genua

genugate firewall 10.0
Security Target

genua GmbH — Kirchheim

2021-03-26

Version 10.0.9(db2b731)

Contents

1 ST Introduction **4**
1.1 ST Reference . . . 4
1.2 TOE Reference . . . 4
1.3 TOE Overview . . . 4
1.3.1 Required non-TOE Hardware/Software/Firmware . . . 6
1.4 TOE Description . . . 6
1.4.1 The Application Level Gateway . . . 8
1.4.2 The Packet Filter . . . 11
1.4.3 High Availability (genugate cluster) . . . 11
1.4.4 Physical Scope . . . 11
1.4.5 Logical Scope . . . 14
1.5 Estimated End-Of Life of the Product . . . 16

2 CC Conformance Claim **17**
2.1 CC Conformance Claim . . . 17
2.2 PP Claim, Package Claim . . . 17
2.3 Conformance Rationale . . . 17

3 Security Problem Definition **18**
3.1 Users . . . 18
3.2 Assets . . . 18
3.3 Threats . . . 19
3.4 Assumptions . . . 19
3.5 Organisational Security Policies . . . 20

4 Security Objectives **21**
4.1 Security Objectives for the TOE . . . 21
4.2 Security Objectives for the Environment . . . 22
4.3 Security Objectives Rationale . . . 23
4.3.1 Threat Rationale . . . 23
4.3.2 Assumption Rationale . . . 24
4.3.3 Policy Rationale . . . 26

5 Extended Components Definition **27**
5.1 Class FAU: Security audit . . . 27
5.1.1 Security audit data generation (FAU_GEN) . . . 27
5.2 Class FIA: Identification and authentication . . . 28
5.2.1 User authentication (FIA_UAU) . . . 28
5.3 Class FPT: Protection of the TSF . . . 29
5.3.1 Simple Self Test (FPT_SST) . . . 29
5.3.2 TOE Update (FPT_UPD) . . . 30
5.4 Class ALC: Life-cycle support . . . 32
5.4.1 Patch Management (ALC_PAM) . . . 32

6 Security Requirements **34**
6.1 Security Functional Requirements . . . 34
6.1.1 Class FAU: Security audit . . . 34
6.1.2 Class FCS: Cryptographic support . . . 36
6.1.3 Class FDP: User data protection . . . 36
6.1.4 Class FIA: Identification and authentication . . . 43
6.1.5 Class FMT: Security management . . . 45
6.1.6 Class FPT: Protection of the TSF . . . 48
6.2 Security Assurance Requirements . . . 51
6.3 Security Functional Requirements Rationale . . . 52
6.3.1 Objectives . . . 55
6.3.2 New or tailored SFR . . . 62
6.4 Security Assurance Requirements Rationale . . . 63

7 TOE Summary **66**
7.1 TOE Summary Specification . . . 66
7.1.1 SF_SA: Security audit . . . 66
7.1.2 SF_DF: Data flow control . . . 67
7.1.3 SF_IA: Identification and Authentication . . . 68
7.1.4 SF_SM: Security management . . . 70
7.1.5 SF_PT: Protection of the TSF . . . 71
7.1.6 SF_PI: Patch installation . . . 72
7.2 Self-Protection against Interference and Logical Tampering . . . 72
7.3 Self-Protection against Bypass . . . 73

8 Use of Cryptographic Functions **73**

A Evaluation Methology for ALC_PAM **73**
A.1 Objectives . . . 73
A.2 Input . . . 74
A.3 Action ALC_PAM.1.1E . . . 74
A.4 Implied evaluator action ALC_PAM.1.2D . . . 77
A.5 Implied evaluator action ALC_PAM.1.3D . . . 78

B References **79**

C Acronyms **80**

1 ST Introduction

1.1 ST Reference

	ST Reference
ST Title	genugate firewall 10.0 Security Target
Version	10.0.9
Developer	genua GmbH
Date	2021-03-26

1.2 TOE Reference

	TOE Reference
TOE Title	genugate firewall 10.0
TOE Reference	genugate 10.0 software
Product Name	genugate 10.0 Z

1.3 TOE Overview

The TOE **genugate firewall 10.0** is part of a larger product, the firewall **genugate 10.0 Z**, which consists of hardware and software. The TOE genugate firewall 10.0 itself is part of the shipped software. The operating system is a modified OpenBSD.
To mitigate hardware failures the genugate has a high availability option where two or more genugate systems are operating in parallel and take over a failing system.
genugate 10.0 Z is a combination of an application level gateway (ALG) and a packet filter (PFL), which are implemented on two different systems (see figure 1). It is thus a two-tiered firewall.
Besides the network interface to the PFL, the ALG has (at least) three more interfaces to connect to the external network, the administration network and the demilitarized zone (DMZ). For the high availability option, the ALG needs another network interface for the HA network. The PFL has a second interface which is connected to the internal network, and optional interfaces for further DMZs.
The aim of the firewall is to control the IP-traffic between the different connected networks. Therefore the ALG uses proxies that implement filter policies in order to control all data transmitted between the different networks, while the PFL uses packet filtering as an additional means to control all data that is sent to and from the internal network.
The TOE, **genugate firewall 10.0**, consists of the software that implements the IP traffic control and related functionality of the firewall. This includes the proxies, the modified OpenBSD kernel modules IP-stack, packet filter, but also other supportive functionality as logging of security events (see the next section for a more detailed definition of the TOE scope and boundary).
The TOE has a special maintenance mode. During normal operation IP packets are handled as usual and the file system is secured by the BSD file flags. In maintenance mode, however, the BSD flags can be altered. In this mode all IP packets are dropped for security reasons.
The genugate product family includes the following security features:

- The TOE supports IPv4 and IPv6.
- The ALG does not perform IP forwarding but uses socket splicing as a fast transport mechanism (see below).

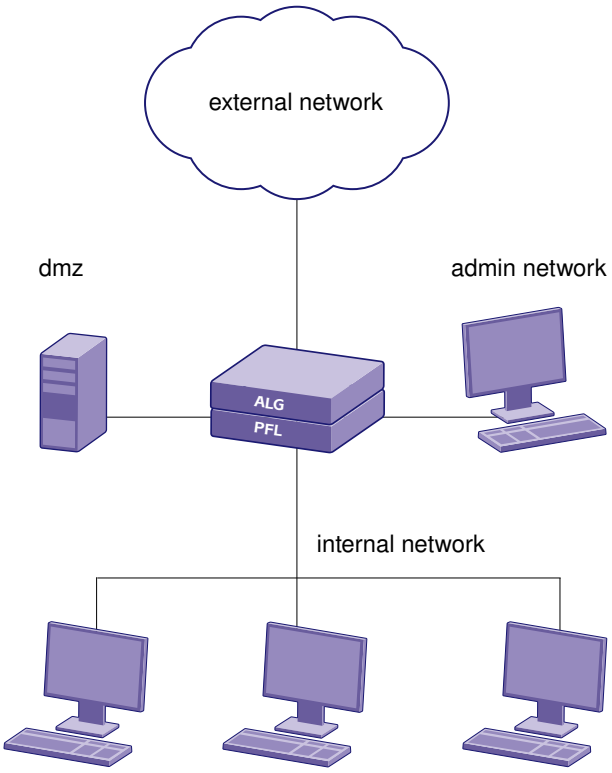


Figure 1: genugate 10.0 Z overview.

- The modified OpenBSD kernel performs extra spoofing checks. The source and destination address of the IP packet are checked against the IP address (and netmask) of the receiving interface.
- The modified OpenBSD kernel logs events related to firewall security that occur while checking incoming IP packets and keeps statistic counters for other events.
- The filter rules of the PFL cannot be modified during normal operation.
- Proxies that accept connections from the connected networks run in a restricted runtime environment.
- The log files are analysed online.
- The administrators are notified about security relevant events.
- File system flags prohibit the deletion of the most important log messages.
- The internal network is protected by a two-tiers security architecture that filter on different levels of the network stack (ALG and PFL).
- The TOE has a special maintenance mode. During normal operation IP packets are handled as usual and the file system is secured by the BSD flags. In maintenance mode, however, the BSD flags can be altered for maintenance operation. In this mode all IP packets are dropped for security reasons.

- To mitigate hardware failures the genugate has a high availability option where two or more genugate systems are operating in parallel and take over a failing system. The different systems synchronize their configuration with one another. The genugate provides two certified mechanisms, OSPF and CARP failover.

1.3.1 Required non-TOE Hardware/Software/Firmware

The product is based on OpenBSD 6.6 that runs on a large scale of hardware using different INTEL compatible processors. The ALG needs at minimum an Intel Celeron with 1 GB memory and four 1GBit network interfaces (the high availability option needs at least five interfaces). The PFL needs an Intel Celeron with 512 MB memory and two 1GBit network interfaces. Nonetheless the hardware is selected by the manufacturer in order to guarantee proper execution of the product.
The currently distributed hardware versions are the genugate S, the genugate M, the genugate L, revisions 2.0 and 3.0. These hardware versions are in scope for this certification.
The hardware revision 1.0 for the genugate S, the genugate M and the genugate L are not in scope for this certification although the software runs with the same security functions.
There are also the legacy versions genugate 200, genugate 400, genugate 600 and genugate 800 in the field with hardware revision 6 and 7 which are out of scope for the current certification. The genugate firewall 10.0 runs on this hardware with the same functionality and security measures, but running the software on the legacy hardware has not been evaluated.
The proxies and other user space programs on the ALG are based on Perl 5.28 which is distributed with the product.
For the high availability option using OSPF a correctly configured OSPF router is needed in the internal network.

1.4 TOE Description

The TOE **genugate firewall 10.0** is used to control the connections and data transfer between different networks, where each network has different security needs and different threat levels for the other networks. **genugate 10.0 Z** is a combination of an application level gateway (ALG) and a packet filter (PFL), which are implemented on two different systems. It is thus a two-tiered firewall for connections into the internal network.
The TOE can be configured in such a way that the security needs for each network are optimally met. A standard configuration consists of the following networks connected to the TOE:

internal network: This is the network that has to be secured against attacks from the external network. Usually only a few services from the internal network are accessible from the external network, secured by user authentication. This is the network that is secured by both the ALG and the PFL, using filtering mechanisms at two different levels of the IP stack. This network is usually controlled by a defined security policy.

external network: This is the most insecure network, e. g. the internet. In general, no security policy exists, and all kind of attacks can occur in this network.

administrative network: his network is used to allow a secure administration of the TOE. This network is isolated from all other networks and only administrators have access. The usual access is through

the HTTPS web interface, but an SSH access for debugging and maintenance operation is also available.

demilitarized zone: his network allows access to common services from the external network, without the need to open the internal network. Usually, Web- and FTP-servers are installed in this network. This network is usually controlled by a defined security policy.

HA network: his internal network is necessary for the high availability option. It is used to synchronize the configuration between the systems.

The TOE includes the following security features:

- The TOE supports IPv4 and IPv6. However, the mcastudprelay supports only IPv4. The internal HA synchronisation network must use IPv4 addresses.
- The ALG does not perform IP forwarding but uses socket splicing for TCP connections and UDP datagrams when appropriate. The connection setup is handled in user space, where information flow control policies are enforced. If the TCP-connections/UDP datagrams pass the control checks, the sockets are set to a “fast"’ mode where no data is copied to user space and back. This mode should not be confused with IP forwarding, where the IP packets are copied between the networks. The socket splicing reconstructs the whole TCP stream/the UDP contents before sending the data.
- The modified OpenBSD kernel performs extra spoofing checks. The source and destination address of the IP packet are checked against the IP address (and netmask) of the receiving interface.
- The modified OpenBSD kernel logs events related to firewall security that occur while checking incoming IP packets and keeps statistics for other events.
- The filter rules of the PFL cannot be modified during normal operation.
- Proxies that accept connections from the connected networks run in a restricted runtime environment.
- All central processes of the ALG are controlled by the process master that monitors the system and keeps it running. In case of strange behaviour the process master can take actions.
- The log files are analysed online and the administrators are notified about security relevant events.
- The log files are intelligently rotated so that they avoid filling the available space but the administrator still can see recent log entries and all events of the process master and the online analysis. There are two classes of log files, the rotated and the flagged. The rotated log files are rotated automatically, based on size and time. The flagged log files are only rotated in maintenance mode with the acknowledgement of the administrator.
- File configuration of the system flags prohibit the deletion of the most important log messages.
- The internal network is protected by a two-tiers security architecture that filter on different levels of the network stack (ALG and PFL).
- The SSH relay intercepts SSH connections, can filter selected SSH protocol messages and can authenticate users. The cryptographic operations of the relay are not part of the certification.

