FÄRIST
TUTUS - DIGITAL GATEKEEPERS

Security Target

Färist, Release 2.5.2

Document Version: 2.1

Date: 2007-09-20

Title	Färist Security Target		
Author(s):	Staffan Persson	Status:	Released
Version:	2.1	Classification:	Public
File name	FaristST.pdf	Date:	2007-09-20

Tutus Data AB, Svärdvägen 11, S-182 33 Danderyd, Sweden, Tel: +46 (0)8 551 102 30, Fax: +46 (0)8 551 102 39, E-mail: info@tutus.se, Web: http://www.tutus.se

Document History

Version	Date	Author	Changes to Previous Version
0.1	2006-06-25	Staffan Persson	Initial version based on ST for the 2.3-RELEASE
0.2	2006-07-13	Sebastian Mayer	Further adaption of ST 2.3-RELEASE and EAL 4+(FLR.1)
0.3	2006-07-26	Sebastian Mayer	- minor – table descriptions, references, image resize
0.4	2006-08-10	Staffan Persson	Made changes for 2.4-RELEASE and changes after review by Gerald Krummeck.
0.5	2006-08-11	Sebastian Mayer	Changes on format
0.6	2006-08-14	Sebastian Mayer	Changes on format
0.7	2006-08-18	Staffan Persson	Completed the changes for the EAL4+ for the application for certification
0.8	2006-08-25	Per Holmer	Changes in formatting
0.9	2006-08-31	Staffan Persson	Minor spelling corrections
1.0	2006-10-17	Staffan Persson	Minor changes due to comments from CSEC and Gerald Krummeck
1.1	2006-10-31	Staffan Persson	Included description of the different versions. Updates due to evaluator comments.
1.2	2007-03-07	Sebastian Mayer	Made changes for 2.5-RELEASE. Updates due to evaluator comments
1.3	2007-03-30	Sebastian Mayer	Updates due to evaluator comments
1.4	2007-04-17	Sebastian Mayer	Update of logging descriptions due to evaluator comments
1.5	2007-05-09	Sebastian Mayer	Remove syslog usage for local administration. Remove mapping of FDP.ACC/ACF to SF.RCHOUT and SF.RCONF. Corrected audit entries in FAU_GEN.1
1.6	2007-05-22	Staffan Persson	Clarification of log messages, descriptions of threats and scope of the TOE regarding crypto.
1.7	2007-06-29	Staffan Persson	Update of the telnet proxy and some assumptions for audit env.
1.8	2007-08-01	Staffan Persson	Changed TOE version to 2.5.1 and updated a policy for the env.
1.9	2007-08-07	Sebastian Mayer	Added statement to the supported cipher suites for TLS clients using the TLS-proxy. Replaced TLS_ RSA_WITH_3DES_EDE_CBC_SHA for Skut and IPSEC with TLS_DH _RSA_WITH_3DES_EDE_CBC_SHA. Added RFC for AES cipher suite.
1.10	2007-08-13	Staffan Persson	Updated TOE version names.
2.0	2007-08-29	Staffan Persson	Version change and audit event.
2.1	2007-09-20	Staffan Persson	Minor changes, as required before publication of the ST.

Table of Contents

Document History 2

1 Introduction 7
1.1 ST Identification 7
1.2 ST Overview 7
1.3 CC Conformance Claim 7
1.4 Strength of Function Claim 7
1.5 ST Content and Organisation 8
1.6 Related Standards and Documents 9

2 TOE Description 10
2.1 Introduction 10
2.1.1 Proxies 10
2.1.2 VPN functionality 11
2.1.3 Failover System 11
2.2 Architecture 12
2.3 Functionality 13
2.3.1 FreeBSD OS Kernel 14
2.3.2 Configuration and Administration 15
2.3.3 Basic Communication Rules 16
2.3.3.1 Failover System 18

3 TOE Security Environment 19
3.1 Assumptions 20
3.2 Threats 21
3.2.1 Threats Addressed by the TOE 22
3.2.2 Threats to be Addressed by the Operating Environment 23
3.3 Organisational Security Policies 23

4 Security Objectives 24
4.1 Security Objectives for the TOE 24
4.2 Security Objectives for the IT and non-IT Environment 25

5 IT Security Requirements 27
5.1 TOE Security Functional Requirements 30
5.1.1 Class FAU - Security Audit 30
5.1.1.1 FAU_GEN.1 – Audit Data Generation 30
5.1.1.2 FAU_SAR.1 - Audit Review 32
5.1.1.3 FAU_SEL.1 - Selective Audit 32
5.1.2 Class FCO - Communication 33
5.1.2.1 FCO_NRO.1 – Selective proof of origin 33
5.1.3 Class FCS – Cryptographic Support 33
5.1.3.1 FCS_CKM.1a Cryptographic key generation (admin) 33
5.1.3.2 FCS_CKM.1b Cryptographic key generation (SKUT) 33
5.1.3.3 FCS_CKM.2a - Cryptographic Key Distribution (admin cert) 34
5.1.3.4 FCS_CKM.2b - Cryptographic Key Distribution (admin keys) 34
5.1.3.5 FCS_CKM.2c – Cryptographic Key Distribution (SKUT cert) 34
5.1.3.6 FCS_CKM.2d - Cryptographic Key Distribution (SKUT keys) 34
5.1.3.7 FCS_COP.1a - Cryptographic Operation (admin RSA) 34
5.1.3.8 FCS_COP.1b - Cryptographic Operation (admin 3DES) 35
5.1.3.9 FCS_COP.1c - Cryptographic Operation (admin SHA) 35
5.1.3.10 FCS_COP.1d - Cryptographic Operation (SKUT RSA) 35
5.1.3.11 FCS_COP.1e – Cryptographic Operation (SKUT 3DES/AES) 35
5.1.3.12 FCS_COP.1f – Cryptographic Operation (SKUT SHA) 36
5.1.3.13 FCS_COP.1g – Cryptographic Operation (IPSEC 3DES/AES) 36
5.1.3.14 FCS_COP.1h – Cryptographic Operation (IPSEC SHA) 36
5.1.3.15 FCS_COP.1i - Cryptographic Operation (MD5) 36

5.1.4 Class FDP - User Data Protection....37
5.1.4.1 FDP_ACC.2 – Complete access control....37
5.1.4.2 FDP_ACF.1 – Security attribute based access control....37
5.1.4.3 FDP_IFC.1a - Subset information flow control (unauthenticated sfp)....38
5.1.4.4 FDP_IFC.1b – Subset information flow control (authenticated sfp)....38
5.1.4.5 FDP_IFF.1a – Simple security attributes (unauthenticated sfp)....38
5.1.4.6 FDP_IFF.1b - Simple security attributes (authenticated sfp)....39
5.1.5 Class FIA - Identification and Authentication....40
5.1.5.1 FIA_ATD.1 – User Attribute Definition....40
5.1.5.2 FIA_UAU.2 – User Authentication before any Action....40
5.1.5.3 FIA_UID.2 – User Identification before any Action....41
5.1.6 Class FMT - Security Management....41
5.1.6.1 FMT_MOF.1 - Management of security functions behaviour....41
5.1.6.2 FMT_MSA.1 - Management of security attributes....41
5.1.6.3 FMT_MSA.3 – Static attribute initialisation....42
5.1.6.4 FMT_MTD.1 – Management of TSF data....42
5.1.6.5 FMT_SMF.1 – Specification of management functions....42
5.1.6.6 FMT_SMR.1 - Security roles....43
5.1.7 Class FPT - Protection of the TOE Security Functions....43
5.1.7.1 FPT_RVM.1 – Non-Bypassability of the TSP....43
5.1.8 Class FTP - Trusted path/channels....43
5.1.8.1 FTP_ITC.1 Inter-TSF Trusted Channel (IPSEC)....43
5.2 TOE Security Assurance Requirements....43
5.3 Security Functional Requirements for the IT Environment....44
5.3.1 Class FAU - Security Audit....45
5.3.1.1 FAU_SAR.1 - Audit Review....45
5.3.2 Class FCS - Cryptographic Support....45
5.3.2.1 FCS_CKM.1a Cryptographic key generation (admin and VPN cert)....45
5.3.2.2 FCS_CKM.1b Cryptographic key generation (MD5 key)....46
5.3.2.3 FCS_CKM.2 Cryptographic key distribution....46
5.3.2.4 FDP_RIP.2 - Full residual information protection....46
5.3.2.5 FDP_SDI.1 - Stored Data Integrity Monitoring....46
5.3.3 Class FPT - Protection of the TOE Security Functions....47
5.3.3.1 FPT_STM.1 – Reliable Time Stamps....47

6 TOE Summary Specification....48
6.1 TOE Security Functions....48
6.1.1 SF.IPS – IP Stack....48
6.1.2 SF.PF – Packet filter....48
6.1.3 SF.PSD – Packet Screening Daemon....49
6.1.4 SF.AP – Application proxies....49
6.1.5 SF.KEYMAN – VPN Key Management....49
6.1.6 SF.VPNCH – VPN Channel....49
6.1.7 SF.LCONF – Local configuration tools....50
6.1.8 SF.RCONF – Remote configuration tools....50
6.1.9 SF.RCHIN – Remote Communication Channel....51
6.1.10 SF.RCHOUT – Remote Communication Channel....51
6.2 TOE Assurance Measures....51
6.2.1 AM.CONFIG – Configuration management....51
6.2.2 AM.DEL – Delivery and operations....52
6.2.3 AM.DEVEL – Development....52
6.2.4 AM.GUIDE – Guidance documents....52
6.2.5 AM.LFC – Life cycle support....53
6.2.6 AM.TEST – Tests....53
6.2.7 AM.VULN – Vulnerability assessment....53

7 PP Claims....54

8 Rationale....55
8.1 Security Objectives Rationale....55
8.1.1 Security Objective Coverage....55

8.1.2 Security Objectives Sufficiency....55
8.2 Security Requirements Rationale....57
8.2.1 Security Requirements Coverage....57
8.2.2 Justification of security requirements for the IT environment....58
8.2.3 Functional Security Requirements Sufficiency....58
8.2.3.1 Security Objectives for the TOE....58
8.2.3.2 Security Objectives for the TOE environment....60
8.2.4 Security requirements dependency analysis....60
8.2.5 Unresolved Dependencies....65
8.2.6 Strength of Function....66
8.2.7 Justification of the chosen EAL....67
8.3 TOE summary specification rationale....67
8.3.1 Security functions and assurance measures coverage....67
8.3.2 Security functions sufficiency....69
8.3.2.1 Assurance measures efficiency....71
8.3.3 Strength of Function....72
8.4 PP Claims Rationale....72
9 Appendix....73

Index of Tables

Table 5-1: Functional Requirements on the TOE....27
Table 5-2: Functional Requirements on the TOE....44
Table 5-3: Security Functional Requirements for Environment....45
Table 8-1: Objectives Related to Assumptions, Threats and Policies....55
Table 8-2: TOE Security Functional Requirements Related to Security Objectives....58
Table 8-3: Security Functional Requirements for the Environment Related to Security Objectives....58
Table 8-4: TOE SFR Dependency Analysis....64
Table 8-5: TOE environment SFR dependency analysis....64
Table 8-6: Security Functions meeting SFRs and Vice Versa....67
Table 8-7: Assurance measures meeting SARs....67
Table 8-8: SARs met by assurance measures....68

Illustration Index

Illustration 1: Example of Firewall Location....10
Illustration 2: Parallel use of the proxy and VPN functionality....11
Illustration 3: Serial use of the proxy and VPN functions....11
Illustration 4: Färist failover system with two peer firewalls....12
Illustration 5: TOE Boundary....13

1 Introduction

1.1 ST Identification

ST Title:	Security Target Färist, Release 2.5.2
Product Name:	Färist
Product Version:	Färist 2.5.2-RELEASE, Färist 2.5.2-R-RELEASE
Assurance level:	EAL 4+
CC Version:	2.3, as of August 2005
ST author:	Staffan Persson
ST publication date:	2007-09-20
ST Version:	2.1

1.2 ST Overview

This Security Target (ST) describes the security aspects of the Target of Evaluation (TOE) and its security environment. The TOE is the Färist, an application level firewall with IP filter and VPN functionality, which can be used to connect internal IP networks to public IP networks, such as the Internet and to connect trusted IP networks using the VPN functionality over untrusted public IP networks.

This ST covers two releases of the TOE: Färist 2.5.2-RELEASE is the version providing full firewall and VPN functionality for use in parallel or serial mode (see section 2.1.2), while Färist 2.5.2-R-RELEASE is a version restricted to serial (VPN) mode only. Both versions are developed for government use and are not available to the general public. For the purpose of this evaluation, both versions have the same security characteristics.

The ST contains a description of the security objectives and the requirements, as well as the necessary functional and assurance measures provided by the TOE. The ST provides the basis for the evaluation of the TOE according to the Common Criteria for Information Technology Security Evaluations (CC).

1.3 CC Conformance Claim

This ST is CC Part 2 Conformant and CC Part 3 Conformant, with the assurance level of EAL4 Augmented with ALC_FLR.1.

The Security Target is following the structure given in part 1 of the Common Criteria, using the guidance from ISO/IEC JTC 1/SC 27 N 2449 “Information technology – Security techniques – Guide for the production of protection profiles and security targets” ([CCG]).

This ST does not claim conformance to any existing Protection Profile (PP).

1.4 Strength of Function Claim

The TOE contains one cryptographic function (SF.PSD) that is realised by a probabilistic mechanism which is integrity check of received NTP packets. The minimum strength of function claimed for this function is SOF-high.

1.5 ST Content and Organisation

The ST has been structured in accordance with [CC] Part 1 and [CCG]. The main sections of the ST are the TOE description, TOE security environment, security objectives, IT security requirements, TOE summary description, rationale and annexes.

The TOE description provides general information about the TOE, serves as an aid to understand the nature of the TOE and its security functionality and provide context for the ST's evaluation.

The TOE security environment describes security aspects of the environment in which the TOE is to be used and the manner in which it is to be employed. The TOE security environment includes:

a) assumptions regarding the TOE's intended usage and environment of use

b) threats relevant to secure TOE operation

c) organisational security policies with which the TOE must comply

The security objectives reflect the stated intent of the ST. They pertain to how the TOE will counter identified threats and how it will cover identified organisational security policies and assumptions. Each security objective is categorised as being for the TOE or for the environment.

The security requirements section provides detailed requirements, in separate subsections, for the TOE and its environment.

The IT security requirements are subdivided as follows:

a) TOE security functional requirements

b) TOE security assurance requirements

c) Security requirements for the IT environment

The TOE summary specification addresses the security functions that are represented by the TOE to answer the security requirements.

The rationale presents evidence that the ST is a complete and cohesive set of requirements and that the TOE would provide an effective set of IT security countermeasures within the security environment. The rationale is in two main parts. First, a security objectives rationale demonstrates that the stated security objectives are traceable to all of the aspects identified in the TOE security environment and are suitable to cover them. Then, a security requirements rationale demonstrates that the security requirements (TOE and environment) are traceable to the security objectives and are suitable to meet them.

The annex contains a list of abbreviations and a glossary relevant for this ST.

1.6 Related Standards and Documents

[CC]	Information Technology – Security Techniques – Evaluation Criteria for IT Security, also known as the Common Criteria or CC - Common Criteria for Information Technology Security Evaluation. » Part 1: Introduction and general model. August 2005. Version 2.3. CCMB-2005-08-001 » Part 2: Security functional requirements. August 2005. Version 2.3. CCMB-2005-08-002 » Part 3: Security Assurance Requirements. August 2005. Version 2.3. CCMB-2005-08-003
[CEM]	Common Methodology for Information Technology Security Evaluation. Evaluation Methodology. August 2005. Version 2.3. CCMB-2005-01-004
[CCG]	ISO/IEC TR 15446 – ISO-Guide for the Production of Protection Profiles and Security Targets. First Edition. July 2004.
[Stevens]	Stevens, W.R.; Wright, G.R.: TCP/IP Illustrated, Volume 2, The Implementation, 1994.
[MCKU]	McKusick, Marshall Kirk / Neville-Neil George V., The Design and Implementation of the FreeBSD Operating System, version FreeBSD 5.2., Addison-Wesley 2005
[Dobbertin]	Dobbertin, H.: The status of MD5 after a recent attack, CryptoBytes, 2(2): 1-6, 1996.
[Wang]	Wang, Xiaoyun; Hongbo Yu (2005). “How to Break MD5 and Other Hash Functions”. EUROCRYPT. ISBN 3-540-25910-4.
[Robshaw]	Robshaw, M.J.B.: RSA Bulletin No. 4, November 12, 1996: On Recent Results for MD2, MD4, and MD5.
[Lenstra]	Lenstra, A.K.; Verheul, E.R.: Selecting Cryptographic Key Sizes
[Holmer]	Holmer, P and Lind R., Simple Key-exchange using TLS (SKUT), March 2004.
[FIPS180-2]	FIPS 180-2, Secure Hash Standard (SHS), 2002 August 1, “To specify a Secure Hash Algorithm to be used by both the transmitter and intended receiver of a message in computing and verifying a digital signature”.
[FMSSL]	FMSSL 2.0 Implementation, Tutus Data AB, Document Version: 0.1, 2007-05-21.
RFCs	The following Internet standards are applicable: FTP – RFC 959; SMTP – RFC 821, RFC 1869, RFC 1870, RFC 1891; DNS – RFC 1035; HTTP – RFC 2616; HTTPS – RFC 2069; TLS v1 – RFC 2246; TLS v1.1 – RFC 4346. IPSEC – RFC 2401; IP Encapsulating Security Payload (ESP) – RFC 2406, The ESP CBC-Mode Cipher Algorithms – RFC 2451, The Use of HMAC-SHA-1-96 within ESP and AH – RFC 2404, Virtual Router Redundancy Protocol (VRRP) – RFC 3768.

2 TOE Description

2.1 Introduction

The Färist is a software only product, packaged as a single contained packet together with a stripped-down version of FreeBSD6.2 and delivered on CD-ROM. It is tested on standard Intel PC-hardware, but can also be used on all hardware supported by FreeBSD (see http://www.freebsd.org).

The Färist has a twofold purpose:

- to provide controlled and audited access to services, both from inside and outside an organisation's network, by allowing, denying, and/or redirecting the flow of data through the firewall;
- and to establish a secure communication channel between two Färists over an unsecured network, as a virtual private network over a public network, such as the Internet.

