gateProtect Firewall Packet-Filtering-Core v10.3 Security Target

Version:	**1.0**
Status:	**Release**
Last Update:	**2013-02-08**
Classification:	**public**

Legal Notice

This document is provided AS IS with no express or implied warranties. Use the information in this document at your own risk.

Revision History

Revision	Date	Author(s)	Changes to Previous Revision
1.0	2013-02-08	Andreas Siegert	Public release

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Security Target Identification

Title:	gateProtect Firewall Packet-Filtering-Core v10.3 Security Target
Version:	1.0
Status:	Release
Date:	2013-02-08
Sponsor:	gateProtect AG
Developer:	gateProtect AG
Certification body:	BSI
Certification ID:	BSI-DSZ-CC-0792
Keywords:	Security Target, Common Criteria, firewall, packet filter, network security, information flow control

1.2 TOE Identification

The TOE is gateProtect Firewall Packet-Filtering-Core Version 10.3.

1.3 TOE Type

The TOE type is Packet Filtering Firewall.

1.4 TOE Overview

The TOE (gateProtect Firewall Packet-Filtering-Core v10.3) is the network information flow enforcing software component of the gateProtect Firewall v10.3.

The gateProtect Firewall v10.3 product containing the TOE is a next generation high performance firewall system. It secures businesses from malware attacks, viruses, unauthorized access and abuse. The gateProtect firewall protects networks from very small to large enterprises. It is characterized by optimal scaleability, security and performance.

Based on an innovative technological architecture the gateProtect Firewall v10.3 is provided with a new web-based ergonomic Graphic User Interface – eGUI. gateProtect's eGUI provides an intuitive and effective visual management interface. The process-oriented eGUI provides the following major advantages:

- Huge time-savings through a significant reduction in the number of rules.
- Reduction in the number of user errors due to the visualization of the entire network.
- Reduced operating costs through active management.

Besides these features the firewall product also provides application visibility and control. V10.3 supports 64Bit Systems, auditing and full IPV6 integration. Network features such as Deep Packet Inspection, bridging, VLAN, and VPN crypto acceleration are included in gateProtect Firewall v10.3.

The gateProtect Firewall v10.3 product is shipped as a Linux-based appliance.

The TOE as part of the above described product is limited to the Packet-Filtering-Core and the configuration engine that simplifies the rule specification. Other product features (Deep Packet Inspection,VPN, eGUI) are excluded from the evaluation due to resource constraints, not because of any security relevant restrictions.

The Packet-Filtering-Core implements the Network Information Flow Control Policy in the running system. That policy acts on IP packets based on packet header information. In contrast to classic network access control devices, the TOE uses a high-level description of the Network Information Flow Control Policy to generate the rules for the filter allowing the administrator to focus on the big picture and not the details of how to protect the devices behind the firewall.

Events from the filter and the configuration subsystem generate audit log events that can be used by the administrator to monitor the system.

The administration of the TOE is performed in the TOE environment. The TOE itself just takes a configuration file from the environment and applies the configuration changes to the running system.

1.5 TOE Description

1.5.1 Introduction

The gateProtect Firewall v10.3 product is based on a Packet-Filtering-Core implemented by the TOE, which provides network access control based on a user-supplied rule base used to model the Network Information Flow Control Policy. The TOE is therefore a solution for secure network segregation and network border protection.

The enforcement of the network information flow control policy is handled by the TOE providing a proprietary packet filtering component used by the appliance's underlying Linux operating system. The key difference between a regular IPTables firewall and the gateProtect firewall product is the way the rule base is configured. Instead of having to manually specify many detailed rules for IPTables, the gateProtect Firewall v10.3 product works with high level descriptive rules that model communication relationships. The configuration loading and rule transforming parts, a configuration database and the enforcing kernel components are all part of the TOE together with the audit daemon.

The Linux appliance that hosts the firewall and the TOE contained within can provide other functionality as well, but such functionality is not part of the evaluated configuration as shown in the section about the evaluated configuration below. The additional product functionality does not interfere with the TSFI and therefore may be used in the evaluated configuration.

1.5.2 Architecture

The enforcing components of the system (the Network Information Flow Control Subsystem) are implemented via IPTables. The filtering modules are embedded in the network stack of the appliance's underlying Linux system. Figure 1 shows the structure of the TOE. The physical and logical network interfaces (If) provided by the Linux environment deliver packets to the Network Information Flow Control Subsystem (NIFC) which handles the information flow control decisions based on the configuration of the NIFC and on the packet header information. This happens for all packets arriving at the network interfaces, regardless of whether they are destined locally or are to be routed through. The "Rules" part in figure 1 is implemented via IPTables.

The Network Information Flow Control Subsystem configuration is loaded by a configuration loader (*cltool*) that reads the supplied configuration file and passes it on to the configuration daemon (*stated*) that transforms the user rules into IPTables rules and manages the available configurations in a configuration database. The configuration daemon then uses the iptables-restore script to load and activate the required IPTables rules into the kernel.

Audit information including statistical data about the packet flow is provided by the packet filter. The configuration daemon provides audit information about configuration changes. The audit log daemons (*ulogd*, *rsyslogd*) gather that information and provide the audit log files on disk. A

watchdog daemon (*monit*) monitors the available file space for the audit data and generates alerts via the log file and e-mail should a configured threshold be reached. The configuration for *monit* is provided via *stated*.

