of users on the internal network are hidden to users on the external network.

Covering P.AUDACC

O.AUDGEN This security objective is necessary to cover policy P.AUDACC because it requires that security relevant events are logged.

O.AUDREV This security objective is necessary to cover policy P.AUDACC because it requires that logged information can be properly reviewed.

This policy is fully covered by these two objectives because security relevant events will be logged (O.AUDGEN) and functionality will be provided to support administrators in reviewing the logged information (O.AUDREV).

Countering T.NODETECT

O.IDS This security objective is necessary and sufficient to counter threat T.NODETECT because it requires that specific attack patterns are detected by the TOE.

Covering P.ROLE

O.ROLE This security objective is necessary and sufficient to cover policy P.ROLE because it mandates that administrators with different privileges can be distinguished.

The following table traces all security objectives for the environment back to aspects of the threats to be countered by the environment and to assumptions about the environment:

Table 10: Tracing Security Objectives for the Environment to Threats and Assumptions

Threats and Assumptions	Objectives
A.MEDEXP	OE.MEDEXP
A.NOEVIL	OE.NOEVIL
A.ONEWAY	OE.ONEWAY
A.PHIONA	OE.PHIONA
A.PHYSEC	OE.PHYSEC
A.WSSEC	OE.WSSEC
A.TIME	OE.TIME

OE.NOEVIL This security objective completely transforms assumption A.NOEVIL.

OE.MEDEXP This security objective completely transforms assumption A.MEDEXP.

OE.ONEWAY This security objective completely transforms assumption A.ONEWAY.

OE.PHIONA This security objective completely transforms assumption A.PHIONA.

OE.PHYSEC This security objective completely transforms assumption A.PHYSEC.

OE.WSSEC This security objective completely transforms assumption A.WSSEC.

OE.TIME This security objective completely transforms assumption A.TIME.

8.2 Security Requirements Rationale

8.2.1 Traceability and Suitability of Functional Requirements

The following table traces all TOE security functional requirements back to aspects of the security objectives for the TOE:

Table 11: Security Objectives for the TOE and TOE Security Functional Requirements

Objectives for the TOE	TOE Security Functional Requirements
O.ADMIN	FMT_MOF.1, FMT_MSA.1, FMT_MSA.3, FMT_MTD.1, FMT_SMF.1, FMT_SMR.2
O.IDENTIFY	FIA_UID.2
O.AUTHEN	FIA_AFL.1, FIA_ATD.1, FIA_SOS.1, FIA_UAU.2, FIA_UAU.5, FIA_UAU.7
O.ROLE	FMT_SMR.2, FMT_MOF.1, FMT_MSA.1, FMT_MTD.1, FIA_ATD.1
O.MEDTF	FDP_IFC.1, FDP_IFF.1, FMT_MSA.3
O.DETSPOOF	FDP_IFF.1
O.PRIVACY	FPT_PSE.1
O.AUDGEN	FAU_GEN.1
O.AUDREV	FAU_SAR.1, FAU_SAR.3
O.IDS	FAU_SAA.1

O.ADMIN The functional requirements listed below in its entity are suitable to completely meet this objective. This is achieved by specifying the security management functions (FMT_SMF.1), the restrictions on usage of administrational roles (FMT_SMR.2), the administration related abilities (FMT_MOF.1, FMT_MSA.1, and the various iterations of FMT_MTD.1), and the providence of static default values for security attributes (FMT_MSA.3).

FMT_MOF.1 This component contributes to meeting objective O.ADMIN by restricting the ability to determine the behaviour, modify the behaviour, disable, enable the functions firewall service, log service, and box access service of the TOE to properly authorised administrators.

FMT_MSA.1 This component contributes to meeting objective O.ADMIN by restricting the ability to add, delete, modify, query the security attributes of the TOE to properly authorised administrators.

FMT_MSA.3 This component contributes to meeting objective O.ADMIN by assisting the administration providing static default values for security attributes.