Illustration 1 shows a logical representation of a firewall mediating traffic, or information flows, among internal and external networks. Although there are a number of firewall architectures and technologies, firewalls basically fall into two major categories: traffic-filter and application-level firewalls. The Färist is an application-level firewall system with proxies for several protocols, it also has IP filter and VPN capabilities.

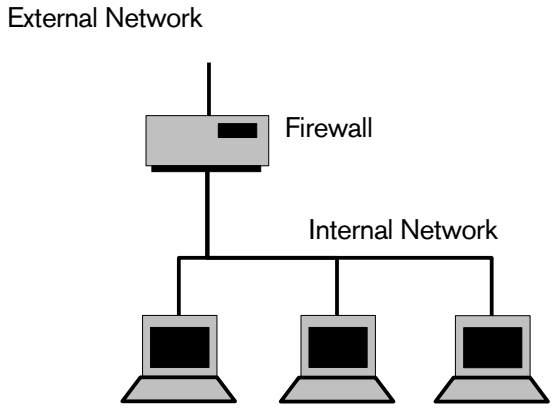

Illustration 1: Example of Firewall Location

There are several functions and proxies included in the Färist distribution such as an HTTP, SNMP or LDAP Proxy, but these are not part of the evaluated configuration. They have to be disabled in the evaluated configuration of the Färist. There is also a VPN client for Windows as part of the distribution which is also not part of the TOE.

In the following the parts of the TOE are described.

2.1.1 Proxies

The following proxies are part of the evaluated configuration of the Färist:

1. FTP-proxy
2. SMTP-proxy

3. DNS-proxy
4. TCP Plug-proxy (also called Plug-proxy).

The telnet protocol is also supported. However, telnet is implemented using the TCP Plug-proxy and will therefore not be mentioned as a separate proxy in this document.

Of all proxies, only the TCP Plug-proxy can be configured to have authentication. It is possible to use the plug proxy as a TLS proxy, which will require connections to present a valid certificate before access is granted. The allowed certificates are configured per proxy. Similarly, the proxy is also able to present a TLS certificate when establishing a connection to the outside.

2.1.2 VPN functionality

The firewall also provides a VPN functionality based on IPSEC, with fixed keys or dynamic key management using the SKUT protocol. The proxy functionality can be used separate (parallel) or in combination (serial) with the VPN functionality.

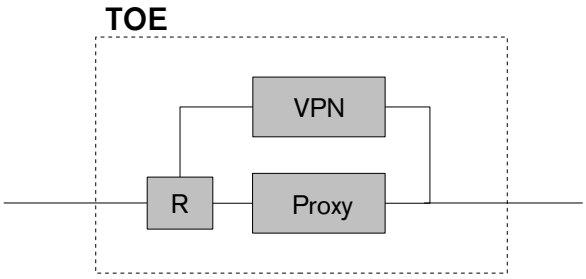

Illustration 2: Parallel use of the proxy and VPN functionality

In the parallel operating case the traffic either passes through a VPN tunnel or through a proxy function. The traffic is routed based on the configuration of the Färist.

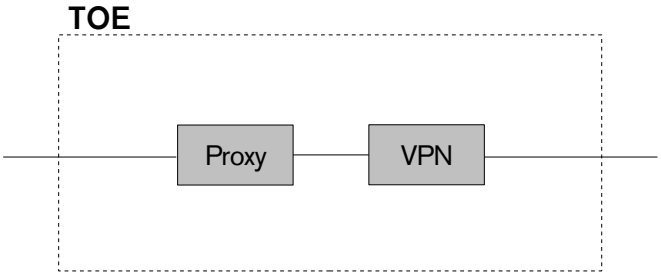

Illustration 3: Serial use of the proxy and VPN functions.

In the serial operating case all of the traffic passes both the proxy function *and* the VPN system. This is the only mode allowed for restricted version (Färist 2.5.2-R-RELEASE). In the picture above the internal interface is on the left and the external interface is on the right.

2.1.3 Failover System

Färist provides a failover system, too. The failover system allows a Färist to be in hot-standby for another Färist (see Illustration 4). They are equally configured and have the same rights so they are "peer firewalls". The definitions active and backup only determine which firewall is currently active and passive respectively. The standby

Färist will become active if the active Färist fails. This increases the availability of the Färist, but not the security as evaluated in this security target. So it is not an additionally claimed security functionality.

There are two evaluated configurations: the single firewall and the failover system.

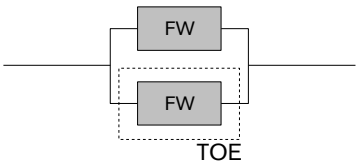

Illustration 4: Färist failover system with two peer firewalls

The TOE can be configured as either a single firewall or as a peer in a failover configuration in which the TOE can act as the active or the backup system. These two configurations apply to both versions of the TOE, the Färist 2.5.2-RELEASE and the Färist 2.5.2-R_RELEASE.

2.2 Architecture

The TOE includes the parts of Färist 2.5.2-RELEASE and Färist 2.5.2-R-RELEASE identified below, including the carp daemon, proxy daemon, the VPN functionality, TCP Plug-, FTP-, SMTP- and DNS-Proxy using FreeBSD version 6.2 as the underlying operating system.

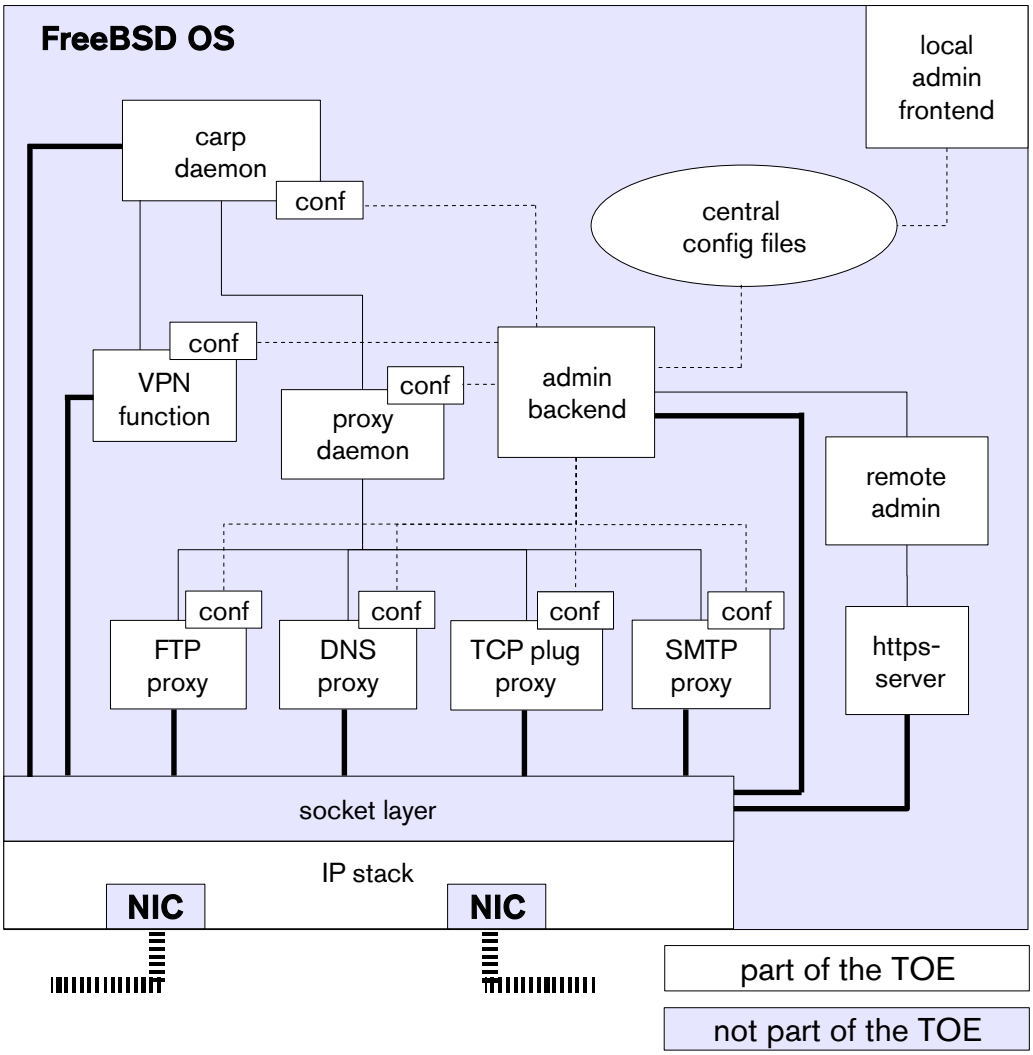

Illustration 5: TOE Boundary

The TOE consists of the following parts:

- the FTP-proxy,
- the SMTP-proxy consisting of omb_smtp and omb_smtpd,
- the DNS-proxy,
- the TCP Plug-proxy,
- the VPN functionality consisting of vpnd and skutd,
- the failover system consisting of the proxy daemon and carp daemon
- the IP stack of the kernel (for specific boundaries see illustration 5) including
  - the FreeBSD packet filter
  - the VPN-check routing function,
  - the packet screening daemon,
- the PHP-based remote administration tool including
  - the reboot cgi application,
  - the https server

- administration backend, consisting of
  - modifications to the FreeBSD sysinstall program,
  - fwadmin wrapper script,
  - fconfig and its wrapper newfconfic
  - the configuration daemon configd using the configc and config_sync script.

working together as described in chapter 2 (TOE description).

All components are handled under version control and are part of the Färist 2.5.2-RELEASE, Färist 2.5.2-R-RELEASE.

The cryptographic functions that are used by the TLS and the IPSEC protocols are provided by the cryptographic module of FMSSL [FMSSL]. The cryptographic module is provided by the Swedish Defence and is not considered part of the TOE. Note that the implementation of the TLS and IPSEC protocols are part of the TOE, it is only the cryptographic modules that are not part of the TOE. It is not configurable but is hard coded which of the modules is used to implement what algorithm.

2.3 Functionality

The proxies and the VPN function are user level daemons running on top of the underlying operating system. The proxies are controlled by the proxy daemon which starts, stops, reconfigures and monitors the proxies so that a reboot of the whole firewall is not needed when only one proxy configuration is changed.

In the case of a failover configuration, the carp daemon handles all network-related parts of the failover system and controls the proxyd and the VPN functionality. It is used to decide which firewall is to be active and has to start the accordant proxy daemon.

The remote administration tools consist mainly of the HTTPS server, a PHP scripting language based web application assisted by a cgi script. The local administration tools are on the one hand a wrapper for the FreeBSD sysinstall program to allow an administrator the modification of the central configuration file and on the other hand some programs to generate individual configuration files out of central configuration files.

Configd is the configuration daemon and mainly used by the remote administration system to interact with the Färist configuration. Its functions are updating the configuration and rebooting the system. It periodically synchronises the configuration with the failover peers using the config_sync script, too. The daemon can send a signal to the carp daemon to indicate that the system has become active. The signals and scripts are sent on a local socket.

2.3.1 FreeBSD OS Kernel

Parts of the kernel of the underlying FreeBSD OS are included in the TOE: these are core elements of the IP stack, dealing with the IP packet handling. The parts included and its external interfaces are:

- the protocol layer IP processing (please refer to [Stevens], chapter 8 and [MCKU], chapter 13.3 for further description), with
  - the functions ip_input() and ip_output() building the interface to the lower

level interface layer (which is part of the environment),

- the function ip_forward() that forwards packets between interfaces and
- the function calls icmp_input() ([Stevens], chapter 11 and [MCKU], chapter 13.8) and udp_input() ([Stevens], chapter 23 and [MCKU], chapter 13.2) of the ICMP and UDP processing building the interface to protocol layer ICMP and UDP processing (which are part of the environment);

» the protocol layer TCP processing with

- the function tcp_ctloutput() ([Stevens], chapter 30 and [MCKU], chapter 13.7) handling the system calls regarding TCP/IP operations as interface to the kernel parts which are part of the TOE environment; and
- the packet filter and a packet screening daemon with only internal interfaces.

Färist contains many other proxies; however, only the proxies listed above are part of the evaluated configuration.

Parts of the Färist software belonging to the IT Environment are listed in chapter 3. Please refer also to illustration 5 for a graphical presentation of the TOE boundaries.

Färist mediates the complete information flow between the internal and external network (provided there is no other network connection between them).

2.3.2 Configuration and Administration

A Färist system consists of an x86 Intel compatible PC with two Ethernet cards, FreeBSD as the operating system (with specific kernel generation options and with a minimum set of commands needed to perform the firewall functions) and the Färist product software which consists of the proxies, VPN functionality, failover system, IP-packet filtering and screening software, software that checks the integrity of critical files (Tripwire), the software for local administration (fconfig and its wrapper newfconfig), the software for updating the configuration (configd and config_sync) and the software needed for remote administration (simple HTTPS server, reboot CGI for rebooting the system and the PHP-based web frontend for the remote configuration). In addition the FreeBSD TCP/IP protocol software has been modified to log all packets to ports without an active listener (if not explicitly excluded from logging in the configuration file), the parser of the packet filter rules has been extended, selective packet forwarding for VPN has been added, packet screening has been implemented and a support for full transparent proxies has been added.

Again, please note that not the whole Färist software is part of the TOE.

The two Ethernet interfaces represent the interface to the “internal” and “external” network. The configuration file defines, which is the interface to the internal and which is the interface to the external network. The configuration file also defines, which network addresses belong to the internal network.

The Färist has a feature for remote administration. This is accomplished by a TLS-protected connection between an administrator’s web client and the web server. The TLS implementation of the web server is being part of the TOE, but the client is part of the TOE environment. The TLS client authentication mechanism is based on client and server certificates. It is assumed that client certificates are issued for administrators only and that administrators protect their private keys. Configuration options for remote administration are:

- whether remote administration is turned on or off
- whether remote administration is allowed from the internal interface, the external interface or both interfaces
- whether reconfiguration without reboot is allowed
- the list of authorised administrators

A remote administrator is able to edit the configuration file. The changes she made will be active after the next reboot of the system, since the system prohibits the direct manipulation of the configuration file when in multi-user mode. The remote administrator is able to initiate the reboot. During the reboot process the new configuration file will be checked for syntax errors and then copied to the active configuration file. If changes without reboot are allowed, the configuration file is checked for syntax errors, the configuration updated and a script is run to enable the new configuration. The old configuration file will be backed up along with audit information on which administrator made the changes and when.

The Färist is also able to update its own configuration by obtaining a configuration from another Färist. This is an important feature in a failover configuration, where all the failover peers must have the same configuration. This configuration update is achieved by having a Färist logging in as a remote administrator to another Färist to get a later version of a configuration file.

2.3.3 Basic Communication Rules

The Färist will implement the following basic rules for the communication between the internal and the external network:

1. Communication is allowed only for those addresses/ports with which either a Färist proxy is associated or a VPN connection is allowed. Packets arriving for other ports will be audited (if not excluded from audit in the configuration file) and dropped.
2. The Färist will check the IP source address of each packet and reject the packet, if it arrives at the "wrong" interface (i.e. if a packet with an internal IP address arrives at the external interface and if a packet with an external IP address arrives at the internal interface).
3. For NTP packets the keyed MD5 hash sum is checked.
4. DNS packets are checked for the length of domain names. Queries with domain names longer than 130 characters are rejected. Queries with domain names between 80 and 130 characters are logged. These values are defaults and can be changed by the administrator.
5. Specific TCP-protocols are guarded by proxies, which check the data on those protocols against specific rules. The rules possible are described in a configuration file for each proxy.
6. For VPN connections, the connection is authenticated when it is established. The connection is then IPSEC-encrypted to provide confidentiality and integrity of the traffic. The VPN connection can be configured to only allow certain IP protocols in specific directions.

The VPN and proxy functionality can be combined such as the proxies are activated also for any traffic going through the VPN.

The proxies running on the TOE and the rules they are able to implement are listed below.

FTP-Proxy

The FTP-Proxy can be used to guard the FTP protocol. Rules can be set up which allow to establish an FTP connection only in one direction (e. g. from the "internal" to the "external" interface). In addition, the possible source and destination addresses of the connection can be defined. The FTP-proxy also performs some checks on FTP commands. Only a subset of the commands of the FTP protocol are known to the proxy. Rules may be specified that restrict the use of those commands.

SMTP-Proxy

The SMTP-Proxy handles the email exchange via the SMTP protocol as described in RFC821. The proxy consists of two programs such that the one receives and queues the email and the other one monitors the queues and does the delivery. The behaviour of the proxy is determined by rules set by the administrator. The proxy can use different internal email servers and reject emails that exceed a certain size or are addressed to multiple recipients. Incoming email from a mail server with an incorrect MX record may be rejected.

DNS-Proxy

The DNS-Proxy is designed to provide protection against DNS responses trying to redirect internal resources to the external network. If configured, all queries and responses are matched and examined to verify that they do not contain any references to internal domains and networks.

The DNS-Proxy receives DNS queries from the inside networks, verifies that they do not contain any internal domain names and forwards them to the external network. Domain names in queries and responses are also checked against configurable name lengths to make sure that domain names stay within reasonable length limits. In addition the DNS-proxy checks that for each reply coming from the external server there has been an associated query from the internal server.

Queries and responses are always either accepted and redistributed unaltered or rejected and dropped. All rejected queries and responses are logged. Accepted queries and responses can also be logged if the proxy is configured to do so.

TCP Plug-Proxy

The Plug-Proxy can be used to guard any TCP-based protocol. It will not perform any checks on the user data. All checks are performed on the data in the TCP header. Checking rules are defined in a configuration file for the proxy. Possible checks are:

- Source and destination address. Rules can be defined that restrict the source and destination addresses allowed to use the service.
- Port number. The Färist checks the incoming port numbers and remaps them if specified.

It is possible to use the plug proxy as a TLS proxy, which will require connections to present a valid certificate before access is granted. The allowed certificates are configured per proxy. Similarly, the proxy is also able to present a TLS certificate when establishing a connection to the outside. The users known to the TOE are the users being trusted with a certificate.

Since the TCP Plug-proxy is so generic it can be used to implement other protocols,

such as Telnet. The Telnet proxy is therefore just an instantiation of the plug proxy using a specific port number.

VPN Functionality

The VPN functionality establishes IPSEC tunnels to the Färist, providing authenticated, encrypted and integrity-protected connections between Färists, compliant with the IPSEC RFCs. A VPN connection is established by mutual authentication using SHA-1 and encrypted using Triple DES or AES256. The encryption between two Färist connections can either be performed by fixed shared secret keys or by using public key certificates and dynamic session keys. Dynamic keys are recommended and fixed keys are only intended to be used as a fallback routine. Only dynamic keys are part of the evaluated configuration.