- The TOE has a special maintenance mode. During normal operation IP packets are handled as usual and the file system is secured by the BSD file flags. In maintenance mode, however, the BSD file flags can be altered for maintenance operation. In this mode all IP packets are dropped for security reasons.
- To mitigate hardware failures the genugate has a high availability option where two or more genugate systems are operating in parallel and take over a failing system. The different systems synchronize their configuration with one another.

1.4.1 The Application Level Gateway

The ALG uses policies to provide and control connections between the different networks. The policies are realised by user-space proxies, called relays. These relays are necessary, because the kernel of the ALG has no capabilities to forward IP packets. Without socket splicing, all IP traffic has to be reassembled and transferred to user space by the kernel. The relays examine the data and perform most of the filtering and controlling function. The protocol-specific relays have enough knowledge about the respective protocol in order to filter possible threatening or insecure protocol elements. The relays implement several access control lists that allow a fine grained control for the usage of services. All relays can be transparent with respect to the source and/or destination address, so that the ALG can be configured transparent with respect to IP addressing. The ALG checks for source or destination spoofing attacks.
Socket splicing optimizes the handling of TCP connections/UDP packets through the ALG. After the initial flow control checks on connection setup, the relays can switch to socket splicing mode. Then the data that would only be copied from kernel mode to application mode and back is kept in kernel memory. The connections are handled by the kernel like all traffic but instead of being copied to user space it is directly directed to the output socket. Socket splicing should be strictly distinguished from IP forwarding. Using IP forwarding, no packet reassembly is done; and all packets are copied verbatim to the outgoing socket including their IP headers, without further checks. With socket splicing, the TCP data stream/UDP contents is extracted out of the IP packets with all associated tests and checks and new IP packets are created by the kernel on output. Socket splicing is not applied for protocols where the whole data stream must be checked. So it is not feasible for protocols that use the virus scanner or that filter HTML.
The generic relays for UDP and TCP can apply a protocol conformance filter (PCF), that match the protocol data at the beginning of the connection against regular expressions. If the match fails, the relays finish the connection.
The administrator can also configure meta-policies which are relays with predefined attributes for common use cases.
The TOE provides proxy support for the following services/policies implemented by the respective relays:

IP: This policy can be used for all IP protocols (besides ICMP ECHO, UDP, or TCP, which are supported by their own proxies). It is a very generic proxy and has no knowledge about any application level protocol.

The iprelay implements this policy.

PING: This policy is used if the ALG should transmit ICMP ECHO REQUEST and ICMP REPLY packets from one network into another.

The pingrelay implements this policy.

UDP: This policy is implemented by a generic proxy than can be used for almost any service that is based on UDP.

The udprelay implements this policy.

This policy knows the following PCF: DNS, MSSQL.

TCP: This policy is implemented by a generic proxy that can be used for services based on TCP. It has no knowledge about application level protocols unless filters are configured that check for a basic protocol conformance by applying regular expressions at the beginning of the communication. It can handle TLS connections.

The tcprelay implements this policy.

This policy knows the following PCF: BGP_v4, DNS, Fernwartungs_App, IMAP_v4, LDAP, MSSQL, MySQL, POP3, PostgreSQL, PostgreSQL_SSL, PPTP, RDP, SMB, SSH, SSH_v2, SSL, SSL_no_v3, TeamViewer, VNC.

SMTP: This policy is implemented by an application specific proxy for the SMTP protocol. All protocol commands are analysed and can be filtered. The mail header and bodies can be filtered. It contains functionality to filter SPAM mail. It has an interface to an optional virus scanner. SMTP authentication can optionally be configured.

The smtprelay implements this policy.

SMTP2SMTP: This policy is implemented by an application specific proxy for the SMTP protocol. All protocol commands are analysed and can be filtered. The mail header and bodies can be filtered. It contains functionality to filter SPAM mail. It has an interface to an optional virus scanner. The SMTP2SMTP relay does not authenticate the users itself, but relies on the responses of the remote MTA. In contrast to the SMTP relay the SMTP2SMTP relay does not queue the mails to `postfix`, but directly connects to the SMTP server.

The smtp2smtprelay implements this policy.

IMAP: This policy is implemented by an application specific proxy for the IMAP and IMAPS (meta-policy) protocols. All protocol commands are analysed and can be filtered. It has an interface to an optional virus scanner.

The imaprelay implements this policy.

POP: This policy is implemented by an application specific proxy for the POP and POPS (meta-policy) protocols. All protocol commands are analysed and can be filtered. It has an interface to an optional virus scanner.

The poprelay implements this policy.

SIP: This policy is implemented by an application specific proxy for the SIP and SIPS (meta-policy) protocols. All protocol commands are analysed and can be filtered.

The siprelay implements this policy.

SSH: This policy is implemented by an application specific proxy for the SSH protocol. It intercepts SSH connections, can filter selected SSH protocol messages and can authenticate users.

The sshrelay implements this policy.

WWW: This policy is implemented by an application specific proxy for the HTTP protocol and its application data. This proxy analyses the HTTP protocol headers and the application data. The content-type of the application data can be used to either filter text data like HTML or to scan binary data for viruses. It can handle TLS connections.

The wwwrelay implements this policy.

The WWWserver is a meta-policy that filters request methods.

HTTP: This policy is implemented by an application specific proxy for the Websocket and/or the HTTP protocol and its application data. This proxy analyses the HTTP protocol headers and the application data. It cannot be configured directly but only via meta-policies.

The httprelay implements this policy.

The Webservice meta-policy can control SOAP and WSDL services by validation against XML schema files that are uploaded onto the genugate. It can handle TLS connections.

The opcuahttp meta-policy validates OPC UA messages.

FTP: This policy is implemented by an application specific proxy for the FTP protocol. All protocol commands are analysed and can be filtered. It has an interface to an optional virus scanner.

The ftprelay implements this policy.

TELNET: This policy is implemented by an application specific proxy for the TELNET protocol. All protocol commands are analysed and can be filtered.

The telnetrelay implements this policy.

MCASTUDP: This policy is implemented by a generic proxy for UDP multicast packets using IPv4. It filters IGMP packets based on the multicast group and allows or blocks multicast UDP packets according to the current group membership. The relay needs support from the `igmpproxy` at the PFL which is needed to properly route the multicast UDP packets on the PFL.

The mcastudprelay implements this policy.

meta-Policies: Besides the already mentioned meta-policies, there are further meta-policies that use the low-level TCP-, UDP- and IP-policies. These are bgp, dns, dnsserver, imapfilter, ipsec, ldap, mssql, mysql, ntpserver, opcuatcp, postgresql, pptp, rdp, smb, snmpserver, snmptrap, teamviewer, vnc, and waf. These are combinations of different policies preconfigured for the respective service.

Depending on the policy, the tcprelay, udprelay and/or iprelay implement the policy.

All relays are highly configurable. The preferred configuration method is through HTML forms at the administrative interface that are transported by secure HTTPS-connections in the administration network. An alternative is a REST oriented interface, that can be used by automated procedures.
User identification and authentication can be configured in two ways. Some relays have support for authentication in the respective protocol. These relays can authenticate their users against authentication servers. The side channel authentication allows the usage of special configured relays after user identification at a special web form at the TOE.
The TELNET and FTP protocols are only supplied for legacy applications. It should be stressed that the protocols TELNET and FTP are not considered secure if they are employed without further security measures. They transmit the user name and password in plain text and can be sniffed with very small

effort. The same concerns apply to the SMTP authentication in specific configurations. The security claims for the TOE only apply if the protocols are sufficiently secured.
Unencrypted SNMP management should only be made from sufficiently secure networks, because the SNMP packets may contain sensitive information. An alternative is strongly encrypted SNMPv3.
The ALG has an interfaces to communicat with a virus scanner (either local or remote). It also has an interface to a web site classification service. (Advanced Web Categories, AWC).

1.4.2 The Packet Filter

The internal network has high security needs and is therefore not directly connected to the ALG, but is connected to the PFL. The PFL has at least two network interfaces. One of them is connected to the ALG with a cross cable. The (small) network is called the cross network. The other interface connects to the internal network.
The PFL works as packet filter with a set of filter rules. Only configured TCP connection requests from the cross network are allowed, but there is no default restriction for packets from the internal network. In order to allow connections into the internal network, extra rules have to be added by administrators.
The PFL is a minimalistic system. In the certified mode it boots from a removable USB stick and has no other permanent memory. The medium is configured and created at the ALG. Physical access is needed to write the medium at the ALG, transfer it from the ALG to the PFL, and reboot the PFL with the new configuration.
The configuration of the PFL is done through the admin web at the ALG.

1.4.3 High Availability (genugate cluster)

For a high availability (HA) setup, the HA option is installed on two or more genugates (peers) and they are connected by a separate HA network that is used to synchronise the configuration and negotiate the active HA nodes. If a system fails some other system takes over its services and IP addresses.
For the variant using OSPF an external OSPF router is needed in the internal network. Figure 2 gives an overview for two parallel systems, although more than two are possible.
The synchronisation of the configuration in the HA network uses IPsec with preshared keys to encrypt the communication.
Optionally the cross networks of the genugate peers can be united into one cross network. Then cross cables can no longer be used and switches must be incorporated. This setup avoids a full HA take over if only one PFL fails.
The CARP setup can operate in two modes, failover and balancing. A certified setup can only use the failover mode.
The CARP setup can also be used in a PAP configuration where an additional packet filter is placed before the ALG. The CARP PAP configuration is not part of this certification.

1.4.4 Physical Scope

Both ALG and PFL run on Intel compatible hardware in 64 bit mode (architecture x86_64). As the product genugate 10.0 Z is a combination of hardware and software, the hardware components are selected by

[0] See section 1.3.1 for legacy hardware

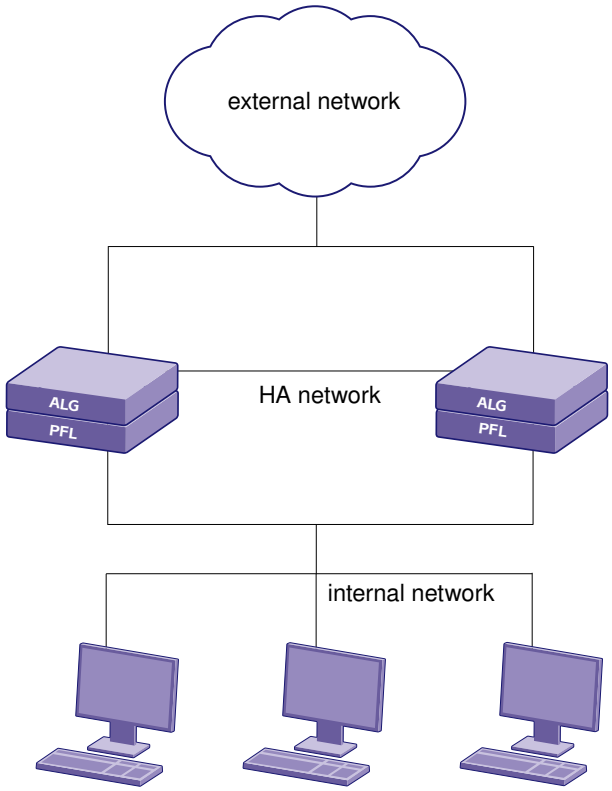


Figure 2: High availability setupIf the OSPF HA setup is used, an OSPF router is needed in the internal network. The admin network and the DMZ are not shown.

Table 1: Scope of delivery

Type	Name	Release	Date	Medium
Software	genugate	10.0 Z	XXX	install image
Software	genugate firewall	10.0	XXX	install image
Documentation	administrator and user guidance manual	10.0 Z	XXX	manual (german version)

genua. The end user has no need to check for compatibility. The scope of delivery can be seen in table 1. The TOE is located as software distributed on an installation image. The Image can either be shipped with the hardware, or can be downloaded from the genua support server. Both an USB install image or an ISO-image are available for installation. The documentation is contained in the install images but can also be downloaded from the genua support server. The TOE is contained in the install image for the product genugate 10.0 Z.

Application Note: While the evaluation was performed only with genugate hardware of revision 3.0 and 2.0, the software is expected to run with all security features also on older and newer hardware revisions, provided the hardware requirements of the preceding paragraph are met.