FMT_MTD.1 (1) This component contributes to meeting objective O.ADMIN by restricting the ability to add, delete, and modify the information about attached networks, routes, used network device drivers, management IP address, administrative access control lists, log size, and log lifetime limits to properly authorised administrators.

FMT_MTD.1 (2) This component contributes to meeting objective O.ADMIN by restricting the ability to clear log files and the firewall access cache to properly authorised administrators.

FMT_MTD.1 (3) This component contributes to meeting objective O.ADMIN by restricting the ability to terminate administrative, active connections to properly authorised administrators.

FMT_MTD.1 (4) This component contributes to meeting objective O.ADMIN by restricting the ability to activate, and deactivate dynamic rules to properly authorised administrators.

FMT_MTD.1 (5) This component contributes to meeting objective O.ADMIN by restricting the ability to initiate, schedule, and terminate traffic traces to properly authorised administrators.

FMT_MTD.1 (6) This component contributes to meeting objective O.ADMIN by restricting the ability to query firewall real time status, firewall access cache, firewall traffic traces, dynamic firewall rule states, list of IP addresses protected by the firewall, log files, subsystem status, network status, process information, memory information, disk status information, administrative sessions, license status, software version, information about attached networks, routes, used network device drivers, management IP address, administrative access control lists, log size, and log lifetime limits to properly authorised administrators.

FMT_MTD.1 (7) This component contributes to meeting objective O.ADMIN by restricting the ability to set, change any administrator passphrase and/or public key to the "root" administrator.

FMT_MTD.1 (8) This component contributes to meeting objective O.ADMIN by allowing each administrator (except the "root" administrator) to only change the own passphrase and/or public key.

FMT_MTD.1 (9) This component contributes to meeting objective O.ADMIN by restricting the ability to create a new administrator to the “root” administrator.

FMT_MTD.1 (10) This component contributes to meeting objective O.ADMIN by restricting the ability to assign an administrator role to the “root” administrator.

FMT_SMF.1 This component contributes to meeting objective O.ADMIN by specifying the security management functions provided by the TOE.

FMT_SMR.2 This component contributes to meeting objective O.ADMIN by restricting the source IP addresses admitted for administration.

O.IDENTIFY The functional requirements listed below in its entity are suitable to completely meet this objective. This is achieved by identifying all human users before any other TSF-mediated actions (FIA_UID.2)

FIA_UID.2 This component contributes to meeting objective O.IDENTIFY by assuring that human users are identified before allowing any other TSF-mediated actions on behalf of that user.

O.AUTHEN The functional requirements listed below in its entity are suitable to completely meet this objective. This is achieved by authenticating all human users before any other TSF-mediated actions (FIA_UAU.2). The strength of the authentication function is enhanced by limiting unsuccessful authentication attempts (FIA_AFL.1), by assuring that authentication secrets meet specified characteristics (FIA_SOS.1), and by preventing an attacker to gain information during an authentication attempt (FIA_UAU.7). Several authentication mechanisms are provided (FIA_UAU.5) meeting different user needs and the related security attributes of an individual human user are maintained (FIA_ATD.1).

FIA_AFL.1 This component contributes to meeting objective O.AUTHEN by assuring that unsuccessful authentication attempts are limited.

FIA_ATD.1 This component contributes to meeting objective O.AUTHEN by maintaining the security attributes passphrase and public RSA key of an individual human user.

FIA_SOS.1 This component contributes to meeting objective O.AUTHEN by assuring that authentication secrets meet specified characteristics.

FIA_UAU.2 This component contributes to meeting objective O.AUTHEN by assuring that human users are authenticated before allowing any other TSF-mediated actions on behalf of that human user.

FIA_UAU.5 This component contributes to meeting objective O.AUTHEN by providing several authentication mechanisms meeting different human user and administration needs.

FIA_UAU.7 This component contributes to meeting objective O.AUTHEN by preventing an attacker to gain information during an authentication attempt.