For dynamic session keys, the key management for session keys uses TLS and the SKUT protocol. For an authenticated and established connection between two Färists, additional VPN traffic filter rules are being applied, based on the protocols used, and the source and destination address.

The TOE is intended to be used in three different ways, all being part of the evaluated configuration:

- As a firewall with IP filtering functionality and proxies. This mode can be used to connect internal networks to public networks, such as the Internet.
- As a VPN crypto channel connecting an internal trusted network to an external trusted network over an untrusted (public) network. When connecting a restricted network to another network, the Färist is configured to block all traffic with the exception of the VPN traffic. This is the only mode of operation for the VPN Färist 2.5.2-R-RELEASE.
- As a firewall with VPN crypto channel connecting an internal network both to traffic to the public network as well as connecting an internal network to an external trusted network over an untrusted network.

2.3.3.1 Failover System

The failover system allows a Färist to be in hot-standby for another Färist. The standby Färist becomes active if the active Färist fails.

The Färist Failover System handles three types of failure:

1. Complete failure of the active Färist

   If the active Färist does not send any “watchdog messages” for a determined time the backup firewall will notice this and become active.

2. Network interface failure of the active Färist

   Both Färists periodically ping configured hosts on all network interfaces and the backup Färist compares the value it got within the watchdog message from the active Färist with its own. If the received value is lower for an amount of time the backup Färist will become active.

3. Partial failure of the active Färist

   All proxies running on the Färist send watchdog messages to the proxyd daemon. The proxyd daemon will restart dead proxies and if that fails reboot the Färist. Rebooting the active Färist will make the backup Färist to become active. After a reboot the Färist will not be rebooted again during a grace period to stop the

system from oscillating.

This functionality increases the availability of the Färist; since this ST makes no availability claims for the TOE, this feature is available to users of the TOE without being considered a security function for this evaluation.

The failover system consists of two daemons:

1. Carpd implements a modified version of the CARP protocol; it handles all network-related parts of the failover system and decides when to become active.
2. Proxyd starts, stops and monitors proxies. When to start and stop proxies is decided by carpd by sending signals to proxyd. The monitoring of proxies works by all proxies sending watchdog messages to proxyd. If watchdog messages are missing from a proxy proxyd will try to restart it, if that fails proxyd can reboot the whole machine.

The configuration of the failover Färist systems must be synchronised, in order for the failover to be secure. This is achieved by the configuration backend, using the daemon configd. In a failover configuration, configd periodically synchronises the configuration with the failover peers. It can also send a signal to the carpd daemon to request that the (passive) failover system becomes active. The configd is then using the remote administration interface of a peer to obtain the configuration file.

3 TOE Security Environment

The TOE is set up between an internal and an external IP network to control connections between those two networks. The TOE is not only in itself an asset, but it aims to protect all assets which are (typically) placed in the internal network and therefore shall be protected appropriately.

Certain trusted networks are connected to the internal network over a trusted channel. Each of these connections may be subject to certain access limitations, based on the services needed between these networks.

The TOE is intended to be used in a physically protected environment. It is assumed that no unauthorised personnel has physical access to the TOE. So all attacks to the TOE have to be performed over the network connections of the TOE.

It is assumed that the TOE is not used for any other tasks than the firewall application. No other applications shall be part of the firewall system.

It is assumed that the underlying hard- and firmware operates according to their specification and has no security critical side-effects on the operation of the TOE. Hard- and firmware are not part of this TOE, but of course the functions of the TOE rely on it.

It is assumed that the TOE or its failover peers, in case of a failover configuration, provide the only network communication link between the "internal" and "external" network separated by the TOE and its failover peers. This implies that all traffic between those two networks has to pass through the TOE or in case of failover through the failover peers.

For a failover configuration of the TOE, it is assumed that the failover peer is an identically configured firewall under the same administration as the TOE. It is also assumed that all the assumptions on the TOE and TOE environment also are valid for the peer.

Furthermore the TOE is assumed to be either TEMPEST protected or operate in an environment where interception of radiation is covered by other environmental measures. The evaluation will therefore not address vulnerabilities caused by emanation from the TOE.

Administrators of the TOE and users authenticated with a TLS certificate are considered to be trustworthy. The TOE will not protect itself against an administrator who tries to bring the TOE into an insecure state. It is also assumed that administrators are well trained, reducing the risk that they accidentally make security critical administration mistakes.

The TOE uses FreeBSD as the operating system basis, which is delivered with the TOE in one package tied together in a way that prevents the possibility of setting up the TOE on another operating system. The OS has been modified to exclude critical system calls from the kernel, to omit critical setuid programs. FreeBSD operates with "securelevel = 3" and all critical files are marked "system immutable", i.e. they can only be changed when the system is in single user mode.

The TOE is based on the FreeBSD operating system. The TOE includes only specified parts of the operating system. In general FreeBSD and its security functions are part of the environment and it is assumed that they work correctly. Functions considered to belong to the environment are, e.g.:

- the memory management functions
- the functions related to program execution
- the access control functions and privilege management
- the boot and shutdown sequences and cron jobs

Other software that is delivered with the Färist and the underlying FreeBSD system respectively and identified as being part of the environment are:

- the sendmail program for delivering SMTP mail, called by the SMTP proxy,
- the syslog facility for generating log files, and
- the Tripwire program to detect modifications to the file system.

The cryptographic functions that are used by the TLS and the IPSEC protocols are provided by the cryptographic module of FMSSL [FMSSL]. The cryptographic module is provided by the Swedish Defence and is not considered part of the TOE. Note that the implementation of the TLS and IPSEC protocols are part of the TOE, it is only the cryptographic modules that are not part of the TOE. It is not configurable but is hard coded which of the modules is used to implement what algorithm.

The integrity check of received NTP packets that is part of the security function (SF.PSD) is using a cryptographic module that is implemented as part of the TOE.

The assumptions made and the threats addressed are summarised in the following sections. In those sections we use the term

- user

  for a person operating a system or a system in the internal or external network generating IP-packets that he wants to pass through the firewall or trying to get access to the TOE either via one of its network interfaces or by trying to get access to other interfaces of the TOE. The identity of those "users" is usually not known to the TOE and the TOE is not able to authenticate them. The only users visible to the TOE are those user who have been given a TLS certificate to be given access for a specific TCP plug-proxy;

- administrator

  for a person authorised to perform administrative functions on the TOE using the TLS secured remote administration interface or sitting at the local system console. Administrators will be identified and authenticated by the TOE environment, either by verifying their TLS certificate or requesting a valid standard FreeBSD login.

3.1 Assumptions

This section describes the assumptions that must be satisfied by the TOE environment.

The following conditions are assumed to exist in the operational environment.

A.AUTKEY It is assumed that private keys used for the certificates imported to the TOE are of high quality and not disclosed, replaced or modified. This applies to the private keys of the certificates for the administrators, the users, the server, as well as for the certificates used by the VPN connection.

A.GENPUR	There are no general-purpose computing capabilities (e.g., the ability to execute arbitrary code or applications) on the TOE. The underlying system of the TOE is solely deployed to host the TOE.
A.MD5KEY	It is assumed that secret keys used for the generation and verification of the keyed MD5 hash sums are of high quality, are not disclosed and do belong to the assigned NTP server.
A.NOEVIL	Authorised administrators and users given privileges, are competent, non-hostile and follow all their guidance; however, they are capable of error.
A.NOEMA	Interception of emanation of any kind is addressed by environmental controls that reduce the signal to noise ratio for an interceptor to a level that prohibits useful evaluation of the intercepted signals.
A.PHYSEC	The TOE is physically secure, i.e. no unauthorised persons have physical access to the TOE and its underlying system.
A.RELHARD	The underlying hardware, firmware (BIOS and device drivers) and the operating system functions needed by the TOE to guarantee secure operation, are working correctly and have no undocumented security critical side effect on the functions of the TOE.
A.SINGE	The connection from the TOE to an external trusted network using the VPN functionality has only one connection point on the other network, i.e. the external network meets the assumption A.SINGI for its Färist.
A.SINGI	Information cannot flow among the internal and external networks unless it passes through the TOE or a failover firewall to the TOE, i.e. the TOE or its failover firewall is the only connection point between those two networks.
A.PEERTRUST	The TOE firewall peers that are configured as failover must be trustworthy. That means that they are all under the same administration as the TOE, identically configured and that the same assumptions can be made about them as for the TOE.

3.2 Threats

The threats described in this chapter are addressed either by the TOE or the environment.

The threat agents are typically attackers (unauthorised users or systems) logically accessing the TOE from the external network connected to the TOE, with a low attack potential.

The assets that are subject to attacks are the IT resources on the internal network behind the Färist and assets communicated over the VPN (user data). The TOE data that is subject to attack is either residing on the TOE and its underlying system resp. itself, which are the operating system, the TOE specific programs (especially the proxy programs), the TOE configuration data and the TOE audit data, or the resources placed in the internal network.

All attackers in the following threat descriptions are expected to possess at maximum the following attack potential:

- **expertise**
  the expertise of potential attackers is estimated to be fairly average - attackers know IP and related networking protocol basics and are trying to find vulnerabilities publicly known to affect standard operating systems, including the underlying system of the TOE.
- **available resources**
  the physical resources required for an attacker to stage an attack are limited. It is assumed that one or a few standard workstation computers and an Internet connection are enough to launch even sophisticated attacks. Time resources are, related to the motivation, expected to do not exhaust the range of at maximum some man days. Network attack tools available on the Internet are considered to be available, too.
- **motivation**
  the TOE aims to protect restricted communication and internal networks against publicly accessible networks like the Internet. So, the attackers are assumed to be motivated by high-value assets and e.g. by the fact to "hack" a restricted network.

3.2.1 Threats Addressed by the TOE

The threats discussed below are addressed by the TOE.

T.ASPOOF	An external attacker may cause information to flow through the TOE into a connected network where the source address in the information is obviously spoofed.
T.DISCLOSE	An external attacker gains unauthorised access to information transmitted between the TOE and an external trusted network.
T.INISEC	For configuration settings which are not provided by an administrator, insecure default values may be set by the TOE.
T.MEDIAT	An attacker in the external network may send impermissible information through the TOE, including illegally formed requests, to exploit vulnerabilities of systems in the internal network.
T.MODIFY	The attempts of an external attacker to modify data transmitted between the TOE and an external trusted network goes undetected.
T.NOAUTH	An attacker may be able to perform administration or configuration of the TOE, or access to network services on the inside of the TOE that requires authentication without being properly authenticated. This may also trigger a TOE in a failover configuration to disclose or obtain a configuration file from a unauthorised machine claiming to be a peer.

T.REMOTE An attacker may through the remote administration and configuration channel gain access to administration information and configuration data, such as secret keys or audit records, or may be able to modify such data. This threat is possible only against remote administration sessions.

T.SELPRO An attacker may read, modify, or destroy TOE internal data by transmitting data to the TOE via one of its network connections that causes modification or deletion of TOE internal data.

T.TIME An attacker with knowledge about the used mechanism for NTP synchronising of the TOE with a time server may try to spoof NTP messages, causing the wrong time to be delivered to the TOE to confuse the audit log entries.

T.USEACC An attacker may try or conduct misuse against the TOE by any means without his actions being detected.

3.2.2 Threats to be Addressed by the Operating Environment

The threat possibility discussed below must be countered by procedural measures and/or administrative methods:

TE.AUDIT An attacker may manipulate the underlying system of the TOE in a way that authorised administrators are not able to read the audit data via console access.

TE.FILE An attacker may gain the possibility to alter user or TSF data without being detected.

TE.INFO An attacker may gain access to TSF data (like TLS session keys used for the remote administration encryption) by allocating memory on the underlying operating environment which still contains such data from a previous allocation.

3.3 Organisational Security Policies

There is only one organisational security policy, making demands on the accountability of administrator actions:

P.ADMACC Administrators and users of authenticated services shall be accountable for the actions they conduct (e.g. erroneous alteration of the TOE configuration) by generating sufficient audit records for their actions.

4 Security Objectives

The security objectives provide a concise statement of the intended response to the security problem. It will describe which security needs will be addressed by the TOE and which will be addressed by the TOE environment, in the form of a statement of security objectives.

4.1 Security Objectives for the TOE

The following are the IT security objectives to be met by the TOE.

O.ATISRC The TOE must provide the means to authenticate the integrity of NTP packets coming from the network.

O.AUDIT The TOE must be able to provide audit evidence of security relevant events as well as for authorised use of security functions and enable the authorised administrator to configure the kinds of evidence recorded and to read the recorded audit trail.

O.DISCLOSE The TOE must be able to provide a trusted channel to external trusted networks and protect information transmitted to and received from such networks against unauthorised disclosure.

O.IDAUTH The TOE must uniquely identify and authenticate the claimed identity of all administrators, users and peer firewalls, before allowing administrators to establish a remote administration channel, giving users access to network services on the inside of the TOE that require authentication, or giving peer firewalls access to the configuration channel.

O.LIMEXT The TOE must restrict the means to control and limit the use of communication protocols between the internal and the external interface to an authorised administrator.

O.MEDIAT The TOE must mediate the flow of all information flowing between the TOE's internal and external interfaces.

O.MODIFY The TOE must be able to provide a trusted channel to external trusted networks to detect any modification of incoming information transmitted from such networks, and to provide the means for the remote network to verify the integrity of information transmitted out of the TOE to such networks.

O.REMOTE The TOE must provide a secure remote communication channel to remote trusted IT products for administration of the TOE and exchange of configuration files to prevent unauthorised users from unauthorised access and modification of the remote administrator actions or to the configuration files.

O.SECSTA	Upon initial start-up of the TOE or during configuration the TOE shall provide well-defined initial settings for security relevant functions.
O.SELPRO	The TOE must protect itself against attempts by attackers to bypass, deactivate, or tamper with TOE security functions.

4.2 Security Objectives for the IT and non-IT Environment

The following are the TOE non-IT security objectives that are to be satisfied without imposing technical requirements on the TOE. That is, they will not require the implementation of functions in the TOE hardware and/or software, but may require such implementation in the underlying hard- and software of the TOE as part of the environment.

Thus, the following environmental objectives may partly be IT specific and partly related to administrative methods and/or procedural measures.

OE.NOEVIL	Authorised administrators and users are trained as to establishment and maintenance of sound security policies and practices for the privileges they have been given.
OE.AUDIT	The underlying operating system must enable the authorised administrator to read and manage the recorded audit trail.
OE.AUTKEY	The private keys for the certificates used for administrator, users, server and VPN must be of high quality and must be protected from disclosure, modification or replacement.
OE.FILESEC	The TOE environment must protect configuration and other TSF data stored in files against any undetected unauthorised modification.
OE.GENPUR	The underlying hard- and software of the TOE must not be used for any general purposes other than TOE-related computing and data storage, i.e. the resources are used solely for operating the TOE.
OE.MD5KEY	The quality of the secret key generation for the generation and verification of keyed MD5 hash sums for NTP packets must be high and the keys must be protected from disclosure when stored outside of the TOE.
OE.NOEMA	Interception of emanation of any kind is addressed by environmental controls according to the relevant security policy.
OE.PHYSEC	The TOE and its underlying hardware must be protected from physical access by unauthorised personnel.
OE.RELHARD	The underlying hardware, firmware (BIOS and device drivers) and operating system functions needed by the TOE to guarantee secure operation, must be working correctly and must not have undocumented security critical side effects on the functions of the TOE.

OE.RESID	The TOE environment must ensure that residual information from a previous information flow is not disclosed.
OE.SINGE	The connection from the TOE to an external trusted network, using the VPN functionality, must only be reachable on one connection point on the other network.
OE.SINGI	The connection provided by the TOE is the only one between external and internal networks.
OE.TIME	The TOE environment must provide a reliable time source.
OE.PEERTRUST	The TOE firewall peers that are configured as failover must be trustworthy. That means that they are all under the same administration as the TOE, identically configured and that the same assumptions can be made for them as for the TOE.

5 IT Security Requirements

The following table gives an overview of the functional components from the Common Criteria Part 2 that are relevant for this TOE.

Component	Component Name
FAU_GEN.1	Audit data generation
FAU_SAR.1	Audit review
FAU_SEL.1	Selective audit
FCO_NRO.1	Selective proof of origin
FCS_CKM.1	Cryptographic key generation
FCS_CKM.2	Cryptographic key distribution
FCS_COP.1	Cryptographic operation
FDP_ACC.2	Complete access control
FDP_ACF.1	Security attribute based access control
FDP_IFC.1	Subset information flow control
FDP_IFF.1	Simple security attributes
FIA_ATD.1	User attribute definition
FIA_UAU.2	User authentication before any action
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_SMR.1	Security roles
FMT_SMF.1	Specification of management functions
FPT_RVM.1	Non-bypassability of the TSP
FTP_ITC.1	Inter-TSF trusted channel

Table 5-1: Functional Requirements on the TOE

The following paragraphs give an overview on the functional requirements listed in the table above with respect to the TOE. They serve as an introduction to the detailed definition of the functional requirements, which are presented in the next section.

Class FAU components are selected to describe the capability of the TOE to generate, read and protect audit data. The TOE generates audit data for events associated with the communication links it monitors. Administrators are able to select the audited events and to read the log file via the remote administration tool.

Class FCO contains requirements related to securing communication. In the case of the TOE only the NTP-packets are requested to have a security mechanism that allows to identify that those packets have been originated by a trusted time source. This is performed by a MD5 hash sum parametrised by a key. Only NTP-servers that possess this key are able to generate a NTP-packet that is accepted as valid by the

TOE.

Class FCS contains the requirements related to cryptographic operations. There are three areas in which the TOE performs cryptographic operations: Remote administration and peer configuration file exchange using TLS version 1; the VPN IPSEC connection which is also using TLS and SKUT for key management; and for the verification of incoming NTP packages. The TOE is importing certificates for the remote administration and for the VPN. MD5 keys are imported and used for the cryptographic signing and verification of the NTP packets.

Class FDP contains the security requirements associated with the access control between remote administrators/users/peer firewalls, and configuration data of the TOE or internal network resources. The class also contains the security requirements associated with the information flow control between the internal and external network interface of the TOE. Information flow control is enforced both by the TOE subsystems (i.e. the packet filter and the proxies) as well as by the VPN tunnel.

Class FIA contains the security requirements for identification and authentication of a remote administrator, performing administrative tasks or a peer firewall synchronising the configuration files. They are relying on the same mechanisms of TLS authentication using X.509 certificates.