The visible interfaces to the TOE are the configuration file, the audit logs and the logical (and therefore also physical) network interfaces.

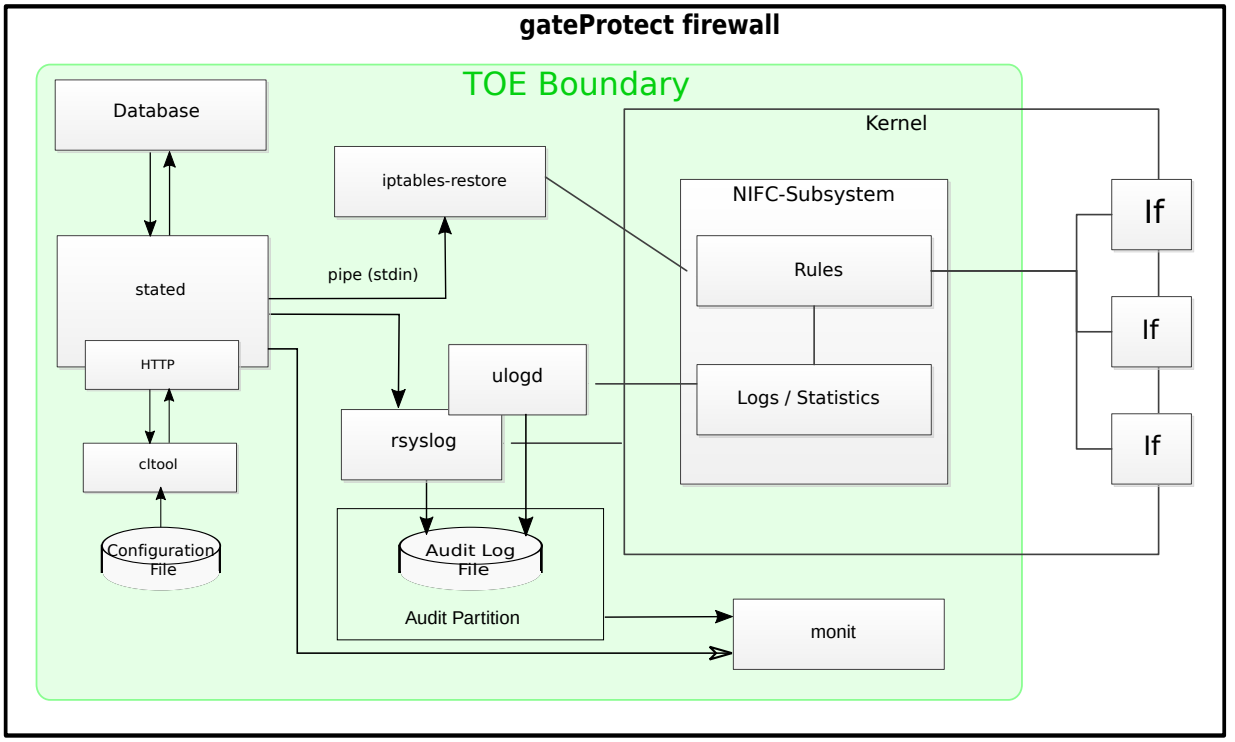

Figure 1: TOE Overview

A typical setup of the firewall is shown in figure 2. The firewall sits between two or more networks. One network is trusted (the internal network) and another one is untrusted (external network). The administrator uses TOE environment functions (secured network access via a dedicated network interface protected by the environment or console access to the underlying operating system) to access the TOE environment to deposit configuration information or fetch audit records.