O.ROLE The functional requirements listed below in its entity are suitable to completely meet this objective. This is achieved by maintaining three administrator roles (FMT_SMR.2), by associating the management of functions (FMT_MOF.1) in the TSF, management of security attributes (FMT_MSA.1), and management of TSF data (FMT_MTD.1) to the related administrator roles and by maintaining the security attribute role of an individual human user (FMT_ATD.1).

FMT_SMR.2 This component contributes to meeting objective O.ROLE by maintaining three administrator roles.

FMT_MOF.1 This component contributes to meeting objective O.ROLE by associating the management of functions in the TSF to the related administrator roles.

FMT_MSA.1 This component contributes to meeting objective O.ROLE by associating the management of security attributes to the related administrator roles.

FMT_MTD.1 This component contributes to meeting objective O.ROLE by associating the management of TSF data to the related administrator roles.

FIA_ATD.1 This component contributes to meeting objective O.ROLE by maintaining the security attribute role of an individual human user.

O.MEDTF The functional requirements listed below in its entity are suitable to completely meet this objective. This is achieved by identifying the entities involved in the Stateful Packet Filter SFP (FDP_IFC.1), the relevant attributes of the users sending and receiving the information in the Stateful Packet Filter SFP as well as the relevant attributes for the information itself (FDP_IFF.1), and by providing restrictive default initial values (FMT_MSA.3).

FDP_IFC.1 This component contributes to meeting objective O.MEDTF by identifying the entities involved in the Stateful Packet Filter SFP.

FDP_IFF.1 This component contributes to meeting objective O.MEDTF by identifying the relevant attributes of the users sending and receiving the information in the Stateful Packet Filter SFP as well as the relevant attributes for the information itself.

FMT_MSA.3 This component contributes to meeting objective O.MEDTF by assisting the administration providing static default values for security attributes.

O.DETSPOOF Functional requirement FDP_IFF.1 is suitable to completely meet this objective.

FDP_IFF.1 This component contributes to meeting objective O.DETSPOOF by performing consistency checks on information to be mediated.

O.PRIVACY The functional requirements listed below in its entity are suitable to completely meet this objective. This is achieved by translating and thus hiding address information about internal network entities and by assuring that this mechanism is always invoked.

FPT_PSE.1 This component contributes to meeting objective O.PRIVACY by translating and thus hiding address information about internal network entities.

O.AUDGEN Functional requirement FAU_GEN.1 is suitable to completely meet this objective.

FAU_GEN.1 This component assures to meet objective O.AUDGEN by assuring that security-relevant information is stored in an audit trail.

O.AUDREV The functional requirements listed below in its entity are suitable to completely meet this objective. This is achieved by translating assuring that the audit trail can be read, understood (FAU_SAR.1) searched, and sorted (FAU_SAR.3) by an administrator.

FAU_SAR.1 This component contributes to meeting objective O.AUDREV by assuring that the audit trail can be read and understood by an administrator.

FAU_SAR.3 This component contributes to meeting objective O.AUDREV by assuring that the audit trail can be searched and sorted by an administrator for analysis.

O.IDS Functional requirement FAU_SAA.1 is suitable to completely meet this objective.

FAU_SAA.1 This component assures to meet objective O.IDS by monitoring audited events for potential security violations.

The following table traces all security functional requirements for the IT environment back to aspects of the security objectives for the IT environment:

Table 12: Security Objectives for the IT environment and SFRs for the IT environment.

Objectives for the IT environment	Security Functional Requirements for the IT environment
OE.TIME	FPT_STM.1

FPT_STM.1 This component assures to meet objective OE.TIME by providing reliable time information to the TOE

8.2.2 Rationale for Assurance Requirements

The assurance requirements are based on the EAL4 package and has been augmented with ALC_FLR.1 and AVA_VLA.3. This choice is to be considered appropriate for a firewall which is used in an environment of changing threats and permanent exposure. . This corresponds to firewalls used to protecet an internal network against the internet or used for internal security segmentation. Attackers are assumed to have solid knowledge of the internet protocols as well as access to all public information about the various firewall types, including information about the mechanisms used, their strengths and their weaknesses. Since firewalls are assumend to be permanentely connected to an untrusted network, an attacker has almost unlimited time for an attack attempt. The tools an attacker can use are widely available tools but also specifically developed tools that are unknown to the public or change rapidely in time.