The physical connections are:

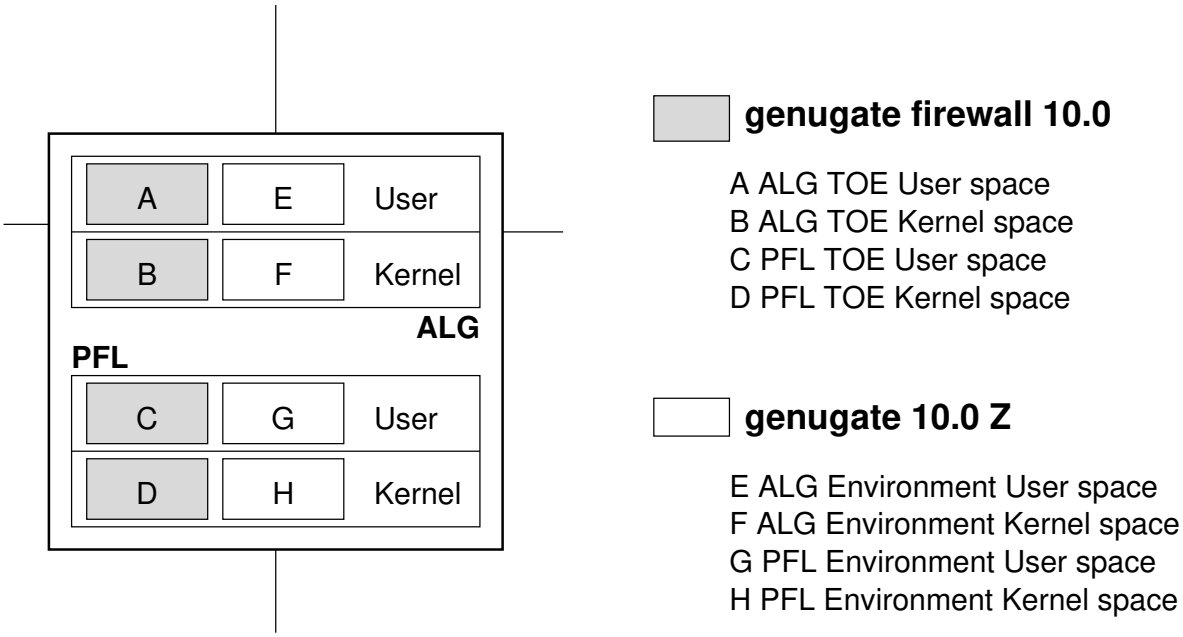


Figure 3: Scope and boundary

- the network interfaces to the external, internal, administration networks and the DMZ
- connections for the keyboard, monitor, and serial interfaces at the ALG and PFL
- power supply

The following list gives a schematic overview on the TOE and its environment. It divides the software on ALG and PFL into user and kernel space parts. On both systems, the user and the kernel space contain part of the TOE, and part of the environment. The following table lists the components in each part. The components for the parts **A**, **B**, **C** and **D** are part of the TOE. The components for **E**, **F**, **G**, and **H** are part of the environment.

- **A** ALG TOE User space
  relays, logging, admin web server, user web server, configuration commands, system startup
- **B** ALG TOE Kernel space
  network layer, logging, system call interface
- **C** PFL TOE User space
  logging, system startup
- **D** PFL TOE Kernel space
  network layer, logging, system call interface
- **E** ALG Environment User space `squid`, `postfix`, DNS server, `ntpd`, snmp server, CARP PAP configuration, genugate options: AWC, virus scanner; authentication methods, OS environment

- **F** ALG Environment Kernel space
  process management, memory management, device drivers, socket layer, tty driver, I/O system, IPC operation, file systems
- **G** PFL Environment User space `igmpproxy`, `ospfd`, `ospf6d`, OS environment
- **H** PFL Environment Kernel space
  process management, memory management, device drivers, socket layer, tty driver, I/O system, IPC operation, file systems

The different parts have the following interfaces with one another:

A	**B**	system call interface
A	**E**	interprocess communication (via system call interface)
B	**F**	kernel interfaces between the kernel components
C	**D**	system call interface
C	**G**	interprocess communication (via system call interface)
D	**H**	kernel interfaces between the kernel components
ALG	**PFL**	serial connection
ALG	**PFL**	network connection
ALG	**PFL**	USB boot medium

Depending on their roles, the users interact with the product in the following ways:

user: relay usage (sending and receiving IP packets to and from the TOE)

user: authentication dialogues for protocols that have authentication enabled

user: user web interface to change password

user: user web interface for the side channel authentication to activate IP addresses

administrator: admin web interface, REST web interface

administrator: interactive access at the shell level at the console

1.4.5 Logical Scope

The genugate is a complex product with many options that are not directly connected to the TSF. Some of these options do not interfere with the security functions of the TOE and are assumed to be configurable and usable in certificated mode. Examples are the DNS and NTP configuration, logging to external log servers or sending SNMP traps. For these options this security target does not contain SFRs.
On the other hand there are options that interfere with the security functions and therefore must not be used in certified mode.
The following paragraphs describe the features that are in scope of the TOE and for which SFRs exist. The list of excluded options consist of those options that interfere with the secure operation of the TOE. These options must not be configured.

The TOE has the following logical scope:

- The kernel components “network"’, “packet filter"’, and “restricted runtime"’ for ALG and PFL. This components perform the spoofing checks, packet filtering and access control for incoming data. The spoofing checks contain detecting any mismatch between the source and destination address of the IP packet and the IP address and netmask of the receiving interface.
- All relays that implement the policies and meta-policies referenced in chapter 1.4.1. These components perform the filtering on application level, ACL checks, and calls to the optional virus scanner. The virus scanning functionality is not part of the TOE. The ssh-, telnet- and ftprelay allow for user authentication. For the SMTP relay the authentication is optional. The authentication methods themselves are not part of the TOE. However, there is no security claim for the OPC UA functionality.
- The TCP and UDP relays can filter protocol conformance by applying regular expressions at the beginning of the communication. There are several predefined protocol conformance filter.
- system startup. This component performs the secure startup of the system and the conversion to maintenance mode.
- the logging and self-monitoring tools. These components perform the accounting and auditing functions.
- admin web interface including the REST web interface. This component allows the configuration by administrators.
- user web server. This component allows users to change their passwords.
- side channel web server. This component allows users to activate IP addresses through the side channel mechanism.
- The configuration for the users, network, relays, dns server, mail server, packet filter, HTTP-proxy squid, virus scanner, AWC, audit, snmp server, and igmpproxy.
- The options HA, Manpages.
- A patch installation mechanism, that only installs patches signed by genua. The mechanism also checks that the patch is for the approriate software version and patch level.

  The public key for the signature verification is contained in the installation image and part of the secure installation process. During operation it is secured by the self protection measures.

The TOE has the following logical boundaries:

- virus scanner interface: delivering the data to the virus scanner and obtaining the scanner result. The virus scanner itself is not part of the TOE.
- Advanced Web Categories (AWC): interface to a web site classification service. The classification service itself is not part of the TOE.
- Web Application Firewall (WAF): interface to a local web site protection service using the Modsecurity Project of the OWASP foundation.
- external authentication methods: interaction with the authentication service. The authentication methods themselves are not part of the TOE.

- configuration interface: sending forms to and receiving form data from a web browser.

This Security Target claims SFRs cryptographic operations only for the patch management. Excluded are especially the following operations:

- although some relays support encryption with TLS, this security target does not contain SFRs for the class FCS (Cryptographic Support). Therefore these cryptographic operations are not part of the TSF.
- the cryptographic operations of the SSH relay. This security target does not contain SFRs for the class FCS (Cryptographic Support). Therefore these cryptographic operations are not part of the TSF.
- the cryptographic operations of the IPsec in the HA network. This security target does not contain SFRs for the class FCS (Cryptographic Support). Therefore these cryptographic operations are not part of the TSF.

The TOE explicitely excludes the following options or services from its logical scope:

- the CARP balancing HA mode
- the Custom HA mode
- the CARP PAP mode
- the Application Filter
- remote administration of the PFL with SSH

1.5 Estimated End-Of Life of the Product

The genugate is usually re-certified every two years. In order to give the customers sufficient time to migrate to a new certified version, the support is usually extended to one year after the next certfied version is available. However, these are only general rules. In special cases, the product is supported by releasing patches up to the end of life of the certification, if that is necessary.

2 CC Conformance Claim

2.1 CC Conformance Claim

This Security Target is Part 2 extended and Part 3 extended to the Common Criteria Version 3.1 Revision 5 (April 2017) [3, 4].

2.2 PP Claim, Package Claim

There are no Protection Profile claims. This Security Target claims to be conformant to the Assurance Packet EAL4 augmented with ALC_FLR.2, ASE_TSS.2 and AVA_VAN.5. These components are defined in CC Part 3. The Security Target also claims ALC_PAM.1, which is an extended assurance component defined in this Security Target.

2.3 Conformance Rationale

The Security Target has no Protection Profile claim, therefore no conformance rationale has to be given. This Security Target uses extended functional component definitions (see sections 5.1–5.3). Therefore it is Part 2 extended. It uses extended assurance requirements (see section 5.4). Therefore it is Part 3 extended.

3 Security Problem Definition

In order to clarify the nature of the security problem that the TOE is intended to solve, this section describes the following:

- Any assumptions about the security aspects of the environment and/or of the manner in which the TOE is intended to be used.
- Any known or assumed threats to the assets against which specific protection within the TOE or its environment is required.
- Any organizational security policy statements or rules with which the TOE must comply.

3.1 Users

The Security Target defines the following users:

user Any person or software agent sending IP packets to or receiving from the TOE. The assumed attack potential is **high**. The general term user is used when it does not matter whether the user did authenticate at the TOE or not.

unauthenticated user Any person or software agent sending IP packets to or receiving from the TOE that did not authenticate at the TOE. The assumed attack potential is **high**. This term is used for users that did not (yet) authenticate at the TOE.

authenticated user Any person or software agent sending IP packets to or receiving from the TOE that authenticated at the TOE. The assumed attack potential is **high**.

administrator These are authenticated users that have the role of an administrator. This role authorises them to change the TOE configuration. Their assumed attack potential is undefined.

auditor These are authenticated users that have the role of an auditor. This is a restricted administrator role and authorises them to view the TOE configuration. Their assumed attack potential is undefined.

developer The developer is a trustworthy entity and provides patches to the product. It's assumed attack potential is undefined.

3.2 Assets

The Security Target defines the following assets:

resources in the connected networks The resources in the connected networks that the TOE is supposed to protect.

security sensitive data on the TOE The data on the TOE that contains security sensitive data.

3.3 Threats

The Security Target defines the following threats:

T.NOAUTH An unauthenticated user may attempt to bypass the security functions of the TOE and gain unauthenticated access to resources in other connected networks or read, modify or destroy security sensitive data on the TOE. The attack method is exploiting authentication protocol weaknesses.

T.SPOOF A user may attempt to send spoofed IP packets to the TOE in order to gain unauthorised access to resources in other connected networks. Without spoofing checks the TOE would route a response to the spoofed IP packet into a connected network that the user is not authorised to access.

T.MEDIAT A user may send non-permissible data through the TOE that result in gaining access to resources in other connected networks.

T.SELPRO A user may gain access to the TOE and read, modify or destroy security sensitive data on the TOE, by sending IP packets to the TOE and exploiting a weakness of the protocol used.

T.MISUSESSH A user may try to open a hidden (encrypted) channel by using SSH protocol messages like port forwardings in order to gain access to resources in other connected networks.

T.PATCH An administrator applies patches incorrectly. This might be a non-autorized patch, or actual patch and patch level do not conform. The system then runs in an inconsistent state and an attacker can possibly gain access resources in the connected networks.

3.4 Assumptions

The Security Target defines the following assumptions:

A.PHYSEC The TOE is physically secure. Only authorised persons have physical access to the TOE and the hardware including the PFL boot stick.

A.NOEVIL Administrators and auditors are non-hostile and follow all administrator and auditor guidance; however, they are capable of error. They use passwords that are not easily guessable.

A.ADMIN All administration is done only in the administration network during normal operation mode. The administration network and the attached workstation from which the administrators work are physically secure.

A.LOCAL Configuration using local files is only done by trained adminstrators that have a profound knowledge of OpenBSD and the installed tools. They know to estimate the security impact of the local files and only create local files that have no impact on the security functions of the TOE.

A.REST The automated clients that use the REST interface save the password in a secure way.

A.SINGEN Information can not flow among the internal, external, or DMZ, unless it passes through the TOE.

A.POLICY The security policy of the internal network allows only the administrators access to the network components and the network configuration.

A.TIMESTMP The environment provides reliable time stamps.

A.HANET The environment provides a physical separate network for TSF data transfer for the optional high availability setup.

A.USER The users use passwords that are not easily guessable and keep them secret.

A.TRUSTK The non-TOE parts of the kernel space are trustworthy and do not interfere with the security functions of the TOE.