Class FMT contains the security management requirements. This includes that the initial default values of the TOE will be well-defined.

Class FPT contains the requirement for the protection of the TSF that contains the requirements for non-bypassability which in the case of this TOE means mainly the non-bypassability of the filtering functions.

Class FTP contains requirements for trusted communication path between the TSF and other trusted IT products. This is the VPN connection.

The TOE implements three Security Function Policies (SFPs), these SFPs applies to information flow through the TOE, and can be used either separate (in parallel) or in combination (sequential) for traffic passing through the TOE.

UNAUTHENTICATED SFP

The TOE will implement a Security Function Policy (SFP) called UNAUTHENTICATED SFP. The TSF shall enforce the SFP on the unauthenticated external IT entities that send and/or receive data through the TOE for all traffic sent through the TOE from one subject to another. The policy is named UNAUTHENTICATED SFP to indicate that the subjects are entities on the external or internal network that can not be authenticated. So the information flow policy has to be based on the information that is to be passed through the firewall. The rules of the information flow policy are based on the direction of the information flow and the content of specific fields in the packet header or body, but are not able to check if the content of those fields has not been spoofed.

The TSF shall enforce the UNAUTHENTICATED SFP based on at least the following types of subject and information security attributes:

Subject security attributes

- IP-source and destination address

Information security attributes

- port number

- protocol type
- protocol specific rules.

The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold:

- The IP packet does belong to a protocol that is supported by the TOE and
- the protocol is one of those allowed to pass through the TOE from the receiving to the sending interface and
- the IP source and destination address are defined as being allowed to use the protocol and
- the protocol specific filter rules do not deny the information to flow.

The TSF shall explicitly deny an information flow based on the following rules:

- IP packets arriving on the internal interface where the IP source address is one assumed to be on the external network
- IP packets arriving on the external interface where the IP source address is one assumed to be on the internal network
- Malformed service requests, where the requests are not valid in the current protocol above the TCP layer
- IP packets in a wrong context, when an IP packet is not part of a valid TCP session

AUTHENTICATED SFP

The TOE will implement a Security Function Policy (SFP) called AUTHENTICATED SFP. The TSF shall enforce the SFP on the authenticated external IT entities that send and/or receive data using a VPN through the TOE. The policy is named AUTHENTICATED SFP to indicate that connections are between entities that are authenticated.

The TSF shall enforce the AUTHENTICATED SFP for IP packets arriving on the internal interface based on at least the following types of subject and information security attributes:

Subject security attributes:

- IP-networks to be reaching using the AUTHENTICATED SFP

Information security attributes:

- port number
- protocol type.

The TSF shall explicitly route all traffic using the AUTHENTICATED SFP based on the following rule:

- IP packets arriving on the internal interface where the IP destination address is part of the IP-network to be reached using the AUTHENTICATED SFP

No other traffic shall be routed using the AUTHENTICATED SFP.

AUTHENTICATED USER ACCESS SFP

The TOE implements an access policy called Authenticated User Access SFP. The TOE shall enforce identification and authentication of administrators and peer

firewalls to access TSF data, and users requesting access to the internal network, by mutual authentication.

5.1 TOE Security Functional Requirements

This section presents the security functional requirements (SFRs) chosen from part two of the Common Criteria. Note that references to standards have been made throughout this section to claim compliance with specific parts of the referenced standards, as indicated in the SFR text. The compliance claim is restricted to the standard parts specific to the subject of the SFR, and does not imply a compliance claim to the referenced standard as a whole.

5.1.1 Class FAU - Security Audit

5.1.1.1 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) Generic events:

GEN-1: **Reconfiguration with reboot**

GEN-2: **IP-packets received on ports without active listeners**

GEN-3: **Client attempts to connect to a proxy without being authorised to do so.**

GEN-4: **Client attempts to set up a connection to a server without being authorised to connect to this server using the specified protocol.**

GEN-5: **Errors parsing the configuration file**

GEN-6: **<not used>**

GEN-7: **Connections allowed by a proxy**

Specific for FTP-connections:

FTP-1: **Protocol violation (e.g. reply too long)**

FTP-2: **Proxy received a command that is not implemented and therefore has been denied**

FTP-3: **Proxy received a command that requires a data channel to be opened when the channel was not opened before**

FTP-4: **Proxy received a command that has been disabled in the configuration file**

FTP-5: **Syntax error in command**

FTP-6: **Unsupported command parameter specified**

Specific for SMTP-connections:

SMTP-1: **Data received exceeded the maximum size specified in the configuration file**

SMTP-2: Maximum reply line length exceeded

SMTP-3: Forward path rejected

SMTP-4: Recipient domain rejected

SMTP-5: Bad command sequence

Specific for the DNS-connections:

DNS-1: DNS queries with domain names longer than specified

DNS-2: Domain name longer than configured reject / warning limit

DNS-3: A query containing an internal domain name was detected

DNS-4: The query was accepted

DNS-5: The answer was accepted

DNS-6: The response contained an internal domain name / network address

DNS-7: An invalid DNS record was received

DNS-8: Failure to match response against queries

Plug proxy:

PLUG-1: Access controls

PLUG-2: TLS authentication

PLUG-3: TLS access control

Packet screening daemon:

PSD-1: NTP packet too short

PSD-2: Keynum missing

PSD-3: MD5 error

Remote administration:

REM-1: Changes to the TOE configuration done by a remote administrator (reboot not necessary)

REM-2: Remote user login

VPN-system (vpnd):

VPND-1: HMAC mismatch in ESP mode

VPND-2: invalidate: not on local interface

VPND-3: Unknown SPI from unknown host

VPND-4: Unknown SPI from host

VPND-5: new local key

VPND-6: new remote key

VPN-system (skutd)

SKUTD-1: Verify error

SKUTD-2: **Access denied**

SKUTD-3: **Verify error:num**

SKUTD-4: **No user certificate**

SKUTD-5: **Wrong CN**

SKUTD-6: **SSL_accept**

SKUTD-7: **returning spi and accepting push spi since NAT detected**

SKUTD-8: **returning spi**

Remote administration system (admind)

ADMIND-1: **SSL_accept error**

ADMIND-2: **Authorised administrator**

ADMIND-3: **Audit records of configuration synchronisation**

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 ST: **none**

Application note: Syslog messages are not created during local administration, because the syslog daemon is not working in single-user mode.

5.1.1.2 FAU_SAR.1 - Audit Review

FAU_SAR.1.1 The TSF shall provide **administrators** with the capability to read **all data** 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.

Application note: The TSF shall provide the remote administrator with the capability to read all audit records. The capability for local administrators to read audit records is part of the TOE environment.

5.1.1.3 FAU_SEL.1 - Selective Audit

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

a) **event type**

b) **rejected IP-packets with following attributes: arriving at a specific interface, having a specific port number, having a specified IP-source address, having a specific IP-destination address.**

Application note: The event type selected can only be ICMP packet.

5.1.2 Class FCO - Communication

5.1.2.1 FCO_NRO.1 – Selective proof of origin

FCO_NRO.1.1 The TSF shall be able to generate evidence of origin for transmitted **NTP-packets** at the request of the **recipient**.

FCO_NRO.1.2 The TSF shall be able to relate the **key** of the originator of the information, and the **keyed MD5 hash sum** of the information to which the evidence applies.

FCO_NRO.1.3 The TSF shall provide a capability to verify the evidence of origin of information to **the responsible user daemon process** given **that different NTP servers use different keys for the generation of the keyed MD5 hash sum**.

Application note: The recipient of FCO_NRO.1.1 is the IP stack.

5.1.3 Class FCS – Cryptographic Support

5.1.3.1 FCS_CKM.1a Cryptographic key generation (admin)

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **as defined in the TLS v1 standard for 3DES-EDE-CBC and SHA keys** and specified cryptographic key sizes **168 and 160 bit** that meet the following: **generation and exchange of session keys as defined in the TLS v1 standard with the cipher suites defined in FCS_COP.1b and FCS_COP.1c**.

Application note: The session keys are negotiated and established during an TLS session for remote administration. The TLS standard allows other cryptographic algorithms and key sizes, but only 3DES EDE-CBC and SHA is supported. This functionality is provided by a TLS server on the server side. The web browser provides corresponding functionality on the TLS client's side (as part of the environment).

The key destruction of session keys as specified by CC in FCS_CKM.4 is covered by FDP_RIP.2 Full residual information protection.

5.1.3.2 FCS_CKM.1b Cryptographic key generation (SKUT)

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **as defined in the TLS v1 standard for 3DES-EDE-CBC, AES (TLS v.1.1) and SHA keys** and specified cryptographic key sizes **168 or 256 and 160 bit respectively** that meet the following: **generation and exchange of session keys and master secrets as defined in the TLS v1 and TLS v1.1 standard with the cipher suites defined in FCS_COP.1e and FCS_COP.1f**.

Application note: The keys are negotiated and established by the TLS session and used by the VPN for symmetrical encryption and integrity protection of VPN packages. The TLS standard allows other cryptographic algorithms and key sizes, but only 3DES EDE-CBC, AES and SHA are supported.

The key destruction of session keys as specified by CC in FCS_CKM.4 is covered by FDP_RIP.2 Full residual information protection.

5.1.3.3 FCS_CKM.2a - Cryptographic Key Distribution (admin cert)

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method **digital certificates** that meets the following: **X.509 Version 3**.

Application note: This requirement addresses the exchange of X.509 certificates as part of the TLS authentication of the remote administrators and peer firewalls.

5.1.3.4 FCS_CKM.2b - Cryptographic Key Distribution (admin keys)

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method **TLS handshake using RSA encrypted exchange of 3DES session keys and SHA keys** that meets the following: **TLS v1 (RFC 2246)**.

Application note: This requirement addresses the exchange of 3DES session keys and SHA keys as part of the TLS handshake protocol for remote administration and peer configuration.

5.1.3.5 FCS_CKM.2c – Cryptographic Key Distribution (SKUT cert)

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method **digital certificates** that meets the following: **X.509 Version 3**.

Application note: This requirement addresses the exchange of X.509 certificates as part of the TLS authentication, used for the establishing the key management of the VPN channel.

5.1.3.6 FCS_CKM.2d - Cryptographic Key Distribution (SKUT keys)

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method **TLS handshake using RSA encrypted exchange of 3DES and AES session and SHA keys** that meets the following: **TLS v1 (RFC 2246) and TLS v.1.1 (RFC 4346)**.

Application note: This requirement addresses the exchange of TLS keys as part of the TLS handshake protocol for the key management part of the VPN.

5.1.3.7 FCS_COP.1a - Cryptographic Operation (admin RSA)

FCS_COP.1.1 The TSF shall perform **digital signature generation and verification** in accordance with a specified cryptographic algorithm **RSA** and cryptographic key sizes **1024 bit** that meet the following: **TLS v1 (RFC 2246)**.

Application note: This requirement addresses the RSA digital signature generation and verification operations using the RSA algorithm as required by the TLS session establishment protocol. Details of the signature format are defined in the TLS v1

standard.

5.1.3.8 FCS_COP.1b - Cryptographic Operation (admin 3DES)

FCS_COP.1.1 The TSF shall perform **encryption and decryption** in accordance with a specified cryptographic algorithm **3DES** and cryptographic key sizes **168 bit** that meet the following: **TLS v1 (RFC 2246) and the following cipher suites: TLS_RSA_WITH_3DES_EDE_CBC_SHA as defined in the TLS v1 standard**.

Application note: The TLS v1 standard allows other ciphers, but the TOE supports only 3DES-EDE-CBC. If a client tries to use any other cipher suite, the client will be rejected by the TOE.

5.1.3.9 FCS_COP.1c - Cryptographic Operation (admin SHA)

FCS_COP.1.1 The TSF shall perform **message digest generation and verification** in accordance with a specified cryptographic algorithm **SHA** and cryptographic key sizes **160 bit** that meet the following: **TLS v1 (RFC 2246) and the following cipher suites:**

TLS_RSA_WITH_3DES_EDE_CBC_SHA as defined in the TLS v1 standard.

Application note: The TLS v1 standard allows other ciphers, but the TOE supports only SHA. If a client tries to use any other another cipher suite for the message digest, the client will be rejected by the TOE.

5.1.3.10 FCS_COP.1d - Cryptographic Operation (SKUT RSA)

FCS_COP.1.1 The TSF shall perform **digital signature generation and digital signature verification** in accordance with a specified cryptographic algorithm **RSA** and cryptographic key sizes **1024 bit** that meet the following: **TLS v1 (RFC 2246) and TLS v1.1 (RFC 4346)**.

Application note: This requirement addresses the RSA digital signature generation and verification operations using the RSA algorithm as required by the TLS session establishment protocol. Details of the signature format are defined in the TLS version 1 standard. This TLS session is part of the key TLS protocol used for the VPN key negotiation.

5.1.3.11 FCS_COP.1e – Cryptographic Operation (SKUT 3DES/AES)

FCS_COP.1.1 The TSF shall perform **encryption and decryption** in accordance with a specified cryptographic algorithm **3DES or AES** and cryptographic key sizes **168 bit or 256 bit respectively** that meet the following: **TLS v1 (RFC 2246) and TLS v1.1 (RFC4346), and the following cipher suites:**

TLS_DH_RSA_WITH_3DES_EDE_CBC_SHA or TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in the TLS v1 and TLS v1.1 standard, respectively.

Application Note: The TLS v1 standard allows other ciphers, but the TOE supports only 3DES, AES and SHA. These cipher suites are also the only ones which are supported by the TLS-proxy. If the other VPN node tries to use any other cipher suite for the message digest, the client will be rejected by the TOE.

5.1.3.12 FCS_COP.1f – Cryptographic Operation (SKUT SHA)

FCS_COP.1.1 The TSF shall perform **message digest and verification** in accordance with a specified cryptographic algorithm **SHA** and cryptographic key sizes **160 bit** that meet the following: **TLS v1 (RFC 2246) and TLS v1.1 (RFC 4346), and the following cipher suites: TLS_DH_RSA_WITH_3DES_EDE_CBC_SHA and TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in the TLS v1 and TLS v1.1 standard, respectively**.

Application note: The TLS v1 standard allows other ciphers, but the TOE supports only SHA. If the other VPN node tries to use any other cipher suite for the message digest, the client will be rejected by the TOE.

5.1.3.13 FCS_COP.1g – Cryptographic Operation (IPSEC 3DES/AES)

FCS_COP.1.1 The TSF shall perform **encryption and decryption** in accordance with a specified cryptographic algorithm **3DES or AES** and cryptographic key sizes **168 bit or 256 bit respectively** that meet the following: **IPSEC (RFC 4301 RFC 2406 and RFC 3602) and the following cipher suites 3DES and AES as defined in the IPSEC standard**.

Application Note: The IPSEC standard allows other ciphers, but the TOE supports only 3DES and AES.

5.1.3.14 FCS_COP.1h – Cryptographic Operation (IPSEC SHA)

FCS_COP.1.1 The TSF shall perform **message digest and verification** in accordance with a specified cryptographic algorithm **SHA** and cryptographic key sizes **160 bit** that meet the following: **IPSEC (RFC 4301, RFC 2406 and RFC 3602) and the following cipher suites HMAC-SHA1 as defined in the IPSEC standard**.

Application note: The IPSEC standard allows other ciphers, but the TOE supports only SHA1.

5.1.3.15 FCS_COP.1i - Cryptographic Operation (MD5)

FCS_COP.1.1 The TSF shall perform **verification of cryptographic checksums (HMAC)** in accordance with a specified cryptographic algorithm **MD5** and cryptographic key sizes **(variable)** that meet the following: **requirements of HMAC message authentication with MD5**.

Application note: This requirement is only added to support FCO_NRO.1, which aims at authenticity checks for NTP packets. For the verification of HMAC with MD5 hash sums, the strength of function SOF-high is claimed and substantiated in

the rationale part of the ST.

5.1.4 Class FDP - User Data Protection

5.1.4.1 FDP_ACC.2 – Complete access control

FDP_ACC.2.1 The TSF shall enforce the **AUTHENTICATED USER ACCESS SFP** on **the subjects:**

- **remote administrators**
- **peer firewalls**
- **remote users**

and objects:

- **configuration data of the TOE**
- **resources in the internal network**

and all operations among subjects and objects covered by the SFP.

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

5.1.4.2 FDP_ACF.1 – Security attribute based access control

FDP_ACF.1.1 The TSF shall enforce the **AUTHENTICATED USER ACCESS SFP** to objects based on the following:

subject remote administrator, peer firewall:

- **TLS certificate**
- **UID field of the certificate**

subject remote user:

- **TLS certificate**
- **CN field of the certificate**
- **UID field of the certificate**

objects (configuration data of the TOE, resources in the internal network):

- **none.**

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

- **If the UID field of the subject's certificate is part of a list managed by the TOE that allows remote administration, then the remote administrator is allowed to administrate the TOE.**
- **If the UID field of the subject's certificate is part of a list managed by the TOE that allows remote administration, then the peer firewall is allowed to**

administrate the TOE.

- **If the CN field and/or the UID field of the subject's certificate is part of a list managed by the TOE that allows to connect as TLS client to the TOE, the user is allowed access to resources in the internal network.**

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **none**.

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the:

If the client certificate is not signed by a certificate of a certification authority trusted by the TOE, then access is denied.

Application note: Several other configuration settings of the TOE can further restrict access to the TOE or to the internal network, but these are optional and not considered as security attributes/security functions (e.g., remote administration only from internal network, remote administration allowed/not allowed).

5.1.4.3 FDP_IFC.1a - Subset information flow control (unauthenticated sfp)

FDP_IFC.1.1 The TSF shall enforce the **UNAUTHENTICATED SFP** on **the unauthenticated external IT entities that send and/or receive data through the TOE for at least all traffic not subject to the AUTHENTICATED SFP sent through the TOE from one subject to another.**

Application note: The UNAUTHENTICATED SFP (i.e. the proxies) is enforced for all traffic that are not subject to the AUTHENTICATED SFP. The UNAUTHENTICATED SFP may also apply to traffic subject to the AUTHENTICATED SFP (i.e. going through the VPN).

5.1.4.4 FDP_IFC.1b – Subset information flow control (authenticated sfp)

FDP_IFC.1.1 The TSF shall enforce the **AUTHENTICATED SFP** on **the external IT entities that send and/or receive data through the TOE for at least all traffic not subject to the UNAUTHENTICATED SFP sent through the TOE from one subject to another.**

Application note: The AUTHENTICATED SFP (i.e. the VPN) is enforced for all outgoing and incoming traffic to and from dedicated networks. The AUTHENTICATED SFP may also be subject to the UNAUTHENTICATED SFP.