8.2.3 Rationale for Strength of Function

The strength of function claim SOF-high is appropriate for this type of TOE and the selected assurance level. This claim is applicable to FIA_UAU.2 and FIA_UAU.5. SOF-high implies protection against deliberately planned or origanised breach of TOE security by attackers possessing a high attack potential. The authentication method used by the TOE enforces restrictions on login name and passphrases as well as authentication attempt limitations. Therefore the TOE will withstand an attack performed by an attacker using specially crafted tools performing automated authentication attempts over a significantly long period of time. We state that SOF-high is approriate, since the authentication mechanism protects the administartive access to the TOE and therefore the access to the security attributes.

8.2.4 Rationale for Mutual Support of Security Requirements

The following section demonstrates that the dependencies between functional components introduced by CC Part 2 are resolved.

- Selected functional component **FAU_GEN.1** is dependent upon functional component FPT_STM.1. Component FPT_STM.1 has not been selected for the TOE itself, but for the IT environment (i.e. the underlying operating system).
- Selected functional component **FAU_SAA.1** is dependent upon functional component FAU_GEN.1. Component FAU_GEN.1 has also been selected.
- Selected functional component **FAU_SAR.1** is dependent upon functional component FAU_SAR.1. Component FAU_SAR.1 has also been selected.
- Selected functional component **FAU_SAR.3** is dependent upon functional component FAU_GEN.1. Component FAU_GEN.1 has also been selected.
- Selected functional component **FDP_IFC.1** is dependent upon functional component FDP_IFF.1. Component FDP_IFF.1 was also selected.
- Selected functional component **FDP_IFF.1** is dependent upon functional components FDP_IFC.1 **and** FMT_MSA.3. Components FDP_IFC.1 and FMT_MSA.3 were also selected.
- Selected functional component **FIA_AFL.1** is dependent upon functional component FIA_UAU.1. Component FIA_UAU.2, which is hierarchical to FIA_UAU.1 was also selected.
- Selected functional component **FIA_ATD.1** is not dependent upon other functional or assurance components.
- Selected functional component **FIA_SOS.1** is not dependent upon other functional or assurance components.
- Selected functional component **FIA_UAU.2** is dependent upon functional component FIA_UID.1. Component FIA_UID.2, which is hierarchical to FIA_UID.1 was also selected.
- Selected functional component **FIA_UAU.5** is not dependent upon other functional or assurance components.
- Selected functional component **FIA_UAU.7** is dependent upon functional component FIA_UAU.1. Component FIA_UAU.2, which is hierarchical to FIA_UAU.1 was also selected.
- Selected functional component **FIA_UID.2** is not dependent upon other functional or assurance components.
- Selected functional component **FMT_MOF.1** is dependent upon functional component FMT_SMR.1 **and** FMT_SMF.1[8]. Components FMT_SMR.2, which is hierarchical to FMT_SMR.1 and FMT_SMF.1 were also selected.

[8] Family FMT_SMF.1 and the dependency of FMT_MOF.1 to FMT_SMF.1 were introduced by interpretation 065.