A.TRUSTU The non-TOE parts of the user space are trustworthy and do not interfere with the security functions of the TOE.

A.LEGACY The legacy protocols TELNET and FTP (and SMTP if authentication is used) are used only in sufficiently secure environments.

A.REMOTE_AUTH The server for external authentication (RADIUS, LDAP) are located in secure networks.

A.OSPF The OSPF and OSPFv6 routers in the internal network are secured against attacks from the internal network.

Application Note: The Admin Web server is the preferred configuration interface. However the praxis shows that not all necessary configurations can be made by this interface. Therefore **A.LOCAL** is necessary.

3.5 Organisational Security Policies

The Security Target defines the following organisational security policies:

P.AUDIT All users must be accountable for their actions.

P.AVAIL A high availability operation must be possible where peers can take over the services of a failing system. (This policy only applies if needed.)

P.PASSWD The files imported for password file authentication must contain good passwords.

4 Security Objectives

The purpose of the security objectives is to describe the planned response to a security problem or threat. Threats can be directed against the TOE or the security environment. The CC identifies two categories of security objectives:

- security objectives for the TOE
- security objectives for the operating environment

4.1 Security Objectives for the TOE

The Security Target defines the following security objectives for the TOE:

O.IDAUTH The TOE must identify all network packets from the connected networks. It must check the IP addresses of the packet with the receiving interface to recognize IP-spoofing. It must identify all users before granting access to the security functions of the TOE. It must authenticate the users where an authentication is required.

O.MEDIAT The TOE must mediate the flow of all data between all connected networks.

O.SECSTA On start-up, the TOE must not compromise its resources or those of the connected networks.

O.SELPRO The TOE must have self-protection mechanisms that hinder attempts by users to bypass, deactivate or tamper with TOE security functions.

O.SECFUN The TOE must allow administrators to use the TOE security functions and must ensure that only authorised administrators have access to the functionality.

O.MISUSESSH The TOE must prevent SSH connections to set up SSH protocol messages that are not approved.

O.AUDREC The TOE must provide an audit trail of security-related events, and a means to present a readable and searchable view to authorised users.

O.ACCOUN The TOE must provide user accountability for data flows through the TOE and for the use of the security functions of administrators.

O.AVAIL The TOE must optionally provide a fail over solution where the services of a failing system are taken over by a peer machine.

O.PATCH The developer provides a patch mechanismus for product updates that is integrity protected and signed with a developer key. The signature of the patch is automatically checked during installation. Then the patch is applied in a secure and correct way. Activation of the patch and update of the identification data shall be performed at the same time. The TOE shall always be in a defined state during the update. Each patch level is uniquely identified. The patch mechanism verifies the version and patch level information contained in the patch and only applies the patch if the patch is for the current software version and patch level.

4.2 Security Objectives for the Environment

The Security Target defines the following security objectives for the environment:

OE.PHYSEC Those responsible for the TOE must assure that the TOE is placed at a secured place where only authorised people have access.

OE.NOEVIL Those responsible for the TOE must assure that all administrators and auditors are competent, regularly trained and execute the administration in a responsible way.

OE.ADMIN Those responsible for the TOE must assure that administration is only done in the physically secured administration network during normal operation mode.

OE.LOCAL Those responsible for the TOE must assure that configuration by local files is only done by trained adminstrators that know how to esitmate the security impact of the local files.

OE.REST The automated clients that use the REST interface save the password in a secure way.

OE.SINGEN Those responsible for the TOE must assure that the TOE is the only connection between the different networks.

OE.POLICY Those responsible for the TOE must assure that the security policy for the internal network allows only administrators access to the network components and the network configuration. They must assure that the policy is maintained.

OE.TIMESTMP The IT-environment must supply reliable time stamps for the TOE.

OE.RTCLOCK The IT-environment must supply a real-time clock.

OE.HANET The IT-environment must supply a physical network for transfer of TSF data between nodes for the optional high availability setup.

OE.USER Those responsible for the TOE must assure that the users follow the user guidance, especially that they choose not easily guessable passwords and that they keep them secret.

OE.TRUSTK The IT-environment must assure that the non-TOE parts of the kernel space do not interfere with the security functions of the TOE.

OE.TRUSTU The IT-environment must assure that the non-TOE parts of the user space do not interfere with the security functions of the TOE.

OE.LEGACY The IT-environment must provide a sufficiently secure environment for the legacy TELNET and FTP protocols (and SMTP if authentication is used).

OE.REMOTE_AUTH The IT-environment must assure that the server for external authentication (RADIUS, LDAP) are located in secure networks.

OE.OSPF The IT-environment must provide OSPF and OSPFv6 routers that are secured against attacks from the internal network.

OE.PASSWD The files imported for password file authentication contain good passwords.

4.3 Security Objectives Rationale

This chapter contains the ST security objectives rationale. It must show that the security objectives are consistent. Table 2 shows an overview of the rationale.

Table 2: Security Objectives Rationale

	O.IDAUTH	O.MEDIAT	O.SECSTA	O.SELPRO	O.SECFUN	O.MISUSESSH	O.AUDREC	O.ACCOUN	O.AVAIL	O.PATCH	OE.PHYSEC	OE.NOEVIL	OE.ADMIN	OE.LOCAL	OE.REST	OE.SINGEN	OE.POLICY	OE.TIMESTMP	OE.RTCLOCK	OE.HANET	OE.USER	OE.TRUSTK	OE.TRUSTU	OE.LEGACY	OE.REMOTE_AUTH	OE.OSPF	OE.PASSWD
T.NOAUTH	X		X		X																						
T.SPOOF	X																										
T.MEDIAT		X																									
T.SELPRO	X		X	X	X																						
T.MISUSESSH						X																					
T.PATCH										X																	
A.PHYSEC											X																
A.NOEVIL												X															
A.ADMIN													X														
A.LOCAL														X													
A.REST															X												
A.SINGEN																X											
A.POLICY																	X										
A.TIMESTMP																		X	X								
A.HANET																				X							
A.USER																					X						
A.TRUSTK																						X					
A.TRUSTU																							X				
A.LEGACY																								X			
A.REMOTE_AUTH																									X		
A.OSPF																										X	
P.AUDIT							X	X																			
P.AVAIL									X																		
P.PASSWD																											X

4.3.1 Threat Rationale

The following lists show that all threats are countert by objectives.

T.NOAUTH: This threat is met by the following objectives:

O.IDAUTH: The objective **O.IDAUTH** guarantees that all interactions with the TOE are identified. Only authenticated users can use functions that need authorisation.

O.SECSTA: The objective **O.SECSTA** assures that the threat is also met at start up.

O.SECFUN: The objective **O.SECFUN** guarantees that only authorised administrators can change the configuration of the TOE.

T.SPOOF: This threat is met by the following objectives:

O.IDAUTH: The objective **O.IDAUTH** makes sure that the identification of the IP addresses of every received packet recognises IP spoofing attacks. The objective requires checking the IP address and netmask of the receiving interface, and the source and destination IP address of the packet. The check has to recognize IP spoofing attacks, i.e. the IP packet was not expected at that interface.

T.MEDIAT: This threat is met by the following objectives:

O.MEDIAT: The objective **O.MEDIAT** (mediation of all network data) prevents that non-permissible data is sent across the TOE.

T.SELPRO: This threat is met by the following objectives:

O.SELPRO: The self protection objective **O.SELPRO** prevents reading, modifying or destroying security sensitive data on the TOE.

O.SECSTA: The objective **O.SECSTA** assures that the threat is also met at start-up.

O.IDAUTH: **O.IDAUTH** guarantees that only authorised administrators can read, modify, or destroy security sensitive data on the TOE.

O.SECFUN: **O.SECFUN** guarantees that only authorised administrators can read, modify, or destroy security sensitive data on the TOE.

T.MISUSESSH: This threat is met by the following objectives:

O.MISUSESSH: The objective **O.MISUSESSH** prevents misuse of SSH connections.

T.PATCH: This threat is met by the following objectives:

O.PATCH: The signature check during patch installation guarantees that the patch is autorized by the developer and has the correct patch level.

4.3.2 Assumption Rationale

The following lists show that all assumptions are met by objectives.

A.PHYSEC: This assumption is met by the following objective:

OE.PHYSEC: This objective assures that the assumption about a physically secure TOE can be made.

A.NOEVIL: This assumption is met by the following objective:

OE.NOEVIL: This objective assures that the administrators and auditors are trained and therefore that they are no threat to the TOE.

A.ADMIN: This assumption is met by the following objective:

OE.ADMIN: This objective assures that the administration only occurs in a distinct physically secured network, only used for administration during normal operation mode.

A.LOCAL: This assumption is met by the following objective:

OE.LOCAL: The objective requires that only trained adminstrators configure the TOE by local files.

A.REST: This assumption is met by the following objective:

OE.REST: The objective assures that the passwords for the automated access of the REST interface are secured.

A.SINGEN: This assumption is met by the following objective:

OE.SINGEN: This objective assures that the TOE can not be bypassed and therefore assures that the assumption is met.

A.POLICY: This assumption is met by the following objective:

OE.POLICY: This objective assures that an assumption about the security policy can be made.

A.TIMESTMP: This assumption is met by the following objectives:

OE.TIMESTMP: This objective provides reliable time stamps from the environment.
OE.RTCLOCK: This objective ensuresthat the environment uses a real-time clock to provide time stamps.

A.HANET: This assumption is met by the following objective:

OE.HANET: This objective provides the extra network to transfer TSF data between nodes in the optional HA setup.

A.USER: This assumption is met by the following objective:

OE.USER: This objective assures that the users use appropriate passwords and keep them secret.

A.TRUSTK: This assumption is met by the following objective:

OE.TRUSTK: This objective assures that the non-TOE parts of the kernel space are trustworthy.

A.TRUSTU: This assumption is met by the following objective:

OE.TRUSTU: This objective assures that the non-TOE parts of the user space are trustworthy.

A.LEGACY: This assumption is met by the following objective:

OE.LEGACY: This objective assures that the legacy protocols are used only in sufficiently secure environments.

A.REMOTE_AUTH: This assumption is met by the following objective:

OE.REMOTE_AUTH: This objective assures that the external authentication servers are located in secure networks.

A.OSPF: This assumption is met by the following objective:

OE.OSPF: This objective assures that the OSPF and OSPFv6 routers are secured against attacks from the internal network.

4.3.3 Policy Rationale

The following lists show that all policies are met by objectives.

P.AUDIT: The policy is met by the following objectives:

O.ACCOUN: The objective **O.ACCOUN** (accounting of all user interactions and all security related events), makes sure that all audit trails are written.

O.AUDREC: The (possible) loss of audit data is recognised by **O.AUDREC**.

P.AVAIL: The policy is met by the following objective:

O.AVAIL The objective **O.AVAIL** provides the optional high availability policy request.

P.PASSWD: The policy is met by the following objective:

OE.PASSWD: The objective **OE.PASSWD** provides the password quality needed by **P.PASSWD**.

5 Extended Components Definition

5.1 Class FAU: Security audit

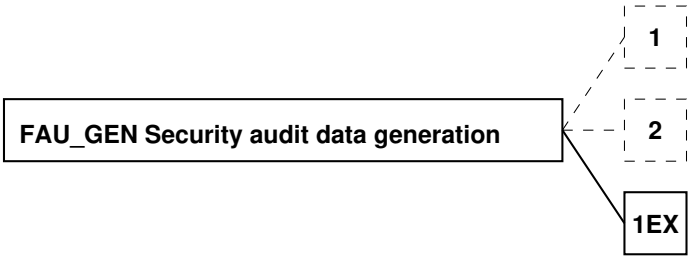
5.1.1 Security audit data generation (FAU_GEN)

The family has been enhanced by one component FAU_GEN.1EX. It is intended to be a replacement for FAU_GEN.1 when the security function does not support audit generation for startup and shutdown of the audit functions. This component can be used as a replacement for the dependencies on FAU_GEN.1, because all other audit events can be specified as in FAU_GEN.1.

Family behaviour

This family defines requirements for recording the occurrence of security relevant events that take place under TSF control. This family identifies the level of auditing, enumerates the types of events that shall be auditable by the TSF, and identifies the minimum set of audit-related information that should be provided within various audit record types.

Component levelling



The components FAU_GEN.1 and FAU_GEN.2 are already described in [1]. Only FAU_GEN.1EX is new and described here.

Management: FAU_GEN.1EX

There are no management activities foreseen.

Audit: FAU_GEN.1EX

There are no actions identified that should be auditable if FAU_GEN Security audit data generation is included in the PP/ST.