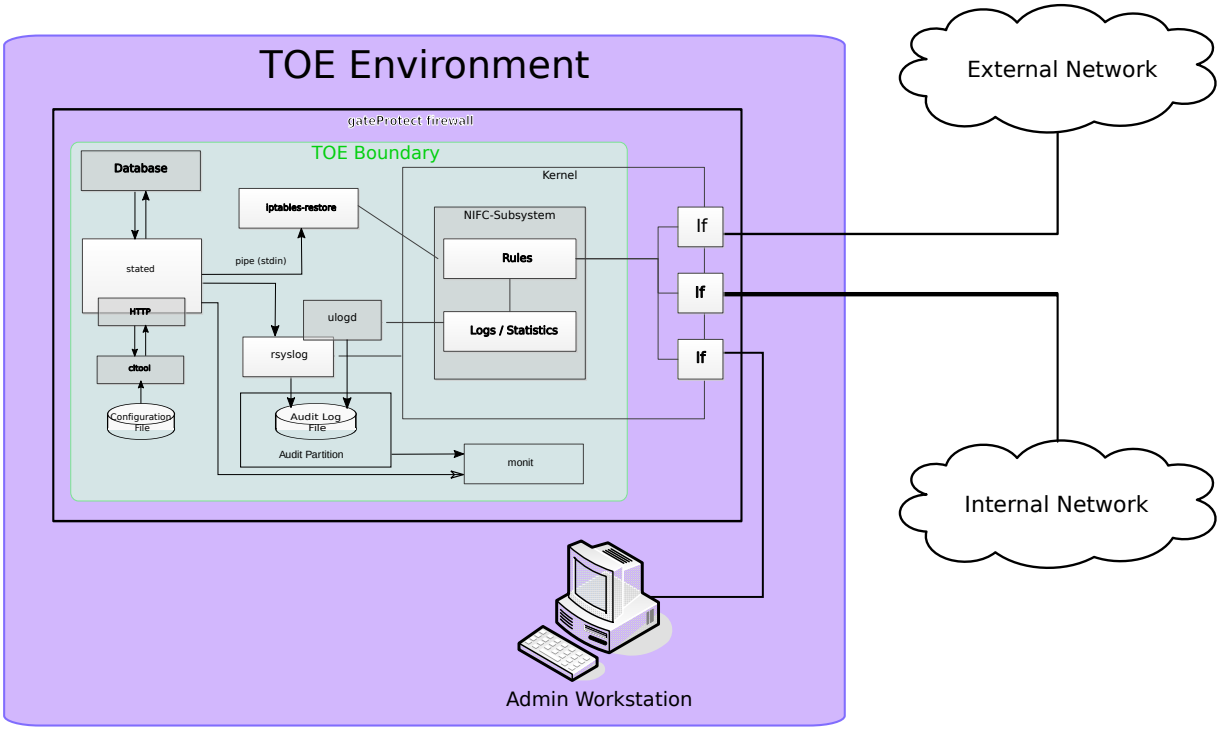

Figure 2: TOE Environment

1.5.3 TOE Scope

1.5.3.1 Physical and Logical

The TOE is software only. It is included in the downloadable ISO install image for the gateProtect firewall appliances that is to be installed by the user. The explicit installation is only needed so that the integrity of the code can be verified.

Relevant guidance documents for the secure operation of the TOE are:

- gateProtect Firewall v10.3 Packet Filtering Core Evaluated Configuration Guide

The following components can be found in the IT environment:

- Debian Linux and its hardware platform.

The logical scope of the TOE consists of

- the packet filter (IPTables) that are integrated into the IP stack of the underlying host system
- the configuration loading mechanism (*cltool*, *stated* with the associated database and iptables-restore script)
- the audit log daemons (*ulogd*, *rsyslogd*) and the monitoring watchdog daemon (*monit*).

1.5.3.2 Evaluated Configuration

The following configuration specifics apply to the evaluated configuration of the TOE:

Hardware

The following appliances are supported platforms for the TOE:

- GPA 250
- GPA 400
- GPX 2500

The operating system used is Debian Linux 6.0

Network Protocols

- IPv4, IPv6
- ICMP, ICMPv6, UDP, TCP, ESP, AH

Bridging and VLAN support

The following features and functions of the gateProtect Firewall appliance may be used but are not part of the evaluated TSF:

- Management eGui to generate the configuration file.
- Deep Packet Inspection
- VPN Support (the packet filter does control the packet flow of VPN packets, but the use of the TOE as a VPN endpoint is not in the scope of the evaluation)

1.5.4 Security Functionality

1.5.4.1 TOE Security Functions

Network Information Flow Control

Packets arriving at the physical or logical network interfaces are subject to the network information flow control policy. They are either passed on or dropped according to the policy.

Audit

All packets handled by the firewall are subject to a statistics gathering module that records connections and provides a log of all connections and connection attempts handled by the firewall. Configuration changes are subject to audit record generation.

Configuration

The network information flow control can be modified by a configuration file in the TOE environment that the TOE uses to configure its runtime behaviour.

The default policy is to drop packets and only by configuring explicit policies packets can be transported to or through the firewall.

The configuration encompasses the behaviour of the Network Information Flow Control and the audit subsystem.

Configuration changes are performed by editing a configuration file in the environment that is read by the configuration daemon.

1.5.4.2 IT-Environment Support

The TOE relies on the Linux environment to provide Identification and Authentication for the administrator as well as data storage (configuration information and audit records) and basic networking support (network interfaces, routing).

The reliable timestamps for the audit function are provided by the environment.

2 CC Conformance Claim

This Security Target is CC Part 2 extended and CC Part 3 conformant, with a claimed Evaluation Assurance Level of EAL4, augmented by ALC_FLR.1.

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

Common Criteria [CC] version 3.1 revision 3 is the basis for this conformance claim.

3 Security Problem Definition

3.1 Threat Environment

This section describes the threat model for the TOE and identifies the individual threats that are assumed to exist in the TOE environment.

The **assets** to be protected by the TOE are

Subjects
The network resources that are protected by the NIFCP.

Configuration Data
Configuration data of the TSF.

Audit Records
Audit records generated in the TOE.

The **threat agents** having an interest to subvert the NIFCP are attackers (unauthorized users or systems) accessing the TOE or TOE protected systems from the networks connected to the TOE. Attackers with an Enhanced-Basic attack potential are assumed.

TOE administrators which are actually administrators of the TOE environment, are assumed to be trustworthy, trained and to follow the instructions provided to them with respect to the secure configuration and operation of the systems under their responsibility. Hence, only inadvertent attempts to manipulate the safe operation of the TOE are expected from this community.

3.1.1 Threats countered 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.INISEC

Network packets may inadvertently be routed through the TOE because the packet is not matched by any explicit rule.

T.MEDIAT

An attacker in the external network may send impermissible information through the TOE, including illegally formed packets, that circumvent the restrictions of the Network Information Flow Control Policy.

3.1.2 Threats countered by the Operational Environment

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.

TE.FILE

An attacker may alter TSF data without being detected.