- Selected functional component **FMT_MSA.1** is dependent upon functional components [FDP_ACC.1 **or** FDP_IFC.1] **and** FMT_SMR.1 **and** FMT_SMF.1[9]. Components FDP_IFC.1, FMT_SMR.2, which is hierarchical to FMT_SMR.1 and FMT_SMF.1 were also selected.
- Selected functional component **FMT_MSA.3** is dependent upon functional components FMT_MSA.1 **and** FMT_SMR.1. Components FMT_MSA.1 and FMT_SMR.2, which is hierarchical to FMT_SMR.1 were also selected.
- Selected functional component **FMT_MTD.1** is dependent upon functional component FMT_SMR.1 **and** FMT_SMF.1[10]. Components FMT_SMR.2, which is hierarchical to FMT_SMR.1 and FMT_SMF.1 were also selected.
- Selected functional component **FMT_SMF.1** is not dependent upon other functional or assurance components.
- Selected functional component **FMT_SMR.2** is dependent upon functional component FIA_UID.1. Component FIA_UID.2, which is hierarchical to FIA_UID.1 was also selected.
- Selected functional component **FPR_PSE.1** is not dependent upon other functional or assurance components.

To be able to determine whether the TOE meets its security objectives, some additional dependencies between functional components have to be considered. The primary security objectives of the TOE are related to enforcing the information flow security policy (O.MEDTF, O.DETSPOOF) and maintaining privacy (O.PRIVACY). These security objectives are addressed by FDP_IFC.1, FDP_IFF.1, FMT_MSA.3, and FPR_PSE.1. The other, supporting, security objectives are related to the ability to securely configure, operate, and manage the TOE (O.ADMIN, O.ROLE, O.IDENTIFY, O.AUTHEN, O.AUDGEN, O.AUDREV, O.IDS). This supporting character is inherited by the functional requirements deducted from these supporting objectives. FIA_UID.2 provides the basis for authenticating human users (FIA_UAU.2). FIA_UAU.2 and FIA_ATD.1 provide the basis for secure configuration, operation, and management of the TOE by assuring that only authenticated human users can get access to related functions and data incorporating specific roles. FIA_SOS.1, FIA_AFL.1, FIA_UAU.5, and FIA_UAU.7 require that the relevant mechanisms are flexible (FIA_UAU.5) and strong (FIA_SOS.1, FIA_AFL.1 and FIA_UAU.7). FMT_SMF.1, SMR.2, FMT_MOF.1, FMT_MSA.1, FMT_MSA.3, and FMT_MTD.1 require that access to sensitive functions and data of the TOE is restricted to authorised administrators and that specific administrator roles can be distinguished and enforced. The audited related functional requirements require that relevant information is logged (FAU_GEN.1), can be inspected efficiently (FAU_SAR.1, FAU_SAR.3), and is automatically analysed for peculiarities (FAU_SAA.1).

The following section demonstrates that the dependencies between assurance components introduced by CC Part 3 are resolved.

- Selected assurance component **ACM_AUT.1** is dependent upon assurance component ACM_CAP.3. Component ACM_CAP.4, which is hierarchical to ACM_CAP.3, was also selected.

[9] Family FMT_SMF.1 and the dependency of FMT_MSA.1 to FMT_SMF.1 were introduced by interpretation 065.

[10] Family FMT_SMF.1 and the dependency of FMT_MTD.1 to FMT_SMF.1 were introduced by interpretation 065.