FAU_GEN.1EX Runtime audit data generation

Hierarchical to: No other components.

Dependencies: FPT_STM.1 Reliable time stamps.

FAU_GEN.1EX.1 The TSF shall be able to generate an audit record of the following auditable events:

a) All auditable events for the [selection: choose one of: *minimum, basic, detailed, not specified*] level of audit; and

b) [assignment: *other specifically defined auditable events*].

FAU_GEN.1EX.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, [assignment: *other audit relevant information*].**

5.2 Class FIA: Identification and authentication

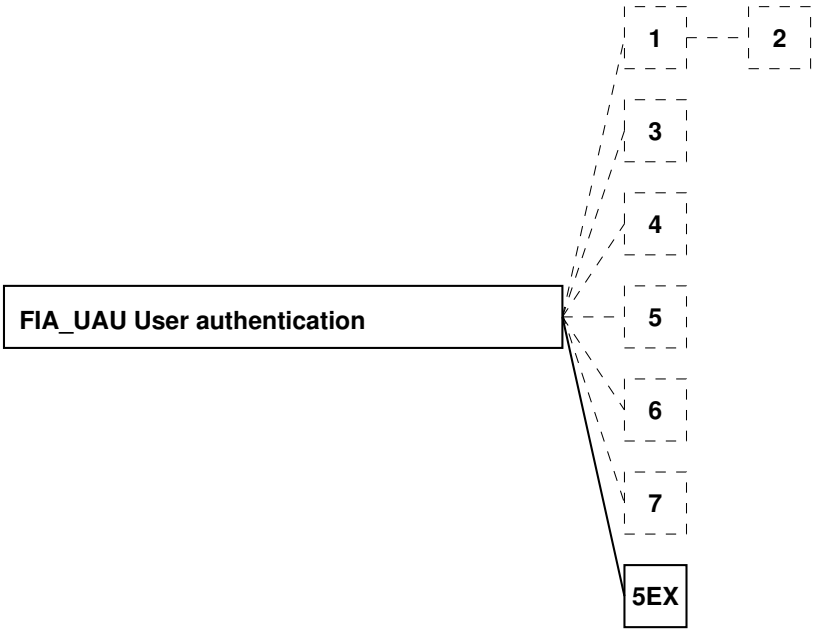
5.2.1 User authentication (FIA_UAU)

The family has been enhanced by one component FIA_UAU.5EX. It is thought as a replacement for FIA_UAU.5 when the proper authentication is done by an external means. This component can also be used as a replacement for the dependencies on FIA_UAU.5, because it requires the same functionality.

Family behaviour

This family defines the types of user authentication mechanisms supported by the TSF. This family also defines the required attributes on which the user authentication mechanisms must be based.

Component levelling



The components FIA_UAU.1, FIA_UAU.2, FIA_UAU.3, FIA_UAU.4, FIA_UAU.5, FIA_UAU.6 and FIA_UAU.7 are already described in CC Part2. Only FIA_UAU.5EX will be described in this chapter.

Management: FIA_UAU.5EX

The following actions could be considered for the management functions in FMT:

a)
the management of authentication mechanisms;

b)
the management of the rules for authentication.

Audit: FIA_UAU.5EX

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

a)
Minimal: The final decision on authentication;

b)
Basic: The result of each activated mechanism together with the final decision.

FIA_UAU.5EX External authentication mechanisms

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UAU.5EX.1 The TSF shall provide [assignment: *list of multiple authentication mechanisms*] to support user authentication by external means.

FIA_UAU.5EX.2 The TSF shall authenticate any user's claimed identity according to the [assignment: *rules describing how the multiple authentication mechanisms provide authentication*].

5.3 Class FPT: Protection of the TSF

The class has been augmented by two families.

5.3.1 Simple Self Test (FPT_SST)

Family behaviour

The family defines the requirements for the self-testing of the TOE with respect to some expected correct operation. Examples are expected running processes or expected files at some location in the file system. These tests can be carried out at start-up, periodically, at the request of the authorised user, or when other conditions are met. The actions to be taken by the TOE as the result of self testing are defined in other families.
The requirements of this family are also needed to detect the corruption of TOE executable code (i.e. TOE software) and TOE data by various failures that do not necessarily stop the TOE's operation (which would be handled by other families). These checks must be performed because these failures may not necessarily be prevented. Such failures can occur either because of unforeseen failure modes or associated oversights in the design of hardware, firmware, or software, or because of malicious corruption of the TOE due to inadequate logical and/or physical protection.

Component levelling

FPT_SST.1EX TOE testing, provides the ability to test the TOE's correct operation. These tests may be performed at start-up, periodically, at the request of the authorised user, or when other conditions are met. It also provides the ability to verify the integrity of TOE data and executable code.

Management: FPT_SST.1EX

The following actions could be considered for the management functions in FMT:

a) management of the conditions under which TOE self testing occurs, such as during initial start-up, regular interval, or under specified conditions;

b) management of the time interval if appropriate.

Audit: FPT_SST.1EX

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

a) Basic: Execution of the TOE self tests and the results of the tests.

FPT_SST.1EX TOE testing

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_SST.1EX.1 The TSF shall run a suite of self tests [selection: *during initial start-up*, *periodically during normal operation*, *at the request of the authorised user*, *at the conditions* [assignment: *conditions under which self test should occur*]] to perform the following checks: [assignment: *list of self tests*].

FPT_SST.1EX.2 The TSF shall provide authorised users with the capability to query the results of the following checks: [assignment: *list of self tests*].

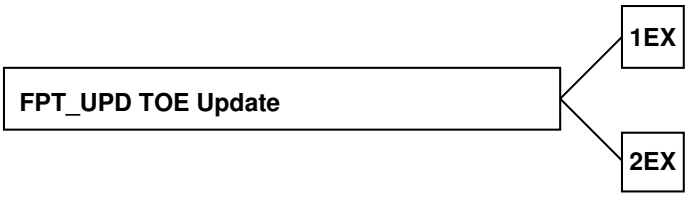
5.3.2 TOE Update (FPT_UPD)

The family specifies the secure and correct installation of patches from the developer by an administrator. The new family is added to the class FPT, because it protects the TSF from manipulation through the installation of malicious patches.

Family behaviour

The requirements of this family assure that only autorized pataches from the developer can be installed in a secure and correct way by an administrator. The family has two components, which are independent from each other.

Component levelling



FPT_UPD.1EX Trusted update, requires that patches are signed using the specified cryptographic standards.
FPT_UPD.2EX Update identification data, requires that the patches have a unique patch level that is updated at the same time.

Management: FPT_UPD.1EX.1, FPT_UPD.1EX.2

The following actions could be considered for the management functions in FMT:

a) determining the time when to apply the patches.

Audit: FPT_UPD.1EX, FPT_UPD.2EX

The following actions should be auditable if FAU_UPD.1EX TOE Update is included in the PP/ST:

a) Basic: The result of the patch update.

FPT_UPD.1EX Trusted update

Hierarchical to: No other components.

Dependencies: FCS_COP.1 Cryptographic operation.

FPT_UPD.1EX.1 The TOE shall cryptographically verify additional code/patches to itself using a digital signature prior to installation using schemes specified in [assignment: *FCS_COP.1 SFR*].

FPT_UPD.1EX.2 A modification of the TOE shall only be allowed if the software update

- **is intended for the current software version,**
- **has the correct patch level and**
- **has been cryptographically verified with regard to integrity and authenticity.**

FPT_UPD.2EX Update identification data

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_UPD.2EX.1 The TSF shall verify if the activation of the patch and the update of the identification data have been both completed.

FPT_UPD.2EX.2 The TSF shall update the active identification data when the patch is applied in order to keep the system in a defined state.

FPT_UPD.2EX.3 The TSF shall use the maintenance mode to activate the final TOE.

5.4 Class ALC: Life-cycle support

The class has been augmented by one family that specifies the secure and correct generation of patches by the developer. The new family is added to the class ALC, because it handles the patch aspect of the product life-cycle that has not been considered by the CC.

5.4.1 Patch Management (ALC_PAM)

Objectives

The objective of this family is to identify procedures to be implemented in the development process, which will be applied after the initial release of a TOE.
The application of these patch management processes cannot be always determined at the time of the base evaluation, but at least, it is possible to evaluate the policies and procedures that a developer has in place to perform management processes in the future, and obtain some evidence of the correct application of the procedures during the patching of the problems found during the evaluation of other assurance classes like AVA and ATE.
These procedures shall include instructions on how to securely sign, distribute and apply patches and how the life cycle of the keys used for providing authenticity of new patches is handled.

Component levelling

This family contains only one component.

Application notes

None.

ALC_PAM.1 Patch Management Processes

Dependencies: ALC_FLR.2 Flaw reporting procedures.

Developer action elements:

ALC_PAM.1.1D The developer shall provide a Patch Management Policy.

ALC_PAM.1.2D The developer shall self-assess and confirm the application of existing policies on a regular basis saving records of its application.

ALC_PAM.1.3D The developer shall provide security patches using the defined policies and procedures at least until the estimated end-of-life of the TOE.

Content and presentation elements:

ALC_PAM.1.1C The developer's patch management policies shall describe what is the criteria used for the decision that a patch has to be released.

ALC_PAM.1.2C The Security Target shall contain the estimated end-of-life of the TOE.

ALC_PAM.1.3C The developer's patch management policies shall describe how to self-assess the security relevance of a patch (i.e. Security Impact Analysis Report, S-IAR) and which procedures have to apply due to which assessment result.

ALC_PAM.1.4C The developer's patch management policies shall describe how to update the evidence documentation used in the base evaluation.

ALC_PAM.1.5C The developer's patch management policies shall describe how unhandled (potential) flaws are documented.

ALC_PAM.1.6C The developer's patch management policies shall describe which organizational role (or group) is responsible for the patch development.

ALC_PAM.1.7C The developer's patch management policies shall describe which policies have to be applied until the end of life of the TOE during the patch management.

ALC_PAM.1.8C Each tool used for the patch management shall be documented.

ALC_PAM.1.9C The patch management policies shall describe the mandatory structure and content of the S-IAR.

ALC_PAM.1.10C Each type of documentation used to record decisions in the patch management process shall be documented.

ALC_PAM.1.11C The patch management policies shall describe the mandatory content of patch release notes.

ALC_PAM.1.12C The patch management policies shall describe the mandatory content for the guidance documents which have to be fulfilled to support the installation of the patch.

ALC_PAM.1.13C The patch management policies shall describe the mandatory procedures during patch release.

ALC_PAM.1.14C The patch management policies shall contain rules in which case the evaluation facility has to perform additional tests before the patch is released.

ALC_PAM.1.15C The patch management policies shall describe how each of the patch management Security Objectives for the Operational Environment are fulfilled until the end of life of the TOE.

Evaluator action elements:

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

6 Security Requirements

This section contains the security functional requirements, the security assurance requirements, and the rationale.

6.1 Security Functional Requirements

Most of the security functional requirements in this subsection have been drawn from the CC Part 2.
The functional requirements for FAU_GEN.1EX, FIA_UAU.5EX, FPT_SST.1EX, FPT_UPD.1EX and FPT_UPD.2EX are not drawn from CC Part 2.
In the following, the unmodified text from the functional requirement templates is displayed in a sans serif font. The operation ***assignment*** is set in a bold italic serif font. The operation *selection* is set in an italic serif font The operation **refinement** is set in a bold font. The iterations are done by repeating the requirements and adding a colon and a sequence number. In a few occasions, the text has been modified slightly as an refinement operation.

6.1.1 Class FAU: Security audit

6.1.1.1 Security audit automatic response (FAU_ARP)

FAU_ARP.1 Security alarms

FAU_ARP.1.1 The TSF shall take ***configurable actions (log, digest, wall, exec, mail, down, halt)*** upon detection of a potential security violation.

6.1.1.2 Security audit data generation (FAU_GEN)

FAU_GEN.1EX Runtime audit data generation

FAU_GEN.1EX.1 The TSF shall be able to generate an audit record of the following auditable events:

a) All auditable events for the *not specified* level of audit; and
b) ***Starting and stopping of the system, changing operation modes, relay configuration, loading of packet filter rules, relay usage, administration, authentication.***

FAU_GEN.1EX.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, ***unspecified log data***.

6.1.1.3 Security audit analysis (FAU_SAA)

FAU_SAA.1 Potential violation analysis

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

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

a) Accumulation or combination of ***configurable events (packet filter violations, selected messages of daemons, selected messages of the relays, ARP spoofing messages, time synchronization errors, usage of duplicate IP addresses, selected kernel messages and messages from the processes that implement the self-tests)*** known to indicate a potential security violation;

b) ***none***.