3.2 Assumptions

3.2.1 Environment of use of the TOE

A.ADMINACC

The IT-environment protects the logical and physical administrative access to the TOE.

A.ADMINPORT

A dedicated network port is used for administrative access to the TOE.

A.NOEVIL

Authorised administrators having access to the TOE environment, are competent, non-hostile and follow all their guidance; however, they are capable of error.

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.SINGIF

Information cannot flow among the internal and external networks unless it passes through the TOE, i.e. the TOE is the only connection point between those two networks.

A.RTS

The IT-environment provides reliable timestamps

3.3 Organizational Security Policies

P.AUDIT

The TOE shall record all of its security relevant actions.

P.CONFIG

The TOE shall support the means to configure the network information flow control policy.

4 Security Objectives

4.1 Objectives for the TOE

O.AUDIT

The TOE must be able to provide audit evidence of security relevant events as well as for the use of security functions.

O.MEDIAT

The TOE must mediate the flow of all information between the TOE's network interfaces.

O.CONFIG

The TOE must provide the means to configure the network information flow control policy.

O.SECSTA

Upon initial start-up of the TOE or during configuration, the TOE shall provide well-defined restrictive initial settings for security relevant functions.

4.2 Objectives for the Operational Environment

OE.ADMINACC

The IT-environment must provide logical and physical protection of the administrative access to the TOE.

OE.ADMINPORT

The administrative network access to the TOE must only use a dedicated port of the appliance and not any of the other available network ports.

OE.NOEVIL

Authorised administrators are competent, non-hostile and 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 the recorded audit trail.

OE.FILESEC

The TOE environment must protect configuration and other TSF data stored in files against any undetected unauthorised modification.

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.SINGIF

The connection provided by the TOE is the only one between the connected networks so that all information must flow through the TOE.

OE.RTS

The IT environment provides reliable timestamps.

4.3 Security Objectives Rationale

4.3.1 Coverage

The following table provides a mapping of TOE objectives to threats and policies, showing that each objective counters or enforces at least one threat or policy, respectively.

Objective	Threats / OSPs
O.AUDIT	P.AUDIT
O.MEDIAT	T.ASPOOF T.MEDIAT
O.CONFIG	P.CONFIG
O.SECSTA	T.INISEC

Table 1: Mapping of security objectives to threats and policies

The following table provides a mapping of the objectives for the Operational Environment to assumptions, threats and policies, showing that each objective holds, counters or enforces at least one assumption, threat or policy, respectively.

Objective	Assumptions / Threats / OSPs
OE.ADMINACC	A.ADMINACC
OE.ADMINPORT	A.ADMINPORT
OE.NOEVIL	A.NOEVIL
OE.AUDIT	TE.AUDIT
OE.FILESEC	TE.FILE
OE.PHYSEC	A.PHYSEC
OE.RELHARD	A.RELHARD
OE.SINGIF	A.SINGIF T.ASPOOF T.MEDIAT

Objective	Assumptions / Threats / OSPs
OE.RTS	A.RTS P.AUDIT

Table 2: Mapping of security objectives for the Operational Environment to assumptions, threats and policies

4.3.2 Sufficiency

The following rationale provides justification that the security objectives are suitable to counter each individual threat and that each security objective tracing back to a threat, when achieved, actually contributes to the removal, diminishing or mitigation of that threat:

Threat	Rationale for security objectives
T.ASPOOF	By demanding that the TOE must mediate (i.e. examine and control) 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. This requires the TOE to be the single interface between the networks (OE.SINGIF).
T.INISEC	By requiring well-defined restrictive default setting in O.SECSTA, an initial insecure configuration of the TOE is prevented and the threat T.INISEC of packets being routed through the firewall because the packet is not matched by any explicit firewall rule is removed.
T.MEDIAT	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 impermissible information sent through the TOE as in T.MEDIAT can be diminished to an acceptable level. This requires the TOE to be the single interface between the networks (OE.SINGIF).
TE.AUDIT	The threat TE.AUDIT that an administrator on the console cannot inspect this audit evidence is removed by demanding the possibility to view this log files in OE.AUDIT.
TE.FILE	Protection of files in the TOE IT-environment against undetected unauthorised modification as in OE.FILESEC diminishes the threat TE.FILE to an acceptable level.

Table 3: Sufficiency of objectives countering threats

The following rationale provides justification that the security objectives for the environment are suitable to cover each individual assumption, that each security objective for the environment that traces back to an assumption about the environment of use of the TOE, when achieved, actually contributes to the environment achieving consistency with the assumption, and that if all security objectives for the environment that trace back to an assumption are achieved, the intended usage is supported:

Assumption	Rationale for security objectives
A.ADMINACC	The protection of the administrative access to the TOE is supported by OE.ADMINACC

Assumption	Rationale for security objectives
A.ADMINPORT	The assumption that the administrative access to the TOE is limited to a dedicated port is supported by OE.ADMINPORT.
A.NOEVIL	The assumption of A.NOEVIL that administrators are non-hostile and trained is supported by OE.NOEVIL.
A.PHYSEC	By demanding physical security for the TOE in OE.PHYSEC the environment is consistent with the assumption of such security in A.PHYSEC.
A.RELHARD	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.
A.SINGIF	The assumption of information that cannot flow among internal and external networks without passing the TOE as in A.SINGIF is consistent with the objective for the environment, OE.SINGIF which demands that the TOE is the only connection between those networks is provided by the TOE.
A.RTS	The assumption about the provisioning of reliable time stamps by the environment is backed by the corresponding objective for the operational environment.