- Selected assurance component **ACM_CAP.4** is dependent upon assurance component ACM_SCP.1 and ALC_DVS.1. Components ALC_DVS.1 and ACM_SCP.2, which is hierarchical to ACM_SCP.1, were also selected.
- Selected assurance component **ACM_SCP.2** is dependent upon assurance component ACM_CAP.3. Component ACM_CAP.4, which is hierarchical to ACM_CAP.3, was also selected.
- Selected assurance component **ADO_DEL.2** is dependent upon assurance component ACM_CAP.3. Component ACM_CAP.4, which is hierarchical to ACM_CAP.3, was also selected.
- Selected assurance component **ADO_IGS.1** is dependent upon assurance component AGD_ADM.1. Component AGD_ADM.1 was also selected.
- Selected assurance component **ADV_FSP.2** is dependent upon assurance component ADV_RCR.1. Component ADV_RCR.1 was also selected.
- Selected assurance component **ADV_HLD.2** is dependent upon assurance components ADV_FSP.1 and ADV_RCR.1. Components ADV_RCR.1 and ADV_FSP.2, which is hierarchical to ADV_FSP.1, were also selected.
- Selected assurance component **ADV_IMP.1** is dependent upon assurance components ADV_LLD.1, ADV_RCR.1, and ALC_TAT.1. Components ADV_LLD.1, ADV_RCR.1, and ALC_TAT.1 were also selected.
- Selected assurance component **ADV_LLD.1** is dependent upon assurance components ADV_HLD.2 and ADV_RCR.1. Components ADV_HLD.2 and ADV_RCR.1 were also selected.
- Selected assurance component **ADV_RCR.1** is not dependent upon other functional or assurance components.
- Selected assurance component **ADV_SPM.1** is dependent upon assurance component ADV_FSP.1. Component ADV_FSP.2, which is hierarchical to ADV_FSP.1, was also selected.
- Selected assurance component **AGD_ADM.1** is dependent upon assurance component ADV_FSP.1. Component ADV_FSP.2, which is hierarchical to ADV_FSP.1, was also selected.
- Selected assurance component **AGD_USR.1** is dependent upon assurance component ADV_FSP.1. Component ADV_FSP.2, which is hierarchical to ADV_FSP.1, was also selected.
- Selected assurance component **ALC_DVS.1** is not dependent upon other functional or assurance components.
- Selected assurance component **ALC_FLR.1** is not dependent upon other functional or assurance components.
- Selected assurance component **ALC_LCD.1** is not dependent upon other functional or assurance components.
- Selected assurance component **ALC_TAT.1** is dependent upon assurance component ADV_IMP.1. Component ADV_IMP.1 was also selected.

- Selected assurance component **ATE_COV.2** is dependent upon assurance components ADV_FSP.1 and ATE_FUN.1. Components ATE_FUN.1 and ADV_FSP.2, which is hierarchical to ADV_FSP.1, were also selected.
- Selected assurance component **ATE_DPT.1** is dependent upon assurance components ADV_HLD.1 and ATE_FUN.1. Components ATE_FUN.1 and ADV_HLD.2, which is hierarchical to ADV_HLD.1, were also selected.
- Selected assurance component **ATE_FUN.1** is not dependent upon other functional or assurance components.
- Selected assurance component **ATE_IND.2** is dependent upon assurance components ADV_FSP.1, AGD_ADM.1, AGD_USR.1, and ATE_FUN.1. Components AGD_ADM.1, AGD_USR.1, ATE_FUN.1 and ADV_FSP.2, which is hierarchical to ADV_FSP.1, were also selected.
- Selected assurance component **AVA_MSU.2** is dependent upon assurance components ADO_IGS.1, ADV_FSP.1, AGD_ADM.1, and AGD_USR.1. Components ADO_IGS.1, AGD_ADM.1, AGD_USR.1, and ADV_FSP.2, which is hierarchical to ADV_FSP.1, were also selected.
- Selected assurance component **AVA_SOF.1** is dependent upon assurance components ADV_FSP.1 and ADV_HLD.1. Component ADV_FSP.2, which is hierarchical to ADV_FSP.1, was selected. Component ADV_HLD.2, which is hierarchical to ADV_HLD.1, was also selected.
- Selected assurance component **AVA_VLA.3** is dependent upon assurance components ADV_FSP.1, ADV_HLD.2, ADV_IMP.1, ADV_LLD.1, AGD_ADM.1, and AGD_USR.1. Component ADV_HLD.2, ADV_IMP.1, ADV_LLD.1, AGD_ADM.1, AGD_USR.1, and ADV_FSP.2, which is hierarchical to ADV_FSP.1, were selected.

8.3 TOE Summary Specification Rationale

8.3.1 Mapping between TOE Security Functions and SFRs

The following table maps all TOE security functions to the TOE security functional requirements.