6.1.1.4 Security audit review (FAU_SAR)

FAU_SAR.1 Audit review

FAU_SAR.1.1 The TSF shall provide ***administrators and auditors*** with the capability to read ***all audit information*** from the audit records.

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

FAU_SAR.2 Restricted audit review

FAU_SAR.2.1 The TSF shall prohibit all users read access to the audit records, except those users that have been granted explicit read-access.

FAU_SAR.3 Selectable audit review

FAU_SAR.3.1 The TSF shall provide the ability to apply ***searches*** of audit data based on ***time, date, process id, additional log data (for relay audit data: relay type, connection state, IP addresses and ports, status of logged event, bytes transferred)***.

6.1.1.5 Security audit event storage (FAU_STG)

FAU_STG.1:1 Protected audit trail storage

FAU_STG.1.1:1 The TSF shall protect the stored **automatically rotated** audit records in the audit trail from unauthorised deletion.

FAU_STG.1.2:1 The TSF shall be able to *prevent* unauthorised modifications to the **automatically rotated** audit records in the audit trail.

Application Note: Automatically rotated audit records are rotated on a regular bases.

FAU_STG.1:2 Protected audit trail storage

FAU_STG.1.1:2 The TSF shall protect the stored **flagged** audit records in the audit trail from unauthorised deletion.

FAU_STG.1.2:2 The TSF shall be able to *prevent* unauthorised modifications to the **flagged** audit records in the audit trail.

Application Note: Flagged audit records are rotated with the acknowledgement of the administrator during maintenance mode.

FAU_STG.4:1 Prevention of audit data loss

FAU_STG.4.1:1 The TSF shall *prevent audited events, except those taken by the authorised user with special rights* and ***execute a configurable action (default: inform the administrators)*** if the **application level** audit trail is full.

Application Note: This SFR applies if the audit trail is flooded with messages so that the storage fills even with log file rotation.

FAU_STG.4:2 Prevention of audit data loss

FAU_STG.4.1:2 The TSF shall *overwrite the oldest stored audit records* and ***execute a configurable action (default: generate a process master event)*** if the **kernel** audit trail is full.

Application Note: The process master actions range from ignoring the event to halting the system.
Application Note: The kernel also generates a process master event if a configurable audit trail threshold is reached, so that the administrator can take preventive measures.

6.1.2 Class FCS: Cryptographic support

6.1.2.1 Cryptographic operation (FCS_COP)

FCS_COP.1 Cryptographic operation

FCS_COP.1.1 The TSF shall perform ***signature verification for the integrity check of update packages*** in accordance with a specified cryptographic algorithm ***RSA signature verification*** and cryptographic key sizes ***4096 bit*** that meet the following: ***RSA signatures according to PKCS#1, v2.1 using RSASSA-PKCS1-v1_5 and SHA-512 (default), SHA-384 or SHA-256***.

6.1.3 Class FDP: User data protection

6.1.3.1 Information flow control policy (FDP_IFC)

FDP_IFC.1:1 Subset information flow control

FDP_IFC.1.1:1 The TSF shall enforce the ***unauthenticated user SFP*** on

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

b) information: traffic sent through the TOE from one subject to another;
c) operation: pass information.

FDP_IFC.1:2 Subset information flow control

FDP_IFC.1.1:2 The TSF shall enforce the ***authenticated user SFP*** on

a) subjects: users that send and receive SMTP, SSH, FTP, or TELNET information through the TOE to one another, only after the user initiating the information flow has authenticated at the TOE through the SMTP, SSH, FTP, or TELNET authentication mechanism;
b) information: SMTP, SSH, FTP, or TELNET traffic sent through the TOE from one subject to another;
c) operation: pass information.

Application Note: This IFC only applies if the authentication method has been activated for the respective protocol.
Application Note: The http-, imap-, pop-, sip-, and smtp2smtprelay do not allow authentication at the TOE even if the respective protocols support authentication.

FDP_IFC.1:3 Subset information flow control

FDP_IFC.1.1:3 The TSF shall enforce the ***identified side channel user SFP*** on

a) subjects: users that send and receive information through the TOE to one another, only after identifying the user by IP address;
b) information: traffic sent through the TOE from one subject to another;
c) operation: pass information.

FDP_IFC.1:4 Subset information flow control

FDP_IFC.1.1:4 The TSF shall enforce the ***authenticated gui user SFP*** on

a) subjects: users that send and receive information to /from the TOE;
b) information: html form data for side channel authentication and user password changes;
c) operation: pass information.

FDP_IFC.1:5 Subset information flow control

FDP_IFC.1.1:5 The TSF shall enforce the ***authenticated administrator SFP*** on

a) subjects: administrators from the administration network that send and receive information to/from the TOE;
b) information: html form data for administration;
c) operation: pass information.

Application Note: All SFRs in this section have been refined by using (external) users instead of (internal) subjects for item a).

6.1.3.2 Information flow control functions (FDP_IFF)

FDP_IFF.1:1 Simple security attributes

FDP_IFF.1.1:1 The TSF shall enforce the ***unauthenticated user SFP*** based on the following types of subject and information security attributes:

- ***The header information of network packets, depending on their type:***
  - ***a) TCP: IP and TCP header;***
  - ***b) UDP: IP and UDP header;***
  - ***c) ICMP: IP header and ICMP message;***
  - ***d) IGMP: IP header and IGMP message;***
  - ***e) IP: IP header;***
- ***The actual date and time.***
- ***The incoming and outgoing interfaces.***
- ***Additional information depending on the handling relay:***
  - ***a) iprelay: none;***
  - ***b) pingrelay: none;***
  - ***c) udprelay: if the protocol conformance filter is active: protocol and/or application data;***
  - ***d) tcprelay: if the protocol conformance filter is active: protocol and/or application data;***
  - ***e) smtprelay: protocol and application data;***
  - ***f) smtp2smtprelay: protocol and application data;***
  - ***g) imaprelay: protocol and application data;***
  - ***h) poprelay: protocol and application data;***
  - ***i) siprelay: protocol and application data;***
  - ***j) sshrelay: protocol data;***
  - ***k) wwwrelay: protocol and application data;***
  - ***l) httprelay: protocol and application data;***
  - ***m) ftprelay: protocol data;***
  - ***n) telnetrelay: protocol data;***
  - ***o) mcastudprelay: IGMP and multicast UDP packets;***

Application Note: The list only considers the relays, but not the meta-policies because they are convenience methods to easily configure services.

FDP_IFF.1.2:1 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold:

- ***IP spoofing check pass.***
- ***IP option check pass.***
- ***The connection is configured:***
  - ***a) iprelay: source and destination IP address and protocol are allowed;***
  - ***b) pingrelay: source and destination IP address are allowed;***
  - ***c) udprelay: source and destination IP address and port are allowed;***
  - ***d) tcprelay: source and destination IP address and port are allowed;***

e) mcastudprelay: packets of the respective multicast group are allowed;
f) all other relays: source and destination IP address and port are allowed.

- ***The ALG packet filter rules pass.***
- ***All ACL checks for the respective relay pass.***
- ***For packets that have a source or destination address from the internal network: The PFL packet filter rules pass.***

FDP_IFF.1.3:1 The TSF shall enforce the ***none***.

FDP_IFF.1.4:1 The TSF shall explicitly authorise an information flow based on the following rules: ***none***.

FDP_IFF.1.5:1 The TSF shall explicitly deny an information flow based on the following rules: ***The protocol data is filtered:***

a) iprelay: none.
b) pingrelay: none.
c) udprelay: none.
d) tcprelay: none.
e) smtprelay: configured checks for mail sender and recipient, greylisting, mail relay lead to the rejection of mail.
E-mail contents of content-type text/html can be filtered for active contents, if configured. A virus scanner can check the application data. MIME-encoded e-mails are (recursively) parsed their parts checked like non encoded e-mails.
f) smtp2smtprelay: E-mail contents of content-type text/html can be filtered for active contents, if configured. A virus scanner can check the application data. MIME-encoded e-mails are (recursively) parsed their parts checked like non encoded e-mails.
g) imaprelay: the ACL and request method checks fail. A virus scanner can check the application data.
h) poprelay: configurable protocol elements from the client are discarded.
Application data of content-type text/html can be filtered for active contents, if configured. A virus scanner can check the application data. MIME-encoded messages are (recursively) parsed their parts checked like non encoded messages.
i) siprelay: the tests for the configured internal and external domains and RTP port ranges fail. The ACL and request method checks fail.
j) sshrelay: a subset of SSH protocol messages can be filtered out of the connection.
k) wwwrelay: configurable protocol elements from the client or server are discarded; configurable cookies are filtered. The application data is filtered.
Server replies of content-type text/html can be filtered for active contents, if configured. A virus scanner can check the application data. MIME-encoded replies are (recursively) parsed their parts checked like non en-

coded contents.

l) httprelay: the ACL and protocol (HTTP/Websockets) checks fail. The XML validation of the application data fails when using the webservice policy.

m) ftprelay: configurable protocol elements from the client are discarded.

n) telnetrelay: none

o) mcastudprelay: none

p) all relays: An authenticated administrator can explicitly terminates an existing connection. An authenticated administrator can add IP addresses to a list of blocked IP addresses.

Application Note: The list only considers the relays, but not the meta-policies because they are convenience methods to easily configure services. However, the httprelay cannot be configured itself, so in that case the meta-policies have to be considered.

FDP_IFF.1:2 Simple security attributes

FDP_IFF.1.1:2 The TSF shall enforce the ***authenticated user SFP*** based on the following types of subject and information security attributes:

- ***The header information of network packets, depending on their type:***
  - ***a) TCP: IP and TCP header. The actual date and time. The interfaces from which the packets are received and to which they are delivered.***
- ***Additional information depending on the configurable handling relay:***
  - ***a) smtprelay: protocol data;***
  - ***b) sshrelay: protocol data.***
  - ***c) ftprelay: protocol data;***
  - ***d) telnetrelay: protocol data;***

FDP_IFF.1.2:2 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold:

- ***IP spoofing check pass.***
- ***IP option check pass.***
- ***The connection is configured:***
  - ***Source and destination IP and port are allowed.***
  - ***The ALG packet filter rules pass.***
  - ***All ACL checks for the relay pass.***
- ***The user can be authenticated by the authentication data.***
- ***For packets that have a source or destination address from the internal network: The PFL packet filter rules pass***

FDP_IFF.1.3:2 The TSF shall enforce the ***none***.

FDP_IFF.1.4:2 The TSF shall explicitly authorise an information flow based on the following rules: ***none***.

FDP_IFF.1.5:2 The TSF shall explicitly deny an information flow based on the following rules: ***The protocol data is filtered:***

a) smtprelay: none;
b) sshrelay: none.
c) ftprelay: configurable protocol elements from the client are discarded.
d) telnetrelay: none;

FDP_IFF.1:3 Simple security attributes

FDP_IFF.1.1:3 The TSF shall enforce the ***identified side channel user SFP*** based on the following types of subject and information security attributes: ***The header information of network packets, depending on their type:***

- ***IP and TCP header.***
- ***The actual date and time.***
- ***The interfaces from which the packets are received and to which they are delivered.***

FDP_IFF.1.2:3 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold:

- ***IP spoofing check pass.***
- ***IP option check pass.***
- ***The connection is configured:***
  - ***tcprelay: source and destination IP and port are allowed.***
  - ***The ALG packet filter rules pass.***
  - ***All ACL checks for the respective relay pass.***
- ***For packets that have a source or destination address from the internal network: The PFL packet filter rules pass.***
- ***The sender IP has been registered as a side channel IP address by an authenticated side channel user.***

FDP_IFF.1.3:3 The TSF shall enforce the ***none***.

FDP_IFF.1.4:3 The TSF shall explicitly authorise an information flow based on the following rules: ***none***.

FDP_IFF.1.5:3 The TSF shall explicitly deny an information flow based on the following rules: ***timeout: no data is transported on this connection for a configurable time (default 10 minutes).***

FDP_IFF.1:4 Simple security attributes

FDP_IFF.1.1:4 The TSF shall enforce the ***authenticated gui user SFP*** based on the following types of subject and information security attributes: ***The header information of network packets, depending on their type:***

- ***TCP: IP and TCP header.***
- ***The actual date and time.***
- ***The interfaces from which the packets are received and to which they are delivered.***
- ***The authentication data (cookie).***

FDP_IFF.1.2:4 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold:

- ***IP spoofing check pass.***
- ***IP option check pass.***
- ***The connection is configured:***
  - ***tcprelay: source and destination IP and port are allowed.***
  - ***The ALG packet filter rules pass.***
  - ***All ACL checks for the respective relay pass.***
- ***For packets that have a source or destination address from the internal network: The PFL packet filter rules pass.***
- ***The authentication data (cookie) is accepted as a valid.***

FDP_IFF.1.3:4 The TSF shall enforce the ***none***.