Table 4: Sufficiency of objectives holding assumptions

The following rationale provides justification that the security objectives are suitable to cover each individual organizational security policy, that each security objective that traces back to an OSP, when achieved, actually contributes to the implementation of the OSP, and that if all security objectives that trace back to an OSP are achieved, the OSP is implemented:

OSP	Rationale for security objectives
P.AUDIT	The policy to provide audit records for all security relevant actions performed by the TOE is implemented by the objective O.AUDIT which provides an audit mechanism and is supported by the objective OE.RTS to provide a reliable time source in the runtime environment.
P.CONFIG	The policy to provide the means to configure the NIFC is implemented by the objective O.CONFIG which supports the configuration of the NIFC.

Table 5: Sufficiency of objectives enforcing Organizational Security Policies

5 Extended Components Definition

The TOE itself does not support administrators, this is handled in the environment. Nevertheless, the TOE does support management by configuration files. To better model this, the family CFG is introduced in class FMT instead of using FMT_MSA/FMT_SMF.

5.1 Class FMT: Security management

5.1.1 TOE Configuration (FMT_CFG)

Family behaviour

This family defines requirements for TOE configuration that are independent of administrative roles.

Component levelling

FMT_CFG.1 specifies configurability of the TOE without the need of management roles.

FMT_CFG.1 is not hierarchical to any other component within the FMT_CFG family.

FMT_CFG.2 specifies TOE configuration value initialization.

FMT_CFG.2 is not hierarchical to any other component within the FMT_CFG family.

Management: FMT_CFG.1

There are no management activities foreseen.

Management: FMT_CFG.2

There are no management activities foreseen.

Audit: FMT_CFG.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Changes of the TOE configuration.

Audit: FMT_CFG.2

There are no audit events foreseen.

5.1.1.1 FMT_CFG.1 - Configuration of security functions

Hierarchical to: No other components.

Dependencies: No dependencies.

FMT_CFG.1.1 The TSF shall be capable of configuring the following security functions: [assignment: **list of functions to be configurable by the TSF**].

Rationale

The configuration of the TOE is specified with a configuration method that is not dependent on TOE supported roles.

5.1.1.2 FMT_CFG.2 - Static attribute initialisation

Hierarchical to: No other components.

Dependencies: No dependencies.

FMT_CFG.2.1 The TSF shall enforce the [assignment: **access control SFP, information flow control SFP**] to provide [selection, choose one of: **restrictive, permissive, [assignment: other property]**] default values for security attributes that are used to enforce the SFP.

FMT_CFG.2.2 The TSF shall support the specification of alternative initial values to override the default values used in enforcing the security policy.

Rationale

The defaults of the TOE configuration are specified with a configuration method that is not dependent on TOE supported roles.

6 Security Requirements

6.1 Network Information Flow Control Policy

The TOE implements the following network information flow control policy (NIFCP):

Subjects:

Users (external entities)
Send and/or receive information through the TOE.

Subject security attributes
Subjects are identified by IP or MAC addresses.

A user in this context is any entity (human or IT) outside the TOE that interacts (or may interact) with the TOE.

Information

Packets
Data sent, received or routed through the TOE.

Information security attributes
Source and Destination IP address; MAC address; the logical or physical network interface through which the network data entered the TOE; VLAN tag; Network protocol: IPv4, IPv6, ICMP, ICMPv6, ESP, AH; Protocol specific header information.

The policy allows or denies information flow according to the rules that use information security attributes to control the network information flow.

6.2 TOE Security Functional Requirements

The following table shows the Security functional requirements for the TOE, and the operations performed on the components according to CC part 2: iteration (Iter.), refinement (Ref.), assignment (Ass.) and selection (Sel.).

Security functional group	Security functional requirement	Source	Operations			
			Iter.	Ref.	Ass.	Sel.
FAU - Security audit	FAU_GEN.1 Audit data generation	CC Part 2	No	Yes	Yes	Yes
	FAU_STG.1 Protected audit trail storage	CC Part 2	No	No	No	Yes
	FAU_STG.3 Action in case of possible audit data loss	CC Part 2	No	Yes	Yes	No
FDP - User data protection	FDP_IFC.1 Subset information flow control	CC Part 2	No	No	Yes	No
	FDP_IFF.1 Simple security attributes	CC Part 2	No	No	Yes	No
FMT - Security management	FMT_CFG.1 Configuration of security functions	ECD	No	No	Yes	No
	FMT_CFG.2 Static attribute initialisation	ECD	No	No	Yes	Yes

Table 6: Security functional requirements for the TOE

6.2.1 Security audit (FAU)

6.2.1.1 Audit data generation (FAU_GEN.1)

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) **The audit events specified in table 7**.

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 (if applicable), and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the ~~PP~~/ST, **the information listed in table 7**.