5.1.4.5 FDP_IFF.1a – Simple security attributes (unauthenticated sfp)

FDP_IFF.1.1 The TSF shall enforce the **UNAUTHENTICATED SFP** based on the following types of subject and information security attributes:

» **subject security attributes:**

- **IP-source and destination address**

» **information security attributes:**

- **port number**

- **protocol type**
- **protocol-specific rules.**

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:

» **The IP packet does belong to a protocol that is supported by the TOE and**

» **the protocol is one of those allowed to pass through the TOE from the receiving to the sending interface and**

» **the IP source and destination address are defined as being allowed to use the protocol and**

» **the protocol specific filter rules do not deny the information to flow.**

FDP_IFF.1.3 The TSF shall enforce **no additional information flow control SFP**.

FDP_IFF.1.4 The TSF shall provide the following **additional SFP capabilities: none**.

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:

» **IP packets arriving on the internal interface where the IP source address is one assumed to be on the external network**

» **IP packets arriving on the external interface where the IP source address is one assumed to be on the internal network**

» **Malformed service requests, where the requests are not valid in the current protocol above the TCP layer**

» **IP packets in a wrong context, when an IP packet is not part of a valid TCP session**

5.1.4.6 FDP_IFF.1b - Simple security attributes (authenticated sfp)

FDP_IFF.1.1 The TSF shall enforce the **AUTHENTICATED SFP** based on the following types of subject and information security attributes:

» **subject security attributes:**

- **IP-source and destination address**

» **information security attributes:**

- **port number**
- **protocol type**

Application note: This is the inter-TSF trusted channel (IPSEC) described in

FTP_ITC.1. It is authenticated and encrypted with integrity protection.

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:

- **the destination address is part of a VPN reachable network**
- **the port number used is allowed for the VPN connection**
- **the protocol type is permitted**

FDP_IFF.1.3 The TSF shall enforce **no additional information flow control SFP**.

FDP_IFF.1.4 The TSF shall provide the following **additional SFP capabilities: none**

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 destination address not part of a VPN reachable network**
- **the port number is not allowed for the VPN connection**
- **the protocol type is not permitted.**

5.1.5 Class FIA - Identification and Authentication

5.1.5.1 FIA_ATD.1 – User Attribute Definition

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

- **identity of the administrator or user**
- **association of the administrator or user with a TLS client certificate.**

Application note: In case of synchronisation of configuration information between TOEs, one peer will have to authenticate with any other peer. This means that the "administrator" in this case will be the other firewall. Because of synchronisation, the identity of remote firewall and the TOE will be identical.

5.1.5.2 FIA_UAU.2 – User Authentication before any Action

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

Application note: Remote authentication is performed by verifying that the administrator or peer firewall possesses the private key part to the TLS client certificate that has been issued to the administrator or peer firewall, while users are authenticated to the TCP plug-proxy using their TLS certificates.

5.1.5.3 FIA_UID.2 – User Identification before any Action

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

Application note: Remote administrators, users or peer firewalls identify themselves by presenting an TLS client certificate.

5.1.6 Class FMT - Security Management

5.1.6.1 FMT_MOF.1 - Management of security functions behaviour

FMT_MOF.1.1 The TSF shall restrict the ability to **modify the behaviour of** the functions **listed below** to **an authorised administrator**:

- **restart the system**
- **administrate the email-queue**
- **change the configuration of the TOE:**
  - **change administrator information (email and postmaster)**
  - **change machine (device names and IP-addresses for the network interfaces)**
  - **change remoteadmin (enable/disable, specify interfaces, administrators, port and service name)**
  - **change inside (specifics of the inside interface)**
  - **change outside (specifics of the outside interface)**
  - **change proxydns**
  - **change log (configuration of audit file transfer to other machines)**
  - **change ipfilter (specification of traffic not to be logged)**
  - **change ntp configuration**
  - **change smtp configuration**
  - **change ftp-gw configuration**
  - **change plug-gw**
  - **change vpn**
  - **change failover functionality**

Application note: The TOE also supports the access control provided by its environment (local FreeBSD login) to authorise an environment administrator to modify the behaviour of the functions mentioned above.

5.1.6.2 FMT_MSA.1 - Management of security attributes

FMT_MSA.1.1 The TSF shall enforce the **AUTHENTICATED USER ACCESS SFP** to restrict the ability to **modify** the security attributes **consisting of possible configuration options** to

authorised administrators.

Application note: The TOE also supports the access control provided by its environment (local FreeBSD login) to authorise an environment administrator to modify the behaviour of the functions mentioned above.

5.1.6.3 FMT_MSA.3 – Static attribute initialisation

FMT_MSA.3.1 The TSF shall enforce the **UNAUTHENTICATED SFP, AUTHENTICATED SFP, and AUTHENTICATED USER ACCESS SFP** to provide **other property** default values for security attributes that are used to enforce the SFP.

Application note: The default security attributes stated as "other property" are well-defined.

FMT_MSA.3.2 The TSF shall allow the **authorised administrator** to specify alternative initial values to override the default values when an object or information is created.

Application note: An administrator can restrict unauthenticated access and specify security relevant initial values by changing the rules in the configuration file.

5.1.6.4 FMT_MTD.1 – Management of TSF data

FMT_MTD.1.1 The TSF shall restrict the ability to **query or modify** the **TSF data listed below** to **an authorised administrator:**

- **audit data [query]**
- **status of the system (NTP, processes, disk space and network connections) [query]**
- **status of the VPN-tunnels [query]**
- **configuration files [query, modify].**

Application note: The TOE also supports the access control provided by its environment (local FreeBSD login) to authorise an environment administrator to modify the behaviour of the functions mentioned above.

5.1.6.5 FMT_SMF.1 – Specification of management functions

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

- **apply changes to the central configuration file**
- **reboot the firewall**
- **query the audit files**
- **inspect the mail queue**
- **inspect the system status and**
- **perform basic backup operations.**

Application note: The security management functions related to changes of the configuration of the TOE are described in more detail in FMT_MOF.1.

5.1.6.6 FMT_SMR.1 - Security roles

FMT_SMR.1.1 The TSF shall maintain the roles:

» **administrator**

» **user**

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

Application note: The administrator certificate for remote administrator login must be associated with the administrator role maintained by the TOE's underlying environment. The peer administrator uses the remote administrator certificate for authentication and does not introduce any new role. The user role is limited to the authenticated users, which means only those users who have been given a TLS client certificate to be given access through a TCP plug-proxy. There may also be other users which the TOE is not aware as individual users.

5.1.7 Class FPT - Protection of the TOE Security Functions

5.1.7.1 FPT_RVM.1 – Non-Bypassability of the TSP

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

Application note: Especially the TOE shall ensure that no communication between the two network interfaces is possible that is not mediated by the TOE functions.

5.1.8 Class FTP - Trusted path/channels

5.1.8.1 FTP_ITC.1 Inter-TSF Trusted Channel (IPSEC)

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

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

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for **VPN services.**

Application note: This channel is the VPN communication channel (IPSEC) the TOE may establish with other Färist.

5.2 TOE Security Assurance Requirements

The target assurance components for this TOE are those defined in the EAL4 level of the Common Criteria augmented with ALC_FLR.1, basic flaw remediation. The Common Criteria authors have ensured that EAL4 is a sound selection of assurance components where all dependencies have been resolved. Since the augmentation of ALC_FLR.1 does not have any dependencies, there is no need to verify the consistency of the assurance component selection.

The following table provides an overview of the assurance components that form the EAL4 assurance level of the Common Criteria, augmented with ALC_FLR.1, basic

flaw remediation:

Assurance class	Assurance components
Configuration management	ACM_AUT.1 Partial CM automation
	ACM_CAP.4 Generation support and acceptance procedures
	ACM_SCP.2 Problem tracking CM coverage
Delivery and operation	ADO_DEL.2 Delivery procedures
	ADO_IGS.1 Installation, generation, and start-up procedures
Development	ADV_FSP.2 Fully defined external interfaces
	ADV_HLD.2 Security enforcing high-level design
	ADV_IMP.1 Evaluation of implementation representation
	ADV_LLD.1 Evaluation of low-level design
	ADV_RCR.1 Informal correspondence demonstration
	ADV_SPM.1 Evaluation of security policy modelling
Guidance and Documentation	AGD_ADM.1 Administrator guidance
	AGD_USR.1 User guidance
Life cycle	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
Tests	ATE_COV.2 Analysis of coverage
	ATE_DPT.1 Testing: high-level design
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent testing - sample
Vulnerability assessment	AVA_MSU.2 Validation of Analysis
	AVA_SOF.1 Strength of TOE security function evaluation
	AVA_VLA.2 Independent vulnerability analysis

Table 5-2: Functional Requirements on the TOE

5.3 Security Functional Requirements for the IT Environment

The Färist relies on its environment to enforce certain security functionality. The requirements to operate the firewall securely in a reliable environment are listed below.

Component	Component name
FAU_SAR.1	Audit review
FCS_CKM.1	Cryptographic key generation
FCS_CKM.2	Cryptographic key distribution
FDP_RIP.2	Full residual information protection
FDP_SDI.1	Stored data integrity monitoring
FPT_STM.1	Reliable time stamps

Table 5-3: Security Functional Requirements for Environment

Class FAU components are selected for the capability of the TOE environment to allow administrators to read audit data. Administrators are able to select the audited events and to read the log file via local administration.

Class FCS contains the security requirements associated with cryptographic operations. The TOE environment must be able to generate key pairs and X.509 certificates for the remote administration and peer firewalls as well as MD5 keys for the NTP verification. These keys must be manually distributed to the administrator or the TOE.

Class FDP deals with the user data protection. The environment has to ensure that user data temporarily stored in volatile memory is not accessible to unauthorised instances and that any data is protected against unauthorised modification. Requirements on the import of certificates, private keys and the MD5 key for the verification of NTP packets are also made.

Class FPT contains requirements for non-bypassability which in the case of the TOE environment means the requirement for reliable time stamps. Those time stamps are available for the generation of audit events, provided to the TOE environment via the NTP protocol.

5.3.1 Class FAU - Security Audit

5.3.1.1 FAU_SAR.1 - Audit Review

FAU_SAR.1.1 The TSF shall provide **administrators** with the capability to read **all data** 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.

Application note: the IT environment must provide local administrators with the necessary means to allow them to read and interpret the audit records.

5.3.2 Class FCS - Cryptographic Support

5.3.2.1 FCS_CKM.1a Cryptographic key generation (admin and VPN cert)

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **suitable for the generation of RSA key pairs** and specified cryptographic key sizes **1024 bit** that meet the following: **requirements of the TLS v1 standard for client and server certificates**.

Application note: RSA key pairs are generated within the TOE environment. One key pair is used for both remote administration, peer firewall configuration and all VPN-channels. This Security Target does not define the exact key generation algorithm for the generation of RSA key pairs, but leaves this to the environment. The keys generated have to satisfy assumption A.AUTKEY defined in chapter 3.1.

5.3.2.2 FCS_CKM.1b Cryptographic key generation (MD5 key)

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **that**

satisfies assumption A.MD5KEY and specified cryptographic key sizes **128 to 256 bit** that meet the following: **requirements of the MD5 protocol for generation and verification of cryptographic hash sums**.

Application note: The considered keys are generated and distributed in the TOE environment. The TOE expects assumption A.MD5KEY as well as the requirement on the key size to be satisfied by the key generation mechanism and distribution process.

5.3.2.3 FCS_CKM.2 Cryptographic key distribution

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method**, manual distribution to the administrator,** that meets the following: **no standards apply, but integrity and confidentiality of the key have to be ensured**.

Application note: The certificates for TLS key management, TLS remote administration and peer firewall configuration must, along with the MD5 key be distributed manually to the administrator.

5.3.3 Class FDP – User Data Protection

5.3.3.1 FDP_RIP.2 - Full residual information protection

FDP_RIP.2.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the **allocation of the resource to** all objects.

Application note: this includes any information associated with users using the TOE such as IP addresses or other control information; data transmitted through the TOE; authentication information or session keys used by administrator sessions.

5.3.3.2 FDP_SDI.1 - Stored Data Integrity Monitoring

FDP_SDI.1.1 The TSF shall monitor user data stored within the TSC for **modifications** on all objects, based on the following attributes:

» **file attributes (i.e. size, creation/modification date, owner, read, write and execute flags).**

Application note: There is no user data stored on the TOE or its environment but this refers to any executables, configuration information or administrative data that is stored. The functionality of this requirement is provided by use of the tool Tripwire.

5.3.4 Class FPT - Protection of the TOE Security Functions

5.3.4.1 FPT_STM.1 – Reliable Time Stamps

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps for its own use.

Application note: The time stamps are generated by the internal clock of FreeBSD as part of the TOE environment, which is synchronised by the use of authenticated

NTP-packets with trusted external time sources.

6 TOE Summary Specification

The TOE Summary Specification provides a complete high-level definition of the security functions and assurance measures of the TOE and their relationship to the security functional and assurance requirements of this ST.

The TOE Summary Specification identifies the security functions that the TOE implements to meet the requirements defined in chapter 5 of the security target.

An SOF claim is made for the security function SF.PSD, since it contains a mechanism using probabilistic and permutational mechanisms resp. (see section 1.4).

6.1 TOE Security Functions

This chapter describes the IT security functions of the TOE and their relation to the security functional requirements which they are supposed to meet.

A mapping of security functions against requirements is provided in clause 8.3 of the rationale part.

6.1.1 SF.IPS – IP Stack

The IP stack is the first and central instance of the TOE to receive incoming IP packets and is responsible for handling these packets to the other security functions for further inspection. The IP stack ensures that no IP packet is forwarded from one to another external interface of the TOE without being inspected by the packet filter, screening daemon and protocol specific proxies or the VPN-system.

After accepting a package the security function SF.IPS determines which of the information flow SFPs applies (UNAUTHENTICATED SFP or AUTHENTICATED SFP), and routes the package accordingly.

The IP stack generates log data for audit events and delivers it to the TOE environment.

6.1.2 SF.PF – Packet filter

The packet filter is the first instance called by the IP stack to enforce the UNAUTHENTICATED SFP. All IP packets entering and leaving the IP protocol stack are examined and checked against defined rules, before they are passed on to other security functions of the TOE.

The packet filter performs its checks based on

- the interface where the packet arrived,
- IP source and destination addresses,
- port source and destination number (in case of UDP and TCP packets),
- ICMP type, if applicable and
- TCP flags, if applicable.

The packet filter generates log data in case of audit events and delivers it to the TOE environment. The packet filter is able to include or exclude events from being logged based on configured attributes.

6.1.3 SF.PSD – Packet Screening Daemon

The packet screening daemon is invoked after packets have passed successfully the packet filter and examines NTP packets.

The size of **NTP packets** is checked and afterwards the MD5 hash sum of the packet is calculated with the locally stored key and compared to the hash sum found in the IP packet.

The packet screening daemon delivers log data of audit events to the syslog facility of the IT environment.

Note: For the mechanism used to implement the verification of MD5 hash sums, the strength of function SOF-high is claimed.

6.1.4 SF.AP – Application proxies

The application proxies of the TOE are the

- FTP proxy,
- SMTP proxy,
- DNS proxy and,
- TCP plug proxy.

For the according high (application) level protocols, these proxies implement protocol-specific filter rules which can be configured by the administrator via the appropriate interfaces.

The proxies generate log data of audit events and deliver it to the environmental syslog facility.

6.1.5 SF.KEYMAN – VPN Key Management

The VPN may operate with fixed keys, agreed in advance between the end points of the VPN channel, or with the SKUT key management negotiating the keys between the two Färists. SKUT key management must be used in the evaluated configuration.

SF.KEYMAN is a security function separate from the VPN channel with the following functions:

- establish a TLS channel with the peer Färist and authenticate the peer Färist during this process with a X.509 certificate.
- generate a key pair consisting of a 168 bit 3DES key or 256 bit AES key and a 160 bit SHA key
- negotiate and exchange the key pair with the peer Färist.

The key pair will subsequently be used by the VPN function (SF.VPNCH).

The VPN key management system generates log data of audit events and deliver it to the environmental syslog facility.

6.1.6 SF.VPNCH – VPN Channel

The channel established is a two-way communication channel, where both parts are mutually authenticated using the their own SHA key. All outgoing packages are integrity protected using SHA, and 3DES or AES encrypted with keys that are either pre-shared or generated using the SKUT key management functions being part of

the SF.KEYMAN function. The keys must be provided by SF.KEYMAN using SKUT in the evaluated configuration.

All incoming packages are decrypted using 3DES or AES with the keys provided by SF.KEYMAN and integrity verified using SHA.

In short the SF.VPNCH provides the following security functionality:

- Authentication of the other end-point
- Encryption and integrity protection of all outgoing packages
- Decryption and integrity verification of all incoming packages

The VPN system generates log data of audit events and deliver it to the environmental syslog facility.

6.1.7 SF.LCONF – Local configuration tools

Local configuration at the console of the underlying system (FreeBSD) the TOE runs on is enabled via an administration interface, representing the administration tools. The administrator, authorised by the TOE environment, is able to

- apply modifications to the central configuration file,
- update the TOE system files based on the configuration file modifications
- initiate backup and restore of configuration file (environment),
- change passwords (environment) and
- manage TLS certificates (environment).

If new security functions are configured (e.g. a new plug proxy), the system provides the administrator with a well-defined default setting - every connection not explicitly set to be allowed will be rejected by the TOE.

Furthermore, the local administration tools provide the backend tools to generate the configuration files out of the central configuration file edited by a local or remote administrator. In a failover configuration, the configuration file may also be obtained from a firewall peer that has a newer version of the configuration file. However, any change to a configuration file originates from the administrator changes and will only be replicated in peer firewalls.

6.1.8 SF.RCONF – Remote configuration tools

Remote configuration of the TOE is enabled via a web application, accessible by clients over the HTTP protocol. The administrator, authorised by the TOE, is able to

- apply changes to the central configuration file,
- reboot the firewall,
- query the logs
- inspect the mail queue,
- inspect the system status and
- perform basic backup operations.

If new security functions are configured (e.g., a new plug proxy), the system provides the administrator with well-defined default settings - every connection that is not explicitly set to be allowed will be rejected by the TOE.

The web application generates log data of audit events and delivers it to the environmental syslog facility.