FDP_IFF.1.4:4 The TSF shall explicitly authorise an information flow based on the following rules: ***none***.

FDP_IFF.1.5:4 The TSF shall explicitly deny an information flow based on the following rules: ***timeout: no data is transported on this connection for a configurable time (default 10 minutes).***

FDP_IFF.1:5 Simple security attributes

FDP_IFF.1.1:5 The TSF shall enforce the ***authenticated administrator SFP*** based on the following types of subject and information security attributes: ***The header information of network packets, depending on their type:***

- ***TCP: IP and TCP header.***
- ***The actual date and time.***
- ***The interfaces from which the packets are received and to which they are delivered.***
- ***The authentication data (cookie).***

FDP_IFF.1.2:5 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold:

- ***IP spoofing check pass.***
- ***IP option check pass.***
- ***The connection is configured:***
  - ***tcprelay: source and destination IP and port are allowed.***
  - ***The ALG packet filter rules pass.***
  - ***All ACL checks for the respective relay pass.***
- ***For packets that have a source or destination address from the internal network: The PFL packet filter rules pass.***
- ***The request comes from the administration network***
- ***The authentication data (cookie) is accepted as a valid.***

FDP_IFF.1.3:5 The TSF shall enforce the ***none***.

FDP_IFF.1.4:5 The TSF shall explicitly authorise an information flow based on the following rules: ***none***.

FDP_IFF.1.5:5 The TSF shall explicitly deny an information flow based on the following rules: ***timeout: no data is transported on this connection for a configurable time (default 10 minutes).***

6.1.4 Class FIA: Identification and authentication

6.1.4.1 Authentication failures (FIA_AFL)

FIA_AFL.1 Authentication failure handling

FIA_AFL.1.1 The TSF shall detect when *an administrator configurable positive integer within* ***1 to infinite (default 5)*** unsuccessful authentication attempts occur related to ***authentication for administration, SMTP, SSH, FTP, TELNET, and side channel authentication***.

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been *surpassed*, the TSF shall ***prevent the offending user from successfully authentication until an authorised administrator takes some action to make authentication possible for the user in question***.

Application Note: This SFR only applies if the authentication method has been activated for SMTP, SSH, FTP, or TELNET.

6.1.4.2 User attribute definition (FIA_ATD)

FIA_ATD.1 User attribute definition

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

a) administrative role (or none);
b) user password.

6.1.4.3 Specification of secrets (FIA_SOS)

FIA_SOS.1 Verification of secrets

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet ***the following metric:***

- ***the password length is between 10 and 128 characters***
- ***the password contains at least one number***
- ***the password contains at least one lower case letter***
- ***the password contains at least one upper case letter***
- ***the password contains at least one symbol***

Application Note: This SFR does not apply to the password file authentication, because the file is imported from the outside. This SFR does not apply to authentication at an external RADIUS or LDAP server, because the passwords are configured at the external servers.

6.1.4.4 User authentication (FIA_UAU)

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 **for the authenticated user SFP, the authenticated gui user SFP and the authenticated administrator SFP**.

FIA_UAU.5EX External authentication mechanisms

FIA_UAU.5EX.1 The TSF shall provide ***password, RADIUS, LDAP, password file, and crypto card mechanisms*** to support user authentication by external means.

FIA_UAU.5EX.2 The TSF shall authenticate any user's claimed identity according to the ***following list:***

a) administrator authentication: password or LDAP;
b) gui user authentication: password;
c) user side channel authentication: password, RADIUS, LDAP, or crypto card (as configured by the administrator);
d) user authentication (SMTP, SSH, WWW): password, RADIUS, LDAP, or password file (as configured by the administrator).
e) user authentication (FTP and TELNET): password, RADIUS, LDAP, password file, or crypto card (as configured by the administrator);

Application Note: This SFR only applies if the authentication method has been activated for SMTP, SSH, WWW, FTP, or TELNET

FIA_UAU.6 Re-authenticating

FIA_UAU.6.1 The TSF shall re-authenticate the user under the conditions:

a) administrator authentication: timeout after inactivity (default 10 minutes, can be configured by an administrator);
b) gui user authentication: timeout after inactivity; (default 10 minutes, can be configured by an administrator);
c) user side channel authentication: after inactivity (default 10 minutes, can be configured by an administrator).

6.1.4.5 User identification (FIA_UID)

FIA_UID.2 User identification before any action

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

6.1.5 Class FMT: Security management

6.1.5.1 Management of functions in TSF (FMT_MOF)

FMT_MOF.1:1 Management of security functions behaviour

FMT_MOF.1.1:1 The TSF shall restrict the ability to *disable, enable, modify the behaviour of* the functions

a) the authentication methods for the side channel users, smtp-, ssh-, ftp-, and telnetrelay;
b) the usage of SMTP, SSH, FTP, or TELNET authentication;
c) the generation of audit trails;

to ***the administrator***.

FMT_MOF.1:2 Management of security functions behaviour

FMT_MOF.1.1:2 The TSF shall restrict the ability to *determine the behaviour of* the functions

a) the authentication methods for the side channel users;
b) the generation of audit trails;

to ***the administrator and auditor***.

FMT_MOF.1:3 Management of security functions behaviour

FMT_MOF.1.1:3 The TSF shall restrict the ability to **perform**~~[selection: determine the behaviour of, disable, enable, modify the behaviour of]~~ the functions ***start-up and shut-down, change to maintenance and normal operation mode, apply patches;*** to ***the administrator***.

Application Note: This SFR uses a refinement insetad of a selection, because the security function behavior is not changed by setting attributes but by performing maintenance actions.

6.1.5.2 Management of security attributes (FMT_MSA)

FMT_MSA.1:1 Management of security attributes

FMT_MSA.1.1:1 The TSF shall enforce the ***authenticated administrator SFP*** to restrict the ability to *change_default, modify, delete* the security attributes

a) the administrative role

to ***the administrator***.

FMT_MSA.1:2 Management of security attributes

FMT_MSA.1.1:2 The TSF shall enforce the ***authenticated administrator SFP*** to restrict the ability to *query* the security attributes

a) the administrative role
b) the patch level

to ***the administrator and the auditor***.

FMT_MSA.1:3 Management of security attributes

FMT_MSA.1.1:3 The TSF shall enforce the ***authenticated gui user SFP*** to restrict the ability to *modify* the security attributes

a) the user password

to ***the user***.

FMT_MSA.1:4 Management of security attributes

FMT_MSA.1.1:4 The TSF shall enforce the ***authenticated administrator SFP*** to restrict the ability to *modify* the security attributes

a) the user passwords;
b) the administrator password

to ***the administrator***.

FMT_MSA.3:1 Static attribute initialisation

FMT_MSA.3.1:1 The TSF shall enforce the ***authenticated user SFP*** to provide *restrictive* default values for security attributes that are used to enforce the SFP.

FMT_MSA3.2:1 The TSF shall allow the ***administrator*** to specify alternative initial values to override the default values when an object or information is created.

FMT_MSA.3:2 Static attribute initialisation

FMT_MSA.3.1:2 The TSF shall enforce the ***authenticated gui user SFP*** to provide *restrictive* default values for security attributes that are used to enforce the SFP.

FMT_MSA3.2:2 The TSF shall allow the ***administrator*** to specify alternative initial values to override the default values when an object or information is created.

FMT_MSA.3:3 Static attribute initialisation

FMT_MSA.3.1:3 The TSF shall enforce the ***authenticated administrator SFP*** to provide *restrictive* default values for security attributes that are used to enforce the SFP.

FMT_MSA3.2:3 The TSF shall allow the ***administrator*** to specify alternative initial values to override the default values when an object or information is created.

6.1.5.3 Management of TSF data (FMT_MTD)

FMT_MTD.1:1 Management of TSF data

FMT_MTD.1.1:1 The TSF shall restrict the ability to *modify, delete,* ***create*** the

a) users;
b) network configuration;
c) relay configuration;
d) dns server configuration;
e) mail server configuration;
f) packet filter rules;
g) http-proxy squid configuration;
h) virus scanner configuration;
i) audit configuration;
j) snmp server configuration;
k) igmpproxy configuration (on the PFL);

to ***the administrator***.

FMT_MTD.1:2 Management of TSF data

FMT_MTD.1.1:2 The TSF shall restrict the ability to *query* the

a) users;
b) network configuration;
c) relay configuration;
d) dns server configuration;
e) mail server configuration;
f) packet filter rules;
g) http-proxy squid configuration;
h) virus scanner configuration;
i) audit configuration;
j) snmp server configuration;
k) igmpproxy configuration (on the PFL);

to ***the administrator and auditor***.

6.1.5.4 Specification of Management Functions (FMT_SMF)

FMT_SMF.1 Specification of Management Functions

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

a) user configuration;
b) network configuration;
c) relay configuration;
d) dns server configuration;
e) mail server configuration;

f) packet filter rule configuration;
g) http-proxy squid configuration;
h) virus scanner configuration;
i) audit configuration;
j) snmp server configuration;
k) igmpproxy configuration (on the PFL);
l) verify the current patch level;
m) apply patches from the developer.

6.1.5.5 Security management roles (FMT_SMR)

FMT_SMR.2 Restrictions on security roles

FMT_SMR.2.1 The TSF shall maintain the roles ***administrator, auditor, user***.

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

FMT_SMR.2.3 The TSF shall ensure that the conditions: ***the source IP addresses for traffic controlled by the authenticated administrator SFP is from the administration network,*** are satisfied.

FMT_SMR.3 Assuming roles

FMT_SMR.3.1 The TSF shall require an explicit request to assume the following roles: ***administrator, auditor***.

6.1.6 Class FPT: Protection of the TSF

6.1.6.1 Trusted recovery (FPT_RCV)

FPT_RCV.2 Automated recovery

FPT_RCV.2.1 When automated recovery from ***a failure or service discontinuity*** is not possible, the TSF shall enter a maintenance mode where the ability to return to a secure state is provided.

FPT_RCV.2.2 For ***configurable events (default: none)***, the TSF shall ensure the return of the TOE to a secure state using automated procedures.

6.1.6.2 Simple Self Test (FPT_SST)

FPT_SST.1EX TOE testing

FPT_SST.1EX.1 The TSF shall run a suite of self tests *periodically during normal operation* to perform the following checks:

a) specified processes are running (default: all relays, daemons, web servers).
b) the file system usage is below a threshold (default: 90%)

c) the file system permissions and flags.

FPT_SST.1EX.2 The TSF shall provide authorised users with the capability to query the results of the following checks:

a) specified processes are running (default: all relays, dns server, snmp server, xntpd, postfix)
b) the file system usage is below a threshold (default: 90%)
c) the file system permissions and flags.

6.1.6.3 Time stamps (FPT_STM)

FPT_STM.1 Reliable time stamps

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps.

Application Note: The reliability is realized by synchronizing the real time clock with a time server using the protocol NTP Version4.

6.1.6.4 Internal TOE TSF data replication consistency (FPT_TRC)

FPT_TRC.1 Internal TSF consistency

FPT_TRC.1.1 The TSF shall ensure that TSF data is consistent when replicated between parts of the TOE.

FPT_TRC.1.2 When parts of the TOE containing replicated TSF data are disconnected, the TSF shall ensure the consistency of the replicated TSF data upon reconnection before processing any requests for ***services provided by the unauthenticated user SFP, the authenticated user SFP, the identified side channel use SFP, the authenticated gui user SFP, and the authenticated administrator SFP***.

Application Note: The systems use an internal revision number to check the configuration. They only reactivate services when their configuration is up to date. The new configuration is used only for new connections, existing connections are not reconfigured.

6.1.6.5 TOE Update (FPT_UPD)

FPT_UPD.1EX Trusted Update

FPT_UPD.1EX.1 The TOE shall cryptographically verify additional code/patches to itself using a digital signature prior to installation using schemes specified in ***FCS_COP.1***.

FPT_UPD.1EX.2 A modification of the TOE shall only be allowed if the software update

- is intended for the current software version,
- has the correct patch level and
- has been cryptographically verified with regard to integrity and authenticity.

FPT_UPD.2EX Update identification data

FPT_UPD.2EX.1 The TSF shall verify if the activation of the patch and the update of the identification data have been both completed.