Event	Type	Information
Config daemon started	CONFIGSTART	DateTime, Message string
Config daemon stopped	CONFIGSTOP	DateTime, Message string
Activation of configuration started	ACTIVATING	DateTime, Message string
Activation of configuration finished	ACTIVATED	DateTime, Message string
Configuration element added	ADD	DateTime, Message string, Added value
Configuration element deleted	DELETE	DateTime, Message string, Deleted value
Configuration element changed	CHANGE	DateTime, Message string, Old Value, New Value
Generic packet dropped event	DROPPED	DateTime, Message string, PCAP formatted packet capture
IP was blacklisted	BLACKLISTED	DateTime, Message string, Source IP
New connection established	NEW	DateTime, Message string, Destination IP, Service, Protocol
New local connection established	NEWLO	DateTime, Message string, Destination IP, Service, Protocol
Connection ended	CLOSED	DateTime, Message string, Destination IP, Service, Protocol, Connection time

Table 7: Audit Events

6.2.1.2 Protected audit trail storage (FAU_STG.1)

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

FAU_STG.1.2 The TSF shall be able to **prevent** unauthorised modifications to the stored audit records in the audit trail.

Application Note:

The TOE prevents the unauthorised modifications (including deletion) of the audit trail by specifying the appropriate restrictive permissions when creating the audit trail files.

6.2.1.3 Action in case of possible audit data loss (FAU_STG.3)

FAU_STG.3.1 The TSF shall **generate a notification** if the *available space for the* audit trail ~~exceeds~~*reaches* **a configured limit**.

Application Note:

When the audit trail reaches the configuration file defined limit, a warning message is generated in the audit trail so that any post-processing application can pick up the message. The configuration can also specify that the TOE generates a warning e-mail to an administrator configured address.

Application Note: *A log rotate mechanism is employed that creates a new log file weekly. Four weeks of logs are stored.*

6.2.2 User data protection (FDP)

6.2.2.1 Subset information flow control (FDP_IFC.1)

FDP_IFC.1.1 The TSF shall enforce the **network information flow control policy (NIFCP)** on

- **a) Subjects: entities identified by IP and/or MAC addresses;**
- **b) Information: data packets to be transferred between entities;**
- **c) Operations: the transfer of data packets to and from entities via network connections.**

6.2.2.2 Simple security attributes (FDP_IFF.1)

FDP_IFF.1.1 The TSF shall enforce the **network information flow control policy (NIFCP)** based on the following types of subject and information security attributes:

- **a) Source and Destination IP address**
- **b) MAC address**
- **c) The logical or physical network interface through which the network data entered the TOE and the associated VLAN tag of the interface**
- **d) Network protocol: IPv4, IPv6, ICMP, ICMPv6, ESP, AH**
- **e) Protocol specific header information**

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: **If the packet filter matches the analyzed packet and the rule accepts the packet, the packet is forwarded according to the network protocol stack's configured behavior**.

FDP_IFF.1.3 The TSF shall enforce the **following rules:**

- **a) Identification of network packets using one or more of the following concepts:**
  1. **Subject security attribute matching;**
  2. **Information security attribute matching;**
  3. **Matching based on the state of a TCP connection;**

4. **Statistical analysis matching;**

b) **Performing one of the following actions with identified network data:**

1. **Discard the network data without any further processing, without sending a notification to the sender (DROP);**
2. **Discard the network data without any further processing, with sending a notification to the sender (REJECT);**
3. **Allow the network data to be processed unaltered by the TOE according to the routing information maintained by the TOE (ACCEPT);**

FDP_IFF.1.4 The TSF shall explicitly authorise an information flow based on the following rules: **If the network data is not matched by the rule set and the default rule of the packet filter is ACCEPT then the data is forwarded unaltered based on the normal operation of the host system's networking stack**.

FDP_IFF.1.5 The TSF shall explicitly deny an information flow based on the following rules: **If the network data is not matched by the rule set, one of the following default rules applies:**

a) **DROP: the data is discarded;**

b) **REJECT: then the data is discarded and a notification is returned to the sender.**

Application Note:

The TOE is shipped with an explicit deny policy that drops packets, but can be configured for an explicit allow policy.

6.2.3 Security management (FMT)

6.2.3.1 Configuration of security functions (FMT_CFG.1)

FMT_CFG.1.1 The TSF shall be capable of configuring the following security functions:

a) **Network Information Flow Control**

6.2.3.2 Static attribute initialisation (FMT_CFG.2)

FMT_CFG.2.1 The TSF shall enforce the **network information flow control policy (NIFCP)** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_CFG.2.2 The TSF shall support the specification of alternative initial values to override the default values used in enforcing the security policy.

Application Note:: *The TOE does not distinguish between audit records for normal configuration changes or changes to defaults. Changes to defaults are recorded as regular configuration changes.*

6.3 Security Functional Requirements Rationale

6.3.1 Coverage

The following table provides a mapping of SFR to the security objectives, showing that each security functional requirement addresses at least one security objective.