6.1.9 SF.RCHIN – Remote Communication Channel

Remote administration and configuration is performed via a remote communication channel using mutual authentication based on X.509 certificates. Users may also use the same functionality to be given access to certain TCP plug-proxy ports. The remote communication channel will maintain integrity protection and do encryption using SHA and 3DES provided by an TLS v1. On the TOE side this is done by an HTTPS server(in case of remote administration or configuration) or as TLS using any port (in case for user authentication), supporting TLS. For remote synchronisation the client is a peer obtaining a configuration file or, in the case of remote administration, part of the TOE environment. The client may also be a user going through a TCP plug-proxy (using the the same cipher suites as supported for VPN Key Management and VPN channels: TLS_RSA_WITH_3DES_EDE_CBC_SHA and TLS_DHE_RSA_WITH_AES_256_CBC_SHA).

The https server generates log data of audit events and deliver it to the environmental syslog facility.

6.1.10 SF.RCHOUT – Remote Communication Channel

Configuration updates in a failover configuration are done between the failover peers and initiated by the peer with a configuration file that needs to be updated. This peer will then act as a client in remote administration, by establishing a remote configuration channel to the peer with a more recent configuration. This means that SF.RCHOUT is the client part and SF.RCHIN is the server part in establishing a remote communication in a peer to peer configuration channel. This channel is established using mutual authentication based on X.509 certificate.

The remote communication channel will maintain integrity protection and do encryption using SHA, and 3DES provided by an TLS v1. On the TOE side this is done by configd, which is supporting TLS. The client part is another firewall identical to the TOE, relying on the security functionality provided by SF.RCHIN.

The system responsible for configuration updates generates log data of audit events and deliver it to the environmental syslog facility.

6.2 TOE Assurance Measures

This chapter gives information about the measures the developer has taken to achieve the desired EAL 4 augmented assurance level. Because the TOE security assurance requirements are exclusively based on ISO/IEC 15408 assurance components, we only provide a reference to the documents that show that the assurance requirements are met (see the application notes to ASE_TSS.1-1).

6.2.1 AM.CONFIG – Configuration management

The configuration management tool CVS is used to manage the configuration items of the TOE. The manual of the CVS tool and the procedures for using it are documented in separate documents.

The TOE is referenced by unique version numbers and is labeled with its reference. A CM documentation is provided (see document “Färisten Configuration

Management"). The CM tool is used to provide automated support for generating the TOE from its implementation representation as well as measures for authorised changes to configuration items. It provides unique identification of each configuration item.

The CM system, as documented, tracks the TOE implementation representation, design documentation, test documentation, user documentation, administrator documentation and CM documentation.

The CM documentation provides an acceptance plan describing the acceptance criteria and the process for accepting change into the TOE and releases of the TOE.

All evaluation evidence is under the control of the CM system.

This assurance measure meets the requirements ACM_AUT.1, ACM_CAP.4 and ACM_SCP.2.

6.2.2 AM.DEL – Delivery and operations

The procedures for delivery of the TOE to the user can be found in the document "Färisten Delivery Procedures". Necessary steps for the secure installation, generation and start-up of the TOE are documented in the "Färisten Administrators Manual".

This assurance measures meet the requirements ADO_DEL.2, ADO_IGS.1 and AVA_MSU.2.

6.2.3 AM.DEVEL – Development

The developer provides the functional specification together with a security enforcing high-level design in the documents for each subsystem:

- "Färisten Firewall Base System",
- "Färisten Firewall Proxies",
- "Färisten Firewall Packet Filter",
- "Färist VPN and Crypto Subsystem",
- "Simple Key-exchange Using TLS (SKUT)",
- "FMSSL 2.0 Implementation",
- "Färisten Firewall Administration Tools",
- "Färisten Failover System".

There are separate low-level design documentation being automatically maintained for all parts of the TOE. It is generated out of the implementation representation and therefore provides both the low-level design and the implementation representation for those parts.

An informal correspondence analysis between the security target TOE summary specification, the functional specification and high-level design, and the low-level design and implementation representation, is given in the document "Färisten Design Correspondence Analysis".

A separate document is describing the Security Policy Model.

This assurance measure meets the requirements of ADV_FSP.2, ADV_HLD.2, ADV_IMP.1, ADV_LLD.1, ADV_SPM.1, and ADV_RCR.1.

6.2.4 AM.GUIDE – Guidance documents

Administrator guidance for the TOE is provided in the document "Färisten Administrator's Manual". There is a separate user guide only intended for those users who have been given a TLS certificate. Since the TOE is not aware of any other user, no other user guide exist.

This assurance measure meets the requirements of AGD_ADM.1, AGD_USR.1 and AVA_MSU.2.

6.2.5 AM.LFC – Life cycle support

Development security documentation can be found in the document "Färisten Development Environment". It documents the security aspects in the development environment along with the development processes for the life-cycle definition/model, the tracking and processes for flaw remediation and the documentation of the development tools.

This assurance measure meets the requirements ALC_DVS.1, ALC_LCD.1, ALC_TAT.1, ALC_FLR.1.

6.2.6 AM.TEST – Tests

An analysis of the test coverage and depth of testing is provided together with the test documentation in the document "Färisten Firewall Functional Testing", including a documentation of the performed vulnerability analysis.

The TOE and an equivalent set of resources is provided to the evaluation facility in a manner suitable to testing.

These assurance measures meet the requirements ATE_COV.2, ATE_DPT.1, ATE_FUN.1 and ATE_IND.2.

6.2.7 AM.VULN – Vulnerability assessment

The vulnerability analysis done by the developer is documented in the "Färisten Firewall Vulnerability Analysis". This document also contains the misuse analysis of the guidance documentation.

The TOE includes one mechanism having a strength of TOE security function claim. For this mechanism, a strength of TOE security function identification and justification is provided in chapter 8.2.6 of this Security Target. The strength of function analysis is documented as part of the "Färisten Firewall Vulnerability Analysis".

These assurance measures fulfill the requirements AVA_MSU.2, AVA_SOF.1 and AVA_VLA.2.

7 PP Claims

This ST does not claim conformance with any existing protection profile.

8 Rationale

The rationale section demonstrates how the security objectives of the TOE are met and how objectives, threats and security functions relate to each other. The rationale section will identify, which security functions contribute to which objectives and identify which threats are countered by the individual security functions.

8.1 Security Objectives Rationale

8.1.1 Security Objective Coverage

The following tables provide a mapping of security objectives to the environment defined by the threats, policies and assumptions, illustrating that each security objective covers at least one threat and that each threat is countered by at least one objective, assumption or policy.

	T.ASPOOF	T.DISCLOSE	T.INISEC	T.MEDIAT	T.MODIFY	T.NOAUTH	T.REMOTE	T.SELPRO	T.TIME	T.USEACC	TE.AUDIT	TE.FILE	TE.INFO	A.AUTKEY	A.GENPUR	A.MD5KEY	A.NOEVIL	A.NOEMA	A.PHYSEC	A.RELHARD	A.SINGE	A.SINGI	A.PEERTRUST	P.ADMACC
O.ATISRC									X															
O.AUDIT										X														X
O.DISCLOSE		X																						
O.IDAUTH						X																		
O.LIMEXT						X																		
O.MEDIAT	X			X																				
O.MODIFY					X																			
O.REMOTE							X																	
O.SECSTA			X																					
O.SELPRO								X																
OE.NOEVIL																	X							
OE.AUDIT											X													
OE.AUTKEY														X										
OE.FILESEC												X												
OE.GENPUR															X									
OE.MD5KEY																X								
OE.NOEMA																		X						
OE.PHYSEC						X													X					
OE.RELHARD																				X				
OE.RESID													X											
OE.SINGE																					X			
OE.SINGI																						X		
OE.TIME																								X
OE.PEERTRUST																							X	

Table 8-1: Objectives Related to Assumptions, Threats and Policies

8.1.2 Security Objectives Sufficiency

The auditing of administrator actions as in O.AUDIT, assisted by correct time delivery in OE.TIME, satisfies the organisational security policy of administrators being accountable for their actions as in **P.ADMACC**.

By demanding that the TOE must mediate (i.e. examine) every information sent between different networks connected to the TOE as in O.MEDIAT, the threat of address spoofing as in **T.ASPOOF** can be removed as well as the threat of impermissible information sent through the TOE as in **T.MEDIAT** can be diminished to an acceptable level.

By requiring the TOE to be able to provide a protected channel against disclosure of information as in O.DISCLOSE, and against modification as in O.MODIFY, the threats of **T.DISCLOSE** and **T.MODIFY** are respectively being met.

By requiring well-defined default setting in O.SECSTA, an initial insecure configuration of the TOE is prevented and the threat **T.INISEC** of insecure configuration due to lack of administrator settings is removed.

The threat of a non-administrator gaining access to administration information or modifies administrator actions performing remote administration of the TOE as in **T.REMOTE** is addressed by O.REMOTE requiring a secure remote communication channel between the remote administrator and the TOE.

The threat of a non-administrator performing remote administration of the TOE as in **T.NOAUTH** is sufficiently diminished by requiring a remote communication channel to be established only with authorised users for the administration or the usage of network services, or with authorized peers in the failover configuration in O.IDAUTH and further restricting administration as in O.LIMEXT. **T.NOAUTH** is only relevant for remote administration, because no unauthorized personal can physically access the TOE as demanded by OE.PHYSEC. OE.PHYSEC fulfils A.PHYSEC.

By protecting itself against bypass, deactivation and tampering as in O.SELPRO, the threat **T.SELPRO** is diminished to an acceptable level.

The threat of NTP time server spoofing as in **T.TIME** is removed by enabling the TOE to authenticate NTP time packets in O.ATISRC.

The threat of undetected misuse of the TOE as in **T.USEACC** is extremely diminished by demanding accounting for all security relevant events in O.AUDIT. The threat **TE.AUDIT** that an administrator on the console cannot inspect this accounting evidence is removed by demanding the possibility to view this log files in OE.AUDIT.

By demanding a high quality for the cryptographic keys involved in setting up an authorised TLS for the remote administrator and the VPN connections and protecting the keys from disclosure, OE.AUTKEY fulfils the assumptions **A.AUTKEY**.

Protection of files in the TOE environment against undetected unauthorised modification as in OE. FILESEC diminishes the threat **TE.FILE** to an acceptable level.

Demanding from the TOE environment to prevent disclosure residual information from former information flows in OE.RESID, the threat **TE.INFO** is removed.

The objective to prevent general purpose computing capabilities OE.GENPUR establishes consistency with **A.GENPUR**.

The assumption on high quality MD5 keys for the NTP packet hash sum verification **A.MD5KEY** is fulfilled by the demands of OE.MD5KEY.

The assumption of A.NOEVIL that administrators are non-hostile and trained is supported by OE.NOEVIL.

The environment is consistent with **A.NOEMA** assuming radiation control as provided by OE.NOEMA.

By demanding physical security for the TOE in OE.PHYSEC the environment is

consistent with the assumption of such security in **A.PHYSEC**.

The assumption of correct underlying hardware, firmware and operating system without security critical side effects as in **A.RELHARD** is consistent with OE.RELHARD demanding the absence of such side effects. The correct working of the underlying machine, e.g. related to memory management, program execution, access control and privilege management or identification and authentication, is the basis for the correct working of the TSF.

The assumption that a connection from the TOE to an external network using the VPN functionality only has one connection at the external network as in **A.SINGE** is consistent with the objective for the environment, OE.SINGE which demands that the external networks connection to the TOE using the VPN functionality only has one one connection with the TOE.

The assumption of information that cannot flow among internal and external networks without passing the TOE as in **A.SINGI** is consistent with the objective for the environment, OE.SINGI which demands that the TOE is the only connection between those networks is provided by the TOE.

The assumption that firewall peers in a failover configuration of the TOE are trustworthy and identically configured by the same administrator as in A.PEERTRUST is consistent with OE.PEERTRUST, which stated that exactly that. This means that even if the TOE fails to provide the firewall connectivity it will be provided by a peer firewall. This can only be assumed to be secure if the peer firewall is doing this under the same strong assumptions and with the same configuration as the TOE, as demanded by OE.PEERTRUST.

The security objectives have been derived from the intended use and list of requirements.

8.2 Security Requirements Rationale

8.2.1 Security Requirements Coverage

The following tables provide a mapping of the relationships of security requirements to objectives, illustrating that each security requirement covers at least one objective and that each objective is covered by at least one security requirement.

	O.ATISRC	O.AUDIT	EO.DISCLOS	O.IDAUTH	O.LIMEXT	O.MEDIAT	O.MODIFY	O.REMOTE	O.SECSTA	O.SELPRO
FAU_GEN.1		X								
FAU_SAR.1		X								
FAU_SEL.1		X								
FCO_NRO.1	X									
FCS_CKM.1a								X		
FCS_CKM.1b			X				X			
FCS_CKM.2a				X						
FCS_CKM.2b								X		
FCS_CKM.2c			X				X			
FCS_CKM.2d			X				X			
FCS_COP.1a				X						
FCS_COP.1b								X		
FCS_COP.1c								X		
FCS_COP.1d			X				X			
FCS_COP.1e			X							
FCS_COP.1f							X			
FCS_COP.1g			X							
FCS_COP.1h							X			
FCS_COP.1i	X									
FDP_ACC.2				X				X		
FDP_ACF.1				X				X		
FDP_IFC.1a	X					X				
FDP_IFC.1b						X				
FDP_IFF.1a	X					X				
FDP_IFF.1b						X				
FIA_ATD.1				X						
FIA_UAU.2				X						
FIA_UID.2				X						
FMT_MOF.1					X					
FMT_MSA.1					X					
FMT_MSA.3									X	
FMT_MTD.1					X					
FMT_SMR.1				X						
FMT_SMF.1					X					
FPT_RVM.1										X
FTP_ITC.1			X				X			

Table 8-2: TOE Security Functional Requirements Related to Security Objectives

8.2.2 Justification of security requirements for the IT environment

	OE.AUDIT	OE.AUTKEY	OE.FILESEC	OE.MD5K EY	OE.RESID	OE.TIME
FAU_SAR.1	X					
FCS_CKM.1a		X				
FCS_CKM.1b				X		
FCS_CKM.2		X		X		
FDP_RIP.2					X	
FDP_SDI.1			X			
FPT_STM.1						X

Table 8-3: Security Functional Requirements for the Environment Related to Security Objectives

8.2.3 Functional Security Requirements Sufficiency

8.2.3.1 Security Objectives for the TOE

The security objectives for the TOE are met by the security functional requirements for the TOE in the following way: The objective **O.ATISRC** to authenticate the

integrity of NTP packets is achieved by the requirement of FCO_NRO.1 to generate evidence of origin for NTP packets and is assisted by FCS_COP.1i (MD5) for the cryptographic operation. The objective is also assisted by the UNAUTHENTICATED SFP information flow control (FDP_IFC.1a) of NTP packets and FDP_IFF.1a, blocking packets failing the MD5 verification.

To achieve the provision of evidence for security relevant events and the use of security functions as in **O.AUDIT** the security requirements for the generation of audit data (FAU_GEN.1), for the ability to review the audit (FAU_SAR.1) and for the selection of the events to be audited (FAU_SEL.1) work suitably together.

The objective **O.DISCLOSE** to protect the information on the trusted network (VPN) from being disclose as it is transmitted of the network, is achieved by the FTP_ITC.1 (IPSEC) supported by requirements for key management FCS_CKM.1b (SKUT), key distribution FCS_CKM.2c (SKUT cert) and FCS_CKM.2d (SKUT keys), the cryptographic operations FCS_COP.1d (SKUT RSA), FCS_COP.1e (SKUT 3DES) for the key management and FCS_COP.1g (IPSEC 3DES) for the encryption and decryption of the information received through the VPN tunnel.

The identification and authentication of remote administrators, users and peer firewalls as in **O.IDAUTH** is being met by FIA_ATD.1, FIA_UAU.2, FIA_UID.2, restricted to administrators by the FMT_SMR.1 or authentication failover peers requesting or obtaining configurations, and supported by cryptography for key management FCS_CKM.2a (admin cert) for X.509 certificates and FCS_COP.1a (admin RSA) for the RSA operation. The combination of certificate and certificate owner information leads to the authorisation of user connections, remote administration and peer firewall communication that is met by FDP_ACC.2 and FDP_ACF.1.

The remote administrator by **O.LIMEXT** to provide and limit the control of the TOE security functions behaviour to an authorised administrator by FMT_MOF.1 and FMT_MSA.1 using the management functions described in FMT_SMF.1. This management of TSF data is restricted as by FMT_MTD.1.

The mediation of information flowing between different networks connected to the TOE by the TOE as in **O.MEDIAT** is achieved by the establishment of requirements to enforce at least the UNAUTHENTICATED SFP or AUTHENTICATED SFP for all traffic sent through the TOE (FDP_IFC.1a and FDP_IFC.1b), based on the security attributes defined in FDP_IFF.1a and FDP_IFF.1b.

The trusted channel (VPN) established between the TOE and a Färist must be able to detect any modifications of incoming information as in **O.MODIFY.** This is first met by FTP_ITC.1 (IPSEC) supported by requirements for key management FCS_CKM.1b (SKUT), key distribution FCS_CKM.2c (SKUT cert) and FCS_CKM.2d (SKUT keys), the cryptographic operations FCS_COP.1d (SKUT RSA), FCS_COP.1f (SKUT SHA) for the key management and FCS_COP.1h (IPSEC SHA) for the integrity verification of the information received through the VPN tunnel.

The objective **O.REMOTE** is to provide a secure remote communication channel for remote administration of the TOE by authorised administrators, users going through a TLS plug-proxy, or the TOE submitting the configuration to a failover peer, as well as the TOE requesting a configuration from a peer. It is achieved by the requirements FCS_CKM.1a (admin) for key generation, FCS_CKM.2b (admin keys)

for key distribution, FCS_COP.1b (admin 3DES) for encryption and decryption and FCS_COP.1c (admin SHA) for message digest and verification. The authorisation for performing these remote operations is specified by FDP_ACC.2 and FDP_ACF.1.

The objective **O.SECSTA** to start up and configure the TOE with well-defined initial settings is achieved by the requirement on well-defined default values in FMT_MSA.3.

Self-protection of the TOE against bypass, deactivation and tampering as formulated in **O.SELPRO** is provided by the requirement on TSP enforcement functions in FPT_RVM.1.

8.2.3.2 Security Objectives for the TOE environment

The security objectives for the TOE environment are met by the security functional requirements for the TOE environment in the following way:

OE.AUDIT requires a possibility for the administrator logged in at the console to inspect the audit files, which is fulfilled by FAU_SAR.1.