Table 13: Mapping of TOE Security Functions to TOE Security Functional Requirements

TOE Security Function	TOE Security Functional Requirements
Security Administration	FMT_MOF.1, FMT_SMF.1, FMT_SMR.2, FMT_MSA.1, FMT_MTD.1, FIA_ATD.1, FAU_SAR.1
Identification and Authentication	FIA_UID.2, FIA_UAU.2, FIA_UAU.5, FIA_AFL.1, FIA_ATD.1, FIA_SOS.1, FIA_UAU.7
Information Flow Control	FDP_IFC.1, FDP_IFF.1, FMT_MSA.3
Privacy	FPT_PSE.1
Security Audit	FAU_GEN.1, FAU_SAR.1, FAU_SAR.3, FAU_SAA.1

Security Administration

This security function provides the authorised administrator to configure, manage and visualise the TOE and its security functions. It allows to perform the security management function as specified in FMT_SMF.1 and supports the roles as specified in FMT_SMR.2. The role is maintained as required by FIA_ATD.1. An authorised administrator can manage the security function as specified in FMT_MOF.1, the security – attributes as specified in FMT_MSA.1, and the TSF data as specified in FMT_MTD.1. An administrator has to obey the restriction specified in FMT_SMR.2. An administrator can review logged information as specified in FAU_SAR.1.

Identification and Authentication

This security function identifies and authenticates users before allowing any other TSF-mediated actions on behalf of that user. It thus traces back to FIA_UID.2, FIA_UAU.2 and FIA_ATD.1. It supports several authentication mechanisms as required by FIA.UAU.5. The strength of this function is enhanced in accordance to the specifications in FIA_AFL.1, FIA_UAU.7, and FIA_SOS.1

Information Flow Control

This security function ensures that communication between internal and external networks through the firewall is controlled according to the Stateful Packet Filter SFP. This function traces back to FDP_IFC.1, FDP_IFF.1, FMT_MSA.3.

Privacy

This security function allows to hide information about internal networks. This function traces back to FPT_PSE.1.

Security Audit

This security function ensures that information about security relevant events are logged in an audit trail and allows an administrator to analyse the audit trail. This security function is capable of detecting port and address scans which indicate potential security violations. This function traces back to FAU_GEN.1, FAU_SAR.1, FAU_SAR.3, FAU_SAA.1.

The following table demonstrates the coverage of all TOE security functional requirements by the TOE security functions.

Table 14: Coverage of TOE Security Functional Requirements by TOE SecurityFunctions

SFR	TOE Security Functions
FAU_GEN.1	Security Audit
FAU_SAA.1	Security Audit
FAU_SAR.1	Security Audit Security Administration
FAU_SAR.3	Security Audit Security Administration
FDP_IFC.1	Information Flow Control

SFR	TOE Security Functions
FDP_IFF.1	Information Flow Control
FIA_AFL.1	Identification and Authentication
FIA_ATD.1	Identification and Authentication Security Administration
FIA_SOS.1	Identification and Authentication
FIA_UAU.2	Identification and Authentication
FIA_UAU.5	Identification and Authentication
FIA_UAU.7	Identification and Authentication
FIA_UID.2	Identification and Authentication
FMT_MOF.1 (1)	Security Administration
FMT_MOF.1 (2)	Security Administration
FMT_MSA.1 (1)	Security Administration
FMT_MSA.1 (2)	Security Administration
FMT_MSA.3	Information Flow Control
FMT_MTD.1 (1)	Security Administration
FMT_MTD.1 (2)	Security Administration
FMT_MTD.1 (3)	Security Administration
FMT_MTD.1 (4)	Security Administration
FMT_MTD.1 (5)	Security Administration
FMT_MTD.1 (6)	Security Administration
FMT_MTD.1 (7)	Security Administration
FMT_MTD.1 (8)	Security Administration
FMT_MTD.1 (9)	Security Administration
FMT_MTD.1 (10)	Security Administration
FMT_SMF.1	Security Administration
FMT_SMR.2	Identification and Authentication Security Administration
FPT_PSE.1	Privacy

The following discussion shows that all Functional Security Requirements are completely supported by the TOE Security Functions:

FAU_GEN.1: TOE Security Function "Security Audit" assures that all events are logged and all additional information is included in the related log entry as required by this security functional requirement.