Security Functional Requirements	Objectives
FAU_GEN.1	O.AUDIT
FAU_STG.1	O.AUDIT
FAU_STG.3	O.AUDIT
FDP_IFC.1	O.MEDIAT
FDP_IFF.1	O.MEDIAT
FMT_CFG.1	O.CONFIG
FMT_CFG.2	O.SECSTA

Table 8: Mapping of security functional requirements to security objectives

6.3.2 Sufficiency

The following rationale provides justification for each security objective for the TOE, showing that the security functional requirements are suitable to meet and achieve the security objectives:

Security objectives	Rationale
O.AUDIT	The objective is satisfied by the audit generation in FAU_GEN.1. Audit records are protected via FAU_STG.1 and the TOE configuration can specify the actions to be taken by the TOE when the audit trail can not be written via FAU_STG.3.
O.MEDIAT	The objective is satisfied by the instantiation of the network flow control policy in FDP_IFC.1 and FDP_IFF.1.
O.CONFIG	The network information flow control policy is configured via FMT_CFG.1.
O.SECSTA	Secure default values are mandated through FMT_CFG.2

Table 9: Security objectives for the TOE rationale

6.3.3 Security Requirements Dependency Analysis

Dependencies within the EAL4 package selected for the security assurance requirements have been considered by the authors of CC Part 3 and are not analyzed here again. The included component on flaw remediation, ALC_FLR.1, has no dependencies on other requirements.

The security functional requirements in this Security Target do not introduce dependencies on any security assurance requirement; neither do the security assurance requirements in this Security Target introduce dependencies on any security functional requirement.

The following table demonstrates the dependencies of SFRs modeled in CC Part 2 and how the SFRs for the TOE resolve those dependencies:

Security Functional Requirement	Dependencies	Resolution
FAU_GEN.1	FPT_STM.1	Not resolved: The TOE itself does not generate reliable time stamps but uses them from the environment (OE.RTS).
FAU_STG.1	FAU_GEN.1	FAU_GEN.1
FAU_STG.3	FAU_STG.1	FAU_STG.1
FDP_IFC.1	FDP_IFF.1	FDP_IFF.1
FDP_IFF.1	FDP_IFC.1	FDP_IFC.1
	FMT_MSA.3	Not resolved: The TOE does not support administrative roles. Therefore the configuration aspects are covered by FMT_CFG.2 instead of FMT_MSA.3.
FMT_CFG.1	No dependencies.	
FMT_CFG.2	No dependencies.	

Table 10: TOE SFR dependency analysis

6.4 Security Assurance Requirements

The security assurance requirements for the TOE are the Evaluation Assurance Level 4 components as specified in [CC] part 3, augmented by ALC_FLR.1.

The following table shows the Security assurance requirements, and the operations performed on the components according to CC part 3: iteration (Iter.), refinement (Ref.), assignment (Ass.) and selection (Sel.).

Security assurance class	Security assurance requirement	Source	Operations			
			Iter.	Ref.	Ass.	Sel.
ADV Development	ADV_ARC.1 Security architecture description	CC Part 3	No	No	No	No
	ADV_FSP.4 Complete functional specification	CC Part 3	No	No	No	No
	ADV_IMP.1 Implementation representation of the TSF	CC Part 3	No	No	No	No
	ADV_TDS.3 Basic modular design	CC Part 3	No	No	No	No
AGD Guidance documents	AGD_OPE.1 Operational user guidance	CC Part 3	No	No	No	No
	AGD_PRE.1 Preparative procedures	CC Part 3	No	No	No	No
ALC Life-cycle support	ALC_CMC.4 Production support, acceptance procedures and automation	CC Part 3	No	No	No	No
	ALC_CMS.4 Problem tracking CM coverage	CC Part 3	No	No	No	No
	ALC_DEL.1 Delivery procedures	CC Part 3	No	No	No	No

Security assurance class	Security assurance requirement	Source	Operations			
			Iter.	Ref.	Ass.	Sel.
	ALC_DVS.1 Identification of security measures	CC Part 3	No	No	No	No
	ALC_FLR.1 Basic flaw remediation	CC Part 3	No	No	No	No
	ALC_LCD.1 Developer defined life-cycle model	CC Part 3	No	No	No	No
	ALC_TAT.1 Well-defined development tools	CC Part 3	No	No	No	No
ASE Security Target evaluation	ASE_INT.1 ST introduction	CC Part 3	No	No	No	No
	ASE_CCL.1 Conformance claims	CC Part 3	No	No	No	No
	ASE_SPD.1 Security problem definition	CC Part 3	No	No	No	No
	ASE_OBJ.2 Security objectives	CC Part 3	No	No	No	No
	ASE_ECD.1 Extended components definition	CC Part 3	No	No	No	No
	ASE_REQ.2 Derived security requirements	CC Part 3	No	No	No	No
	ASE_TSS.1 TOE summary specification	CC Part 3	No	No	No	No
ATE Tests	ATE_COV.2 Analysis of coverage	CC Part 3	No	No	No	No
	ATE_DPT.1 Testing: basic design	CC Part 3	No	No	No	No
	ATE_FUN.1 Functional testing	CC Part 3	No	No	No	No
	ATE_IND.2 Independent testing - sample	CC Part 3	No	No	No	No
AVA Vulnerability assessment	AVA_VAN.3 Focused vulnerability analysis	CC Part 3	No	No	No	No

Table 11: Security assurance requirements

6.5 Security Assurance Requirements Rationale

The basis for the justification of EAL4 is the threat environment experienced by the typical consumers of the TOE. This matches the package description for EAL4 (enhanced-basic).

In addition, the evaluation assurance level has been augmented with ALC_FLR.1 commensurate with the flaw remediation capabilities offered by the developer beyond those required by the evaluation assurance level.