The objective **OE.AUTKEY** to have high quality cryptographic keys for the remote administrator and peer firewall authentication, and the VPN connection, as well as secure transportation of the private keys not to disclose them, is met by the security requirements on the generation of those keys, i.e. for the generation of TLS key pairs (and certificates) FCS_CKM.1a (admin and VPN cert), for the distribution of these keys FCS_CKM.2. It is supported by OE.PHYSEC and OE.NOEVIL, which ensure that import of these keys, which can only occur during maintenance mode (single user mode) at the system console, can only occur through authorised administrators and cannot be interfered with by malicious third parties.

Protection of configuration and other data stored in files against undetected modification as in **OE.FILESEC** is achieved by the requirement FDP_SDI.1 to monitor such modifications.

OE.MD5KEY aims at the high-quality generation of MD5 keys for the message digest generation and verification of NTP packets with an MD5 hash sum. This if fulfilled by the requirement on key generation FCS_CKM.1b (MD5 key) and key distribution in FCS_CKM.2. It is supported by OE.PHYSEC and OE.NOEVIL, which ensure that import of these keys, which can only occur during maintenance mode (single user mode) at the system console, can only occur through authorised administrators and cannot be interfered with by malicious third parties.

The objective **OE.RESID** to prevent disclosure of residual information is achieved by FDP_RIP.2 demanding the deletion of such information before allocating resources to objects.

The objective **OE.TIME** to provide a reliable time source is achieved by the requirement FPT_STM.1 to provide reliable time stamps.

Although **OE.RELHARD** is an objective for the IT environment, no specific security function requirements for the TOE environment are specified to cover this objective, because **OE.RELHARD** stands for a correct working underlying machine (hardware, firmware, operating system) in general, on which the TOE relies.

8.2.4 Security requirements dependency analysis

Only the functional requirements have been analysed for dependencies since all the assurance requirements in an predefined assurance class already have all the

dependencies resolved. In the table below "(env)" is indicating SFRs in the TOE environment.

Security Requirement	Dependencies/comment	Resolved
FAU_GEN.1	FPT_STM.1 Reliable time stamps	Yes (env)
FAU_SAR.1	FAU_GEN.1 Audit data generation	Yes
FAU_SAR.1 (env)	FAU_GEN.1 Audit data generation	Yes
FAU_SEL.1	FAU_GEN.1 Audit data generation FMT_MTD.1 Management of TSF data	Yes Yes (env)
FCO_NRO.1	FIA_UID.1 Timing of identification	Yes (env)
FCS_CKM.1a (admin)	[**FCS_CKM.2 Cryptographic key distribution** or **FCS_COP.1 Cryptographic operation**] FCS_CKM.4 Cryptographic key destruction FMT_MSA.2 Secure security attributes	Yes Yes No No
FCS_CKM.1b (SKUT)	[**FCS_CKM.2 Cryptographic key distribution** or **FCS_COP.1 Cryptographic operation**] FCS_CKM.4 Cryptographic key destruction FMT_MSA.2 Secure security attributes	Yes Yes No No
FCS_CKM.2a (admin cert)	[FDP_ITC.1 Import of user data without security attributes or FDP_ITC.2 Import of user data with security attributes or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction FMT_MSA.2 Secure security attributes	No, but covered by A.PHYSEC and A.NOEVIL -- -- No No
FCS_CKM.2b (admin keys)	[FDP_ITC.1 Import of user data without security attributes or FDP_ITC.2 Import of user data with security attributes or **FCS_CKM.1 Cryptographic key generation**] FCS_CKM.4 Cryptographic key destruction FMT_MSA.2 Secure security attributes	-- -- Yes No No
FCS_CKM.2c (SKUT cert)	[**FDP_ITC.1 Import of user data without security attributes** or FDP_ITC.2 Import of user data with security attributes or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction FMT_MSA.2 Secure security attributes	No, but covered by A.PHYSEC and A.NOEVIL -- -- No No
FCS_CKM.2d (SKUT keys)	[FDP_ITC.1 Import of user data without security attributes or FDP_ITC.2 Import of user data with security attributes or **FCS_CKM.1 Cryptographic key generation**] FCS_CKM.4 Cryptographic key destruction FMT_MSA.2 Secure security attributes	No, but covered by A.PHYSEC and A.NOEVIL -- Yes No No
FCS_COP.1a (admin RSA)	[FDP_ITC.1 Import of user data without security attributes or FDP_ITC.2 Import of user data with security attributes or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction FMT_MSA.2 Secure security attributes	No, but covered by A.PHYSEC and A.NOEVIL -- -- No No

Security Requirement	Dependencies/comment	Resolved
FCS_COP.1b (admin 3DES)	[FDP_ITC.1 Import of user data without security attributes or FDP_ITC.2 Import of user data with security attributes or **FCS_CKM.1 Cryptographic key generation**] FCS_CKM.4 Cryptographic key destruction FMT_MSA.2 Secure security attributes	-- -- Yes No No
FCS_COP.1c (admin SHA)	[FDP_ITC.1 Import of user data without security attributes or FDP_ITC.2 Import of user data with security attributes or **FCS_CKM.1 Cryptographic key generation**] FCS_CKM.4 Cryptographic key destruction FMT_MSA.2 Secure security attributes	-- -- Yes No No

Security Requirement	Dependencies/comment	Resolved
FCS_COP.1d (SKUT RSA)	[FDP_ITC.1 Import of user data without security attributes or FDP_ITC.2 Import of user data with security attributes or **FCS_CKM.1 Cryptographic key generation**] FCS_CKM.4 Cryptographic key destruction FMT_MSA.2 Secure security attributes	-- -- Yes No No
FCS_COP.1e (SKUT 3DES)	[FDP_ITC.1 Import of user data without security attributes or FDP_ITC.2 Import of user data with security attributes or **FCS_CKM.1 Cryptographic key generation**] FCS_CKM.4 Cryptographic key destruction FMT_MSA.2 Secure security attributes	-- -- Yes No No
FCS_COP.1f (SKUT SHA)	[FDP_ITC.1 Import of user data without security attributes or FDP_ITC.2 Import of user data with security attributes or **FCS_CKM.1 Cryptographic key generation**] FCS_CKM.4 Cryptographic key destruction FMT_MSA.2 Secure security attributes	-- -- Yes No No
FCS_COP.1g (IPSEC 3DES)	[FDP_ITC.1 Import of user data without security attributes or FDP_ITC.2 Import of user data with security attributes or **FCS_CKM.1 Cryptographic key generation**] FCS_CKM.4 Cryptographic key destruction FMT_MSA.2 Secure security attributes	-- -- Yes No No
FCS_COP.1h (IPSEC SHA)	[FDP_ITC.1 Import of user data without security attributes or FDP_ITC.2 Import of user data with security attributes or **FCS_CKM.1 Cryptographic key generation**] FCS_CKM.4 Cryptographic key destruction FMT_MSA.2 Secure security attributes	-- -- Yes No No
FCS_COP.1i (MD5)	[FDP_ITC.1 Import of user data without security attributes or FDP_ITC.2 Import of user data with security attributes or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction FMT_MSA.2 Secure security attributes	No, but covered by A.PHYSEC and A.NOEVIL -- -- No No
FDP_ACC.2	FDP_ACF.1 Security attribute based control	Yes (FDP_ACF.1)
FDP_ACF.1	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	Yes (FDP_ACC.2) Yes (FMT_MSA.3)
FDP_IFC.1a	FDP_IFF.1 Simple security attributes.	Yes (FDP_IFF.1a)

Security Requirement	Dependencies/comment	Resolved
FDP_IFC.1b	FDP_IFF.1 Simple security attributes.	Yes (FDP_IFF.1b)
FDP_IFF.1a	FDP_IFC.1 Subset information flow control FMT_MSA.3 Static attribute initialisation	Yes (FDP_IFC.1a) Yes (FMT_MSA.1)
FDP_IFF.1b	FDP_IFC.1 Subset information flow control FMT_MSA.3 Static attribute initialisation	Yes (FDP_IFC.1b) Yes (FMT_MSA.1)
FIA_ATD.1	No dependencies	--
FIA_UAU.2	FIA_UID.1 Timing of identification	Yes (FIA_UID.2)
FIA_UID.2	No dependencies	--
FMT_MOF.1	FMT_SMF.1 Specification of management functions FMT_SMR.1 Security roles	Yes Yes
FMT_MSA.1	[**FDP_ACC.1 Access control policy** or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	Yes (FDP_ACC.2) -- Yes Yes
FMT_MSA.3	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	Yes Yes
FMT_MTD.1	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	Yes Yes
FMT_SMF.1	No dependencies	--
FMT_SMR.1	FIA_UID.1 Timing of identification	Yes (FIA_UID.2)
FPT_RVM.1	No dependencies	--
FTP_ITC.1	No dependencies	--

Table 8-4: TOE SFR Dependency Analysis

Security Requirement	Dependencies/comment	Resolved
FAU_SAR.1	FAU_GEN.1 Audit data generation	Yes
FCS_CKM.1a (admin and VPN cert)	[**FCS_CKM.2 Cryptographic key distribution** or **FCS_COP.1 Cryptographic operation**] FCS_CKM.4 Cryptographic key destruction FMT_MSA.2 Secure security attributes	Yes Yes No No
FCS_CKM.1b (MD5 key)	[FCS_CKM.2 Cryptographic key distribution or **FCS_COP.1 Cryptographic operation**] FCS_CKM.4 Cryptographic key destruction FMT_MSA.2 Secure security attributes	-- Yes No No
FCS_CKM.2	[FDP_ITC.1 Import of user data without security attributes or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction FMT_MSA.2 Secure security attributes	No, but covered by A.PHYSEC and A.NOEVIL -- No No
FDP_RIP.2	No dependencies	--
FDP_SDI.1	No dependencies	--
FPT_STM.1	No dependencies	--

Table 8-5: TOE environment SFR dependency analysis

8.2.5 Unresolved Dependencies

The unresolved dependency of key generation FCS_CKM.1a (admin) and FCS_CKM.1b (SKUT), and FCS_CKM.2b (admin keys) and FCS_CKM.2d (SKUT keys) on FCS_CKM.4 and FMT_MSA.2 is limited to TLS session keys (3DES, AES

and SHA) for remote administration, peer configuration and SKUT key management. The FCS_CKM.4 requirement for the destruction of session keys are covered by object reuse FDP_RIP.2, no other key destruction method is necessary since the keys are only temporary and the system hosting the TOE is not publicly accessible. The dependency on FMT_MSA.2 is that only secure values of the session keys should be accepted, i.e. no weak keys. This requirement is implicitly fulfilled by the respective key generation requirement FCS_CKM.1a and FCS_CKM.1b.

The unresolved dependencies of FCS_CKM.2a (admin cert), FCS_CKM.2c (SKUT cert) and FCS_COP.1i (MD5) on FCS_CKM.4 and FMT_MSA.2 are limited to TLS public/private key pairs and to MD5 keys. The FCS_CKM.4 requirement for the destruction of those keys is not applicable, since the intention is to keep them on the underlying system, which is not publicly accessible. Secure values as demanded by FMT_MSA.2, i.e. the prevention of weak keys being used, is implicated by the assumption on A.AUTHKEY and A.MD5KEY key generation.

The unresolved dependency of FCS_COP.1a (admin RSA), FCS_COP.1b (admin 3DES), FCS_COP.1c (admin SHA), FCS_COP.1d (SKUT RSA), FCS_COP.1e (SKUT 3DES), FCS_COP.1f (SKUT SHA), FCS_COP.1g (IPSEC 3DES), FCS_COP.1h (IPSEC SHA) on FCS_CKM.4 and FMT_MSA.2 has two different cases that are handled differently:

- dynamic TLS SHA and 3DES session keys
- public key and private RSA keys

The session keys are covered by object reuse FDP_RIP.2, no other key destruction method is necessary. The dependency on FMT_MSA.2 is that only secure values of the session keys should be accepted, i.e. no weak keys. This requirement is fulfilled as part of the protocol by the key generation and key import requirement (TLS session). The RSA keys and certificates are not destroyed as demanded by FCS_CKM.4, because they are to be kept for further authentication processes.

The unresolved dependencies of FDP_ITC.1 for the import of various cryptographic keys (FCS_CKM.2a (admin cert), FCS_CKM.2c (SKUT cert), FCS_CKM.2d (SKUT keys), FCS_COP.1a (admin RSA), FCS_COP.1i (MD5), FCS_CKM.2) are all covered by the assumption A.PHYSEC with support from A.NOEVIL. A.PHYSEC guarantees that physical access to the system will only be granted to authorized personnel, with A.NOEVIL stating that this personnel is benign. All import of cryptographic keys mentioned in these SFRs is done in the system's maintenance mode and can therefore take place from the system console only. With the system in maintenance mode and operated by the system administrator, the IT environment (i.e. the operating system) does not enforce any security policy on the import, but relies on the administrator to correctly perform the import operation.

8.2.6 Strength of Function

A strength of function claim is made for FCS_COP.1i (MD5): the MD5 mechanism used to generate hash sums for the integrity protection of the NTP packets is claimed to be SOF-high. This is in accordance with the related threat T.TIME, which emerges from a high attack potential for the spoofing of NTP packets and the objective O.ATISRC which demands a possibility for integrity checks on NTP packets.

The following argumentation is provided to underline the choice of SOF-high:

MD5 is a publicly known and often analysed algorithm. In 1996, Dobbertin (see [Dobbertin], chapter 1.4) detected weaknesses in using MD5 as a hash function for digital signatures. In 2005, Wang (see [Wang]) identified further attacks when using MD5 as a hash function. But these weaknesses do not effect the use of MD5 with a key to generate message authentication codes. Up to now, no attack is publicly known that could break MD5 with less effort than a brute force attack (see [Robshaw], chapter 1.4). This justifies the SOF-high claim.

Given estimations on the speed of guessing keys (see [Lenstra], chapter 1.4), one can calculate the estimated time to guess a correct MD5 hash.

No strength of function is given for the generation of cryptographic keys, because cryptographic key generation algorithms are implemented in the cryptographic module of TSALIB, a part of FMSSL. TSALIB is not a part of the TOE. FMSSL is a version of the OpenSSL crypto-library (http://www.openssl.org) in which all underlying cryptographic and random algorithms have been substituted by compatible algorithms developed by the Swedish government agency TSA (see [FMSSL] for more details).

8.2.7 Justification of the chosen EAL

The assurance level EAL4 augmented with ALC_FLR.1 has been chosen as appropriate for a Firewall separating an internal network from a public network since it provides a moderate to high level of independently assured security, and a thorough investigation of the TOE. When operated in VPN mode only with no proxy functionality, the TOE has then been approved to separate restricted networks from public networks. When operated in Firewall proxy mode, the TOE may be used to separate internal unclassified networks from public networks. It is assumed that the TOE is operated in an environment where attackers have average expertise of the involved systems (e.g., general and publicly available knowledge on network protocols), limited resources and may have an average motivation because of possible high-value assets protected by the TOE. The overall attack potential is assumed to be low, which means that EAL4 is considered an appropriate assurance level, because it contains AVA_VLA.2 which ensures resistance against attackers with low attack potential.

8.3 TOE summary specification rationale

As stated in the tables above, every objective is addressed by at least one security functional requirement and every SFR is necessitated to cover at least one objective. By showing that the stated security objectives are met, we are able to demonstrate the suitability and sufficiency of the chosen SFRs.

The requirements are mutual supportive, i.e. there exist no conflicts between different requirements, and they are consistent in defining a proper set of demands on the functionality the TOE is supposed to offer.

8.3.1 Security functions and assurance measures coverage

The following tables provide a mapping between security functions and security functional requirements as well as assurance measures and security assurance requirements.

	SF.IPS	SF.PF	SF.PSD	SF.AP	SF.KEYMAN	SF.VPNCH	SF.LCONF	SF.RCONF	SF.RCHIN	SF.RCHOUT
FAU_GEN.1	X	X	X	X	X	X		X	X	X
FAU_SAR.1								X		
FAU_SEL.1		X								
FCO_NRO.1			X							
FCS_CKM.1a									X	X
FCS_CKM.1b					X					
FCS_CKM.2a									X	X
FCS_CKM.2b									X	X
FCS_CKM.2c					X					
FCS_CKM.2d					X					
FCS_COP.1a									X	X
FCS_COP.1b									X	X
FCS_COP.1c									X	X
FCS_COP.1d					X					
FCS_COP.1e					X					
FCS_COP.1f					X					
FCS_COP.1g						X				
FCS_COP.1h						X				
FCS_COP.1i			X							
FDP_ACC.2									X	
FDP_ACF.1									X	
FDP_IFC.1a	X	X		X						
FDP_IFC.1b	X					X				
FDP_IFF.1a	X	X		X						
FDP_IFF.1b	X					X				
FIA_ATD.1									X	X
FIA_UAU.2									X	X
FIA_UID.2									X	X
FMT_MOF.1							X	X		
FMT_MSA.1							X	X		
FMT_MSA.3								X		
FMT_MTD.1							X	X		
FMT_SMR.1									X	X
FMT_SMF.1							X	X		
FPT_RVM.1	X									
FTP_ITC.1						X				

Table 8-6: Security Functions meeting SFRs and Vice Versa

Assurance measure	SAR
AM.CONFIG	ACM_AUT.1, ACM_CAP.4, ACM_SCP.2
AM.DEL	ADO_DEL.2, ADO_IGS.1, AVA_MSU.2
AM.DEVEL	ADV_FSP.2, ADV_HLD.2, ADV_IMP.1, ADV_LLD.1, ADV_RCR.1, ADV_SPM.1
AM.GUIDE	AGD_ADM.1, AGD_USR.1, AVA_MSU.2
AM.LFC	ALC_DVS.1, ALC_LCD.1, ALC_TAT.1, ALC_FLR.1
AM.TEST	ATE_COV.2, ATE_DPT.1, ATE_FUN.1, ATE_IND.2
AM.VULN	AVA_VLA.2, AVA_SOF.1, AVA_MSU.2

Table 8-7: Assurance measures meeting SARs

SAR	Assurance measure
ACM_AUT.1	AM.CONFIG
ACM_CAP.4	AM.CONFIG
ACM_SCP.2	AM.CONFIG
ADO_DEL.2	AM.DEL
ADO_IGS.1	AM.DEL
ADV_FSP.2	AM.DEVEL
ADV_HLD.2	AM.DEVEL
ADV_IMP.1	AM.DEVEL
ADV_LLD.1	AM.DEVEL
ADV_RCR.1	AM.DEVEL
ADV_SPM.1	AM.DEVEL
AGD_ADM.1	AM.GUIDE
AGD_USR.1	AM.GUIDE
ALC_DVS.1	AM.LFC
ALC_FLR.1	AM.LFC
ALC_LCD.1	AM.LFC
ALC_TAT.1	AM.LFC
ATE_COV.2	AM.TEST
ATE_DPT.1	AM.TEST
ATE_FUN.1	AM.TEST
ATE_IND.2	AM.TEST
AVA_MSU.2	AM.DEL, AM.GUIDE, AM.VULN
AVA_SOF.1	AM.VULN
AVA_VLA.2	AM.VULN

Table 8-8: SARs met by assurance measures

8.3.2 Security functions sufficiency

FAU_GEN.1 requires the generation of an audit record and lists the types of events to be recorded. This requirement is met by all of the individual security functions, which implement the generation of audit data in their functionality. In detail, the generic log information is provided by all security functions, including the IP stack (SF.IPS) and the packet filter (SF.PF). FTP, SMTP, DNS and plug proxy specific audit data is generated by the application level proxies of SF.AP, packet screening daemon specific audit data is generated from the packet screening daemon in SF.PSD, key management audit data from the SKUT protocol will be generated by SF.KEYMAN and the VPN channel (SF.VPNCH), data and administration specific data is generated from the remote (SF.RCONF) configuration tools, and audit data is generated when establishing the incoming (SF.RCHIN) and outgoing (SF.RCHOUT) connections for synchronisation and remote administration. SF.LCONF does not contain audit functions, which does not pose any vulnerability, because SF.LCONF describes local configuration that is only performed by the local administrator. This super user is assumed to be trustworthy and no audit data for his administrative actions are required.