FAU_SAA.1: TOE Security Function "Security Audit" assures that potential security violations (spoofing attempts, port scans, address range scans) are detected as required by this security functional requirement.

FAU_SAR.1: TOE Security Functions "Security Audit" and "Security Administration" together assure that administrators can review logged information as required by this security functional requirement.

FAU_SAR.3: TOE Security Functions "Security Audit" and "Security Administration" together assure that administrators can select from logged information as required by this security functional requirement.

FDP_IFC.1: TOE Security Function "Information flow Control" controls passing of packets between external IT entities as required by this security functional requirement.

FDP_IFF.1: TOE Security Function "Information flow Control" assures that the information flow policy is enforced as specified and required by this security functional requirement.

FIA_AFL.1: TOE Security Function " Identification and Authentication " assures that unsuccessful authentication attempts are limited as required by this security functional requirement.

FIA_ATD.1: TOE Security Functions " Identification and Authentication " and "Security Administration" together assure that the security attributes role, passphrase and public RSA key of human users are maintained as required by this security functional requirement.

FIA_SOS.1: TOE Security Function "Identification and Authentication" assures that the passwords meet the criteria mandated by this security functional requirement.

FIA_UAU.2: TOE Security Function "Identification and Authentication" assures that each human user has to be successfully authenticated before allowing any other interaction with the TOE as required by this security functional requirement.

FIA_UAU.5: TOE Security Function "Identification and Authentication" assures that human users can be authenticated via knowledge of passphrases or RSA keys as required by this security functional requirement.

FIA_UAU.7: TOE Security Function "Identification and Authentication" assures that no information is given to the user while the authentication is in progress as required by this security functional requirement.

FIA_UID.2: TOE Security Function "Identification and Authentication" assures that human users are identified as required by this security functional requirement. TOE Security Function "Information Flow Control" assures that external IT entities are identified as required by this security functional requirement.

FMT_MOF.1 (1): TOE Security Function "Security Administration" assures that the ability to determine the behaviour of the functions firewall service, log service and box access service is restricted to administrators - as required by this security functional requirement.

FMT_MOF.1 (2): TOE Security Function "Security Administration" assures that the ability to disable, enable, or modify the behaviour of the functions firewall service, log service and box access service is restricted to administrators - as required by this security functional requirement.

FMT_MSA.1 (1): TOE Security Function "Security Administration" assures that the ability to add, delete, or modify security attributes of the enforced information flow policy is restricted as required by this security functional requirement.

FMT_MSA.1 (2): TOE Security Function "Security Administration" assures that the ability to query security attributes of the enforced information flow policy is restricted to administrators - as required by this security functional requirement.

FMT_MSA.3: TOE Security Function "Information Flow Control" assures that restrictive default rules are used - as required by this security functional requirement.

FMT_MTD.1: TOE Security Function "Security Administration" assures that the ability to manage TSF data is restricted as required by the various iterations of this security functional requirement.

FMT_SMF.1: TOE Security Function "Security Administration" provides the capability to perform the security management functions as specified and required by this security functional requirement.

FMT_SMR.2: TOE Security Functions "Identification and Authentication" and "Security Administration" together assure that the roles "root administrator", "named administrator with read/write permission", named administrator with read-only permission" can be associated to users and that administrative ACLs are supported - as required by this security functional requirement.

FPT_PSE.1: TOE Security Function "Privacy" assures that alias names can be assigned to the IP addresses of external IT entities in order to hide their real IP addresses as required by this security functional requirement.

8.3.2 Mapping between Security Measures and Assurance Requirements

Chapter 6.2 claims that appropriate procedures are in place and corresponding documents will be created that provide evidence that the requirements are met.

8.4 Protection Profile Claims Rationale

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