7 TOE Summary Specification

7.1 TOE Security Functionality

7.1.1 Network Information Flow Control

The Network Information Flow Control Policy (NIFCP) is enforced by the TOE providing a filtering mechanism that is integrated into the networking stack of the underlying system. All packets flowing to, from or through the system are subject to this filtering mechanism.

The filtering mechanism is implemented via IPTables. It uses a high level configuration language that abstracts from the actual rule set. Instead configuration happens in terms of a network graph consisting of nodes and connections between them. Allowed services, forwarding of packets and connection constraints (like limiting of parallel connections, connection quotas and so on) can all be configured in terms of this network graph. Eventually the graph is translated into a concrete set of enforceable IPTables rules. Additionally enhanced logging and statistics about the packet flow are provided. For example, log information is generated with knowledge about the configured network graph, so that the raw log data can be improved by adding user supplied node and edge labels (office names, user names, ...).

The rules generated for IPTables use the first match for the packet handling decision. The high-level configuration is translated into the IPTables rules so that the best match is the first match.

IPTables handles packets at the network layer and up (IP and higher layers) but also supports filtering at MAC layer. Where exactly the enforcement happens is of no concern for the user supplied configuration, those details are hidden to reduce the complexity of the configuration.

In addition to the connection-oriented filtering, filters can also be set on connection and rate limits which can be used to protect systems that are shielded by the firewall from DOS attacks.

When the configured packet filter connection thresholds are reached for a specific system, the filter can automatically add a rule to block the system that reached this threshold in quota rule.

The filters are using hooks in several places within the IP stack of the underlying system. This allows the filter to provide packet handling decisions based on the Information Flow Control Policy at all points in the system from inception via routing to the final sending.

The filter uses the incoming interface, MAC, VLAN tag and IP Addresses as well as packet header contents depending on the protocol (IP (both v4, v6), ICMP, ICMPv6, UDP, TCP, IPsec (AH and ESP)) and statistical information to allow or deny packets according to the configured rule base.

This security functionality implements the requirements from FDP_IFC.1 and FDP_IFF.1.

7.1.2 Audit

The filtering mechanism can generate audit events for all actions that are performed on packets. The configuration daemon (*stated*) generates audit events for changes to the configuration and the activation of a specific configuration. All the events are collected by user space daemons (*ulogd*, *rsyslogd*) and stored in the TOE's environment with restrictive DAC settings.

Theoretically all packets inspected by the filter could generate audit events with the appropriate configuration. But that would lead to impractical audit log files with huge amounts of redundant information. Therefore the available audit events focus on connections instead of individual packets if connections can be detected from the packet flow.

In addition to packet and connection oriented logging, reaching configured thresholds (quota limits) for connections will also generate audit events.

Changes to the configuration also generate audit records.

The audit trail is protected by the TOE environment, the protection needed is specified via the permission bits set by the audit log daemon when creating the audit trail file.

The audit trail is implemented as multiple files. New log files are generated every week and up to four weeks are stored. The *logrotate* utility in the TOE environment handles this task.

When a configuration-specified threshold for the available audit trail storage is reached, the TOE generates a warning log entry and sends an alert e-mail to an administrator configured address.

This security functionality implements the requirements from FAU_GEN.1, FAU_STG.1 and FAU_STG.3.

The audit mechanism relies on the underlying system to provide a reliable time source for the audit records.

7.1.3 Configuration

The Network Information Flow Control Policy and the events to be audited are loaded from a file in the TOE environment. All of the high level configuration information is stored in a database in the TOE environment which is read by the TOE via a parser and translated into the detailed rules for the enforcement component. Access to the configuration information on disk is controlled by the IT environment granted only to the environments authenticated administrators.

The abstract rule configuration can be changed at any time during the operation of the TOE via a command that reads updated configuration information from a file. These changes will be registered and stored in a configuration database. A separate activation step is necessary to convert this configuration into an actual rule set to be used by the filtering mechanism. The activation mechanism can also be used to recall older configurations that are stored in the configuration database.

The configuration file can be edited with any text editor, it uses JSON. Within the file, objects and object relationships are defined.

Network entities (hosts or networks defined by IP address ranges) are defined by address and interface over which they are connected to the firewall. Connections are defined by specifying source and destination network entities and the allowed and prohibited communication forms

The initial defaults prevent any traffic to pass, therefore the system maintains a secure state also during startup before loading of the configured rules.

This security functionality implements the requirements from FMT_CFG.1 and FMT_CFG.2.

8 Abbreviations, Terminology and References

8.1 Abbreviations

DAC
Discretionary Access Control

DOS
Denial Of Service

8.2 Terminology

This section contains definitions of technical terms that are used with a meaning specific to this document. Terms defined in the [CC] are not reiterated here, unless stated otherwise.

Administrator
Humans that interact with the TOE environment to provide a configuration file for the TOE.

JSON

JavaScript object notation, a structured way to define objects.

PCAP
Packet **Cap**ture, usually a shorthand to refer to functions and formats used by *libpcap*.

User
Humans or machines interacting with the TOE via network interfaces.

8.3 References

CC	**Common Criteria for Information Technology Security Evaluation**	
	Version	3.1R3
	Date	July 2009
	Location	http://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R3.pdf
	Location	http://www.commoncriteriaportal.org/files/ccfiles/CCPART2V3.1R3.pdf
	Location	http://www.commoncriteriaportal.org/files/ccfiles/CCPART3V3.1R3.pdf