The ability for administrators to read the audit records, as required in **FAU_SAR.1**, is provided by the remote administration tool (SF.RCONF), which offers an appropriate menu entry.

The packet filter SF.PF as part of the IP stack is able to include or exclude events for being audited as required in **FAU_SEL.1**.

The requirements **FCO_NRO.1** to generate evidence of origin for transmitted NTP packets with help of MD5 hash sums verified according to **FCS_COP.1i (MD5)** is met by the packet screening daemon SF.PSD, which is able to check the validity of MD5 hash sums of NTP packets.

The enforcement of the AUTHENTICATED USER ACCESS SFP on all operations between remote TLS clients and TSF data/internal network resources is specified in **FDP_ACC.2** and **FDP_ACF.1**. This enables the mutual authentication of the TLS client and the TOE, using certificates, and enables a secure remote communication channel (SF.RCHIN). SF.RCHIN also enables users with a valid certificate to go through a TLS plug-proxy.

The enforcement of the UNAUTHENTICATED SFP and AUTHENTICATED SFP on all data coming from external entities and sent through the TOE as in **FDP_IFC.1a** and **FDP_IFC.1b** respectively based on the types of subjects, information security attributes and rules defined in **FDP_IFF.1a** and/or in **FDP_IFF.1b** is met by the IP stack (SF.IPS) handling the data. The information will then either be routed to the the packet filter (SF.PF) applying certain filtering rules and the application proxies (SF.AP) filtering the data on an application specific level or will be routed to the VPN channel (SF.VPNCH). The requirement for the inter-TSF trusted channel **FTP_ITC.1** is also provided by the VPN channel (SF.VPNCH).

The confidentiality and integrity protection of the data exchanged over the VPN as required by **FCS_COP.1g (IPSEC 3DES)** and **FCS_COP.1h (IPSEC SHA)** respectively is provided by the VPN channel (SF.VPNCH).

The requirement on cryptographic functions for the VPN key management, **FCS_CKM.1b**, **FCS_CKM.2c**, **FCS_CKM.2d**, **FCS_COP.1d**, **FCS_COP.1e** and **FCS_COP.1f** are met by the TLS protocol implemented by SKUT in the SF.KEYMAN.

The requirements on cryptographic function for the remote administration **FCS_CKM.1a**, **FCS_CKM.2a**, **FCS_CKM.2b**, **FCS_COP.1a**, **FCS_COP.1b** and **FCS_COP.1c** are implemented by the TLS protocol implemented by SF.RCHIN and SF.RCHOUT, depending if there is an incoming or outgoing connection request..

The requirements for identification and authentication **FIA_ATD.1, FIA_UID.2, FIA_UAU.2** and **FMT_SMR.1** for remote administrators are implemented using the SF.RCHIN and SF.RCHOUT, using the TLS protocol. The authentication is mutual, which means that there are no differences if there is an incoming or outgoing request.

The requirements for management functions **FMT_MOF.1**, **FMT_MSA.1**, **FMT_MTD.1** and **FMT_SMF.1** are all implemented by SF.LCONF and SF.RCONF, where SF.RCONF provides the remote configuration tools and SF.LCONF provides the local configuration tools. SF.LCONF provides a subset of the management functions of SF.RCONF. Furthermore, SF.LCONF does not fully satisfy the FMT_SMR.1 as required by the management security requirement dependencies, but relies on A.PHYSEC to provide physical access to the system console (from where all SF.LCONF actions are performed) only to authorized personnel.

The requirement for static attribute initialisation **FMT_MSA.3** is implemented by the remote administration function SF.RCONF.

Well-defined default settings for security attributes, as they are required in **FMT_MSA.3**, are provided by the remote administration in a way that no communication between the separated networks is allowed which has not been explicitly configured by an administrator.

The enforcement of the TSF functions before allowing any other functions as in **FPT_RVM.1** is performed by the IP stack (SF.IPS), which controls the information flow of every in- and outgoing data.

The IT security functions provided by the TOE work together to satisfy the TOE security functional requirements defined in this Security Target. The tight relationship between the defined requirements and the fulfilment of these requirements by security functions, as illustrated above, provides no room for the introduction of potential security weaknesses not identified in this document.

The TOE security functions work together as follows: SF.IPS takes care of the central distribution of the user data (IP packets sent from one to the other network through the TOE). It is directly supported by integration of SF.PF, for the examination of each packet on IP level, and SF.PSD, for checks related to the higher level protocols DNS and NTP. SF.IPS then distributes the user data to SF.AP and receives from SF.AP newly generated IP packets for controlled dissemination to the network interfaces. Any data to be routed through a VPN connection will be encrypted and decrypted by the SF.VPNCH. This channel is established and keys are managed using the VPN key management function SF.KEYMAN, that will ensure that new keys are renegotiated on a regular basis.

SF.RCONF and SF.LCONF enable the TOE administrator to alter the TOE configuration and therefore to influence the configuration of all other security functions – SF.LCONF furthermore supports SF.RCONF by generating single configuration files out of a central configuration file. For remote communication the connecting an administrator will be identified and authenticated using SF.RCHIN. The same function is used for synchronisation of configuration files in a failover configuration. The peer Färist will identify and authenticate itself to another Färist using SF.RCHOUT with the SF.RCHIN of the other Färist.

8.3.2.1 Assurance measures efficiency

ACM_CAP.4 requires TOE reference labelling and a configuration management documentation describing the configuration management system used by the developer. It further requires the clear identification of the configuration items and that the ability of modifying these items is properly controlled. This also requires acceptance procedures to confirm that any creation and modification of configuration items is authorised. Such evidence is delivered by AM.CONFIG, mainly referencing to the required documentation. As is the evidence for the requirements in **ACM_SCP.2** for the scope of CM tracking and a description how the tracking is done. **ACM_AUT.1** requires the use of automated CM tools and is also covered by AM.CONFIG.

The requirements of **ADO_DEL.2** documenting and using the delivery procedures for the TOE to the user and **ADO_IGS.1** documenting the procedures necessary for a secure installation, generation an start-up of the TOE are met by AM.DEL referring to the documentation "Färisten Delivery Procedures".

The development documentation is provided by AM.DEVEL, referring to the documents containing the functional specification as required in **ADV_FSP.2**, the security enforcing high-level design as in **ADV_HLD.2**, the low-level design as in **ADV_LLD.1**, the implementation representation as in **ADV_IMP.1**, the informal correspondence analysis as required in **ADV_RCR.1** and the security policy model as required in **ADV_SPM.1.**

The guidance documentation is provided by AM.GUIDE, referring to the document providing the administrator guidance required by **AGD_ADM.1** and the user guidance as required in **AGD_USR.1**.

ALC_DVS.1 requires a development security documentation, **ALC_FLR.1** requires the development of flaw remediation procedures, **ALC_LCD.1** requires the development of a TOE life-cycle documentation, and **ALC_TAT.1** requires the developer to use well-defined development tools. Each of these assurance requirements are met by AM.LFC that refers to the appropriate document.

AM.TEST provides the reference to the test documentation, which includes an analysis of the test coverage required by **ATE_COV.2** and of the test depth as required by **ATE_DPT.1**, as well as the test documentation required by **ATE_FUN.1**. Furthermore, the TOE is said to be provided for testing by the developer as required in **ATE_IND.2**.

AVA_MSU.2 requires the developer to provide guidance documentation and a misuse analysis of the guidance documentation. These requirements are met by AM.GUIDE and AM.DEL stating that guidance documentation (including steps for the secure installation, generation and start-up of the TOE) is provided and AM.VULN, saying that the vulnerability analysis, provided by the developer, contains such a misuse analysis. Additionally AM.VULN identifies the mechanisms with a strength of function claim and claims that the requirement of AVA_SOF.1 for a strength of function analyses for the identified mechanisms is fulfilled. The strength of function is justified as part of section 8.2.6 in this document. The requirement of **AVA_VLA.2** to provide a vulnerability analysis and the corresponding documentation resp. is fulfilled by AM.VULN, which states that the documentation provided by the developer includes the vulnerability analysis.

8.3.3 Strength of Function

For the security function SF.PSD, SOF-high is claimed for the mechanism implementing the verification of MD5 hash sums. This is done in accordance with the strength of function claim for the corresponding security functional requirement FCS_COP.1i (MD5).

No claims are made about the strength of function for any cryptographic functions. This is covered by the cryptographic verification performed by the government agency TSA, as described in section 8.2.6.

8.4 PP Claims Rationale

This ST does not claim conformance with any existing protection profile.

9 Appendix

A.1 Abbreviations

AES	Advanced Encryption Standard
CC	Common Criteria
DES	Data Encryption Standard
EAL	Evaluation Assurance Level
FTP	File Transfer Protocol
HTTP	Hyper Text Transfer Protocol
ICMP	Internet Control Message Protocol
IP	Internet Protocol
IT	Information Technology
NTP	Network Time Protocol
OS	Operating System
PP	Protection Profile
RFC	Requests for Comments
SF	Security Function
SFP	Security Function Policy
SMTP	Simple Mail Transfer Protocol
SOF	Strength of Function
ST	Security Target
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TOE	Target of Evaluation
TSC	TSF Scope of Control
TSF	TOE Security Functions
TSFI	TSF Interface
TSP	TOE Security Policy
UDP	User Datagram Protocol
VPN	Virtual Private Network
WWW	Word Wide Web

A.2 Glossary

Assets	Information or resources to be protected by the countermeasures of a TOE.

Assignment	The specification of an identified parameter in a component.
Assurance	Grounds for confidence that an entity meets its security objectives.
Attack potential	The perceived potential for success of an attack, should an attack be launched, expressed in terms of an attacker's expertise, resources and motivation.
Augmentation	The addition of one or more assurance component(s) from Part3 to an EAL or assurance package.
Authentication data	Information used to verify the claimed identity of a user.
Authorised user	A user who may, in accordance with the TSP, perform an operation.
Authorised administrator	A human user to whom the authorisation has been granted to perform administrative operations which may affect the enforcement of the TSP.
Class	A grouping of families that share a common focus.
Component	The smallest selectable set of elements that may be included in a PP, an ST, or a package.
Connectivity	The property of the TOE which allows interaction with IT entities external to the TOE. This includes exchange of data by wire or by wireless means, over any distance in any environment or configuration.
Dependency	A relationship between requirements such that the requirement that is depended upon must normally be satisfied for the other requirements to be able to meet their objectives.
Domain Name Service	The on-line distributed database system used to identify host IP addresses as human-readable machine names.
Element	An indivisible security requirement.
Evaluation	Assessment of a PP, an ST or a TOE, against defined criteria.
Evaluation Assurance Level	A package consisting of assurance components from Part 3 that represents a point on the CC predefined assurance scale.
Evaluation authority	A body that implements the CC for a specific community by means of an evaluation scheme and thereby sets the standards and monitors the quality of evaluations conducted by bodies within that community.
Evaluation scheme	The administrative and regulatory framework under which the CC is applied by an evaluation authority within a specific community.
Extension	The addition to an ST or PP of functional requirements not contained in Part2 and/ or assurance requirements not contained in Part 3 of the CC.

External IT entity	Any IT product or system, untrusted or trusted, outside of the TOE that interacts with the TOE.
Family	A grouping of components that share security objectives but may differ in emphasis or rigour.
File Transfer Protocol	An application used to transfer files from one site to another. Users normally use an FTP client program to access an FTP server.
Formal	Expressed in a restricted syntax language with defined semantics based on well-established mathematical concepts.
Human user	Any person who interacts with the TOE.
HypterText Transfer Protocol	The protocol used on the World Wide Web to retrieve pages.
ICMP Ping	A program used with TCP/IP networks to test the reachability of destinations, by sending an ICMP echo request and waiting for the reply. Also known as PING (Packet InterNet Groper).
Identity	A representation (e.g., a string) uniquely identifying an authorised user, which can either be the full or abbreviated name of that user or a pseudonym.
Informal	Expressed in natural language.
Internal communication channel	A communication channel between separated parts of the TOE.
Internal TOE transfer	Communicating data between separated parts of the TOE.
Inter-TSF transfers	Communicating data between the TOE and the security functions of other trusted IT products.
Iteration	The use of a component more than once with varying operations.
Object	An entity within the TSC that contains or receives information and upon which subjects perform operations.
Organisational security policies	One or more security rules, procedures, practices, or guidelines imposed by an organisation upon its operations.
Package	A reusable set of either functional or assurance components (e.g., an EAL), combined together to satisfy a set of identified security objectives.
Peer firewall	A firewall that is configured as a failover firewall for other peer firewalls for a specific network connection. All peers have the same configuration and are under the same administration.
Private network	An internal network segment that is to be protected from external or untrusted network segments.

Product	A package of IT software, firmware and/or hardware, providing functionality designed for use or incorporation within a multiplicity of systems.
Protection Profile	An implementation-independent set of security requirements for a category of TOEs that meet specific consumer needs.
Public network	An external network segment, that is considered to have all, or mostly, untrusted users.
Proxy	A method whereby a process pretends to be the intended recipient host, thereby ensuring secure communication through a gateway.
Reference monitor	The concept of an abstract machine that enforces TOE access control policies.
Reference validation mechanism	An implementation of the reference monitor concept that possesses the following properties: it is tamperproof, always invoked, and simple enough to be subjected to thorough analysis and testing.
Refinement	The addition of details to a component.
Role	A predefined set of rules establishing the allowed interactions between a user and the TOE.
Secret	Information that must be known only to authorised users and/or the TSF in order to enforce a specific SFP.
Security attribute	Information associated with subjects, users and/or objects that is used for the enforcement of the TSP.
Security Function	A part or parts of the TOE that have to be relied upon for enforcing a closely related subset of the rules from the TSP.
Security Function Policy	The security policy enforced by an SF.
Security objective	A statement of intent to counter identified threats and/or satisfy identified organisation security policies and assumptions.
Security Target	A set of security requirements and specifications to be used as the basis for evaluation of an identified TOE.
Selection	The specification of one or more items from a list in a component.
Semiformal	Expressed in a restricted syntax language with defined semantics.
Service	A third layer (within the TCP/IP communications protocol model)
Simple Mail Transfer Protocol	The TCP/IP standard protocol for the electronic transfer of mail messages.

Strength of Function	A qualification of a TOE security function expressing the minimum efforts assumed necessary to defeat its expected security behaviour by directly attacking its underlying security mechanisms.
SOF-basic	A level of the TOE strength of function where analysis shows that the function provides adequate protection against casual breach of TOE security by attackers possessing a low attack potential.
SOF-medium	A level of the TOE strength of function where analysis shows that the function provides adequate protection against straightforward or intentional breach of TOE security by attackers possessing a moderate attack potential.
SOF-high	A level of the TOE strength of function where analysis shows that the function provides adequate protection against deliberately planned or organised breach of TOE security by attackers possessing a high attack potential.
Subject	An entity within the TSC that causes operations to be performed.
System	A specific IT installation, with a particular purpose and operational environment.
Target of Evaluation	An IT product or system and its associated administrator and user guidance documentation that is the subject of an evaluation.
Telnet	A remote terminal application that allows users to access a remote computer. The user normally uses a Telnet client program, and the remote computer must have a Telnet server running (usually telnet on UNIX).
TOE resource	Anything useable or consumable in the TOE.
TOE Security Functions	A set consisting of all hardware, software, and firmware of the TOE that must be relied upon for the correct enforcement of the TSP.
TOE Security Functions Interface	A set of interfaces, whether interactive (man-machine interface) or programmatic (application programming interface), through which TOE resources are accessed, mediated by the TSF, or information is obtained from the TSF.
TOE Security Policy	A set of rules that regulate how assets are managed, protected and distributed within a TOE.
TOE security policy model	A structured representation of the security policy to be enforced by the TOE.
Transfers outside TSF control	Communicating data to entities not under control of the TSF.
Transmission Control Protocol	A connection-oriented protocol that provides reliable virtual circuits, running atop IP

Trusted channel	A means by which a TSF and a remote trusted IT product can communicate with necessary confidence to support the TSP.
Trusted path	A means by which a user and a TSF can communicate with necessary confidence to support the TSP.
TSF data	Data created by and for the TOE, that might affect the operation of the TOE.
TSF Scope of Control	The set of interactions that can occur with or within a TOE and are subject to the rules of the TSP.
User	Any entity (human user or external IT entity) outside the TOE that interacts with the TOE.
User data	Data created by and for the user, that does not affect the operation of the TSF.
User Datagram Protocol	Applications with IP services.
World Wide Web	An application used as an information-browsing tool on the Internet. A WWW client program (browser), such as Netscape Navigator, accesses information stored on servers. WWW clients and servers communicate primarily using the HyperText Transfer Protocol (HTTP); however, they can also communicate with Gopher servers, News servers, and FTP servers.