FPT_UPD.2EX.2 The TSF shall update the active identification data when the patch is applied in order to keep the system in a defined state.

FPT_UPD.2EX.3 The TSF shall use the maintenance mode to activate the final TOE.

6.2 Security Assurance Requirements

In order to handle patch management, the Security Target defines one new assurance component for the class ALC: Life-cycle support, defined in chapter 5.4.1.
Table 3 shows the Security Assurance Requirements for the level EAL4. The augmented components ALC_FLR.2, ASE_TSS.2 and AVA_VAN.5 are set in a bold font. For the level EAL4, the SARs ADV_INT and ADV_SPM are not needed.
The table also contains the new assurance component ALC_PAM.1.

Table 3: SAR

Class	**Family**	**Level**	**Name**
Development	ADV_ARC	ADV_ARC.1	Security architecture description
	ADV_FSP	ADV_FSP.4	Complete functional specification
	ADV_IMP	ADV_IMP.1	Implementation representation of the TSF
	ADV_INT		TSF internals
	ADV_SPM		Security policy modelling
	ADV_TDS	ADV_TDS.3	Basic modular design
Guidance	AGD_OPE	AGD_OPE.1	Operational user guidance
	AGD_PRE	AGD_PRE.1	Preparative procedures
Life-cycle	ALC_CMC	ALC_CMC.4	Production support, acceptance procedures and automation
	ALC_CMS	ALC_CMS.4	Problem tracking CM coverage
	ALC_DEL	ALC_DEL.1	Delivery procedures
	ALC_DVS	ALC_DVS.1	Identification of security measures
	ALC_FLR	**ALC_FLR.2**	Flaw reporting procedures
	ALC_LCD	ALC_LCD.1	Developer defined life-cycle model
	ALC_TAT	ALC_TAT.1	Well-defined development tools
	ALC_PAM	**ALC_PAM.1**	Patch Management Processes
Security Target	ASE_CCL	ASE_CCL.1	Conformance claims
	ASE_ECD	ASE_ECD.1	Extended components definition
	ASE_INT	ASE_INT.1	ST introduction
	ASE_OBJ	ASE_OBJ.2	Security objectives
	ASE_REQ	ASE_REQ.2	Derived security requirements
	ASE_SPD	ASE_SPD.1	Security problem definition
	ASE_TSS	**ASE_TSS.2**	TOE summary specification with architectural design summary
Tests	ATE_COV	ATE_COV.2	Analysis of coverage
	ATE_DPT	ATE_DPT.1	Testing: basic design
	ATE_FUN	ATE_FUN.1	Functional testing
	ATE_IND	ATE_IND.2	Independent testing - sample
Vulnerability	AVA_VAN	**AVA_VAN.5**	Advanced methodical vulnerability analysis

6.3 Security Functional Requirements Rationale

The following table 4 shows that all dependencies are met (see notes at end of table):

Table 4: SFR Dependencies

Id	SFR	Dependencies	Satisfied by
1-1	FAU_ARP.1	FAU_SAA.1	1-3
1-2	FAU_GEN.1EX	FPT_STM.1	5-3
1-3	FAU_SAA.1	FAU_GEN.1	1-2
1-4	FAU_SAR.1	FAU_GEN.1	1-2
1-5	FAU_SAR.2	FAU_SAR.1	1-4
1-6	FAU_SAR.3	FAU_SAR.1	1-4
1-7	FAU_STG.1:1	FAU_GEN.1	1-2
1-8	FAU_STG.1:2	FAU_GEN.1	1-2
1-9	FAU_STG.4:1	FAU_STG.1	1-7, 1-8
1-10	FAU_STG.4:2	FAU_STG.1	OE.TRUSTK
1-11	FCS_COP.1	FDP_ITC.1 FCS_CKM.4	ALC_DEL N/A
2-1-1	FDP_IFC.1:1	FDP_IFF.1:1	2-2-1
2-1-2	FDP_IFC.1:2	FDP_IFF.1:2	2-2-2
2-1-3	FDP_IFC.1:3	FDP_IFF.1:3	2-2-3
2-1-4	FDP_IFC.1:4	FDP_IFF.1:4	2-2-4
2-1-5	FDP_IFC.1:5	FDP_IFF.1:5	2-2-5
2-2-1	FDP_IFF.1:1	FDP_IFC.1:1 FMT_MSA.3:X	2-1-1 N/A
2-2-2	FDP_IFF.1:2	FDP_IFC.1:2 FMT_MSA.3:1	2-1-2 4-3-1
2-2-3	FDP_IFF.1:3	FDP_IFC.1:3 FMT_MSA.3:X	2-1-3 N/A
2-2-4	FDP_IFF.1:4	FDP_IFC.1:4 FMT_MSA.3:2	2-1-4 4-3-2
2-2-5	FDP_IFF.1:5	FDP_IFC.1:5 FMT_MSA.3:3	2-1-5 4-3-3
3-1	FIA_AFL.1	FIA_UAU.1	3-4 (hierarchical)
3-2	FIA_ATD.1		
3-3	FIA_SOS.1		
3-4	FIA_UAU.2	FIA_UID.1	3-7 (hierarchical)
3-5	FIA_UAU.5EX		
3-6	FIA_UAU.6		
3-7	FIA_UID.2		

Table 4: SFR Dependencies (continued)

Id	SFR	Dependencies	Satisfied by
4-1-1	FMT_MOF.1:1	FMT_SMF.1	4-5
		FMT_SMR.1	4-6 (hierarchical)
4-1-2	FMT_MOF.1:2	FMT_SMF.1	4-5
		FMT_SMR.1	4-6 (hierarchical)
4-1-3	FMT_MOF.1:3	FMT_SMF.1	4-5
		FMT_SMR.1	4-6 (hierarchical)
4-2-1	FMT_MSA.1:1	FDP_IFC.1:5	2-1-5
		FMT_SMF.1	4-5
		FMT_SMR.1	4-6 (hierarchical)
4-2-2	FMT_MSA.1:2	FDP_IFC.1:5	2-1-5
		FMT_SMF.1	4-5
		FMT_SMR.1	4-6 (hierarchical)
4-2-3	FMT_MSA.1:3	FDP_IFC.1:4	2-1-4
		FMT_SMF.1	4-5
		FMT_SMR.1	4-6 (hierarchical)
4-2-4	FMT_MSA.1:4	FDP_IFC.1:5	2-1-5
		FMT_SMF.1	4-5
		FMT_SMR.1	4-6 (hierarchical)
4-3-1	FMT_MSA.3:1	FMT_MSA.1:3	4-2-3
		FMT_MSA.1:4	4-2-4
		FMT_SMR.1	4-6 (hierarchical)
4-3-2	FMT_MSA.3:2	FMT_MSA.1:3	4-2-3
		FMT_MSA.1:4	4-2-4
		FMT_SMR.1	4-6 (hierarchical)
4-3-3	FMT_MSA.3:3	FMT_MSA.1:1	4-2-1
		FMT_MSA.1:2	4-2-2
		FMT_SMR.1	4-6 (hierarchical)
4-4-1	FMT_MTD.1:1	FMT_SMF.1	4-5
		FMT_SMR.1	4-6 (hierarchical)
4-4-2	FMT_MTD.1:2	FMT_SMF.1	4-5
		FMT_SMR.1	4-6 (hierarchical)
4-5	FMT_SMF.1		
4-6	FMT_SMR.2	FIA_UID.1	3-7 (hierarchical)
4-7	FMT_SMR.3	FMT_SMR.1	4-6 (hierarchical)
5-1	FPT_RCV.2	AGD_OPE.1	R05, table 6
5-2	FPT_SST.1EX		
5-3	FPT_STM.1		
5-4	FPT_TRC.1	FPT_ITT.1	environment (OE.HANET)
5-5	FPT_UPD.1EX	FCS_COP.1	1-11
5-6	FPT_UPD.2EX		

- FAU_ARP.1 depends on audit data, that is generated by FAU_SAA.1.
- FAU_GEN.1EX depends on FPT_STM.1 that requires reliable time stamps. The objectives **OE.TIMESTMP** and **OE.RTCLOCK** provide means to attain these reliable time stamps.
- FAU_SAA.1 depends on FAU_GEN.1 which is fulfilled by FAU_GEN.1EX.
- FAU_SAR.1 depends on FAU_GEN.1 which is fulfilled by FAU_GEN.1EX.
- FAU_SAR.2 depends on FAU_SAR.1 which provides audit review.
- FAU_SAR.3 depends on FAU_SAR.1 which provides audit review.
- FAU_STG.1:1 depends on FAU_GEN.1 which is fulfilled by FAU_GEN.1EX.
- FAU_STG.1:2 depends on FAU_GEN.1 which is fulfilled by FAU_GEN.1EX.
- FAU_STG.4:1 depends on FAU_STG.1:1 and FAU_STG.1:2, because the application level audit trail consists of the rotated and the flagged audit trail.
- FAU_STG.4:2 depends on FAU_STG.1. In this case the environment provides the security functionality because it is trustworthy not to alter the log data, by **OE.TRUSTK**.
- FCS_COP.1 depends on FDP_ITC.1. The (public) key to verify the signature of the patch is distributed on the installation medium that is secured by the delivery process in ALC_DEL. Therefore no explicit import function is necessary.
  The SFR also depends on FCS_CKM.4. Only the public key is needed, therefore the SFR is not needed.
- FDP_IFC.1:1: The policy for the unauthenticated user SFP is FDP_IFF.1:1.
- FDP_IFC.1:2: The policy for the authenticated user SFP is FDP_IFF.1:2.
- FDP_IFC.1:3: The policy for the identified side channel user SFP is FDP_IFF.1:3.
- FDP_IFC.1:4: The policy for the authenticated gui user SFP is FDP_IFF.1:4.
- FDP_IFC.1:5: The policy for the authenticated administrator SFP is FDP_IFF.1:5.
- FDP_IFF.1:1: This is the flow control function for the unauthenticated user SFP defined in FDP_IFC.1:1. The dependency of FMT_IFF.1:1 on FMT_MSA.3:X is not applicable because the users that fall under this SFP do not have the security attributes administrative role or password.
- FDP_IFF.1:2: This is the flow control function for the authenticated user SFP defined in FDP_IFC.1:2.
- FDP_IFF.1:3: This is the flow control function for the identified side channel user SFP defined in FDP_IFC.1:3. The dependency of FMT_IFF.1:3 on FMT_MSA.3:X is not applicable because the users that fall under this SFP do not have the security attributes administrative role or password.
- FDP_IFF.1:4: This is the flow control function for the authenticated gui user SFP defined in FDP_IFC.1:4.
- FDP_IFF.1:5: This is the flow control function for the authenticated administrator SFP defined in FDP_IFC.1:5.
- FIA_AFL.1 depends on FIA_UAU.1 which is fulfilled by FIA_UAU.2.
- FIA_ATD.1 has no dependencies.
- FIA_SOS.1 has no dependencies.
- FIA_UAU.2 depends on FIA_UID.1 which is met by FIA_UID.2 which is hierarchical.
- FIA_UAU.5EX has no dependencies.
- FIA_UAU.6 has no dependencies.
- FIF_UID.2 has no dependencies.
- FMT_MOF.1:1: The management functions are specified in FMT_SMF.1. The security role administrator is defined in FMT_SMR.2 which is hierarchical to FMT_SMR.1.
- FMT_MOF.1:2: The management functions are specified in FMT_SMF.1. The security roles

administrator and auditor are defined in FMT_SMR.2 which is hierarchical to FMT_SMR.1.

- FMT_MOF.1:3: The management functions are specified in FMT_SMF.1. The security role administrator is defined in FMT_SMR.2 which is hierarchical to FMT_SMR.1.
- FMT_MSA.1:1: The flow control function for the authenticated administrator SFP is defined in FDP_IFC.1:5. The management functions are specified in FMT_SMF.1. The security role administrator is defined in FMT_SMR.2 which is hierarchical to FMT_SMR.1.
- FMT_MSA.1:2: The flow control function for the authenticated administrator SFP is defined in FDP_IFC.1:5. The management functions are specified in FMT_SMF.1. The security roles administrator and auditor are defined in FMT_SMR.2 which is hierarchical to FMT_SMR.1.
- FMT_MSA.1:3: The flow control function for the authenticated gui user SFP is defined in FDP_IFC.1:4. The management functions are specified in FMT_SMF.1. The security role user is defined in FMT_SMR.2 which is hierarchical to FMT_SMR.1.
- FMT_MSA.1:4: The flow control function for the authenticated administrator SFP is defined in FDP_IFC.1:5. The management functions are specified in FMT_SMF.1. The security role administrator is defined in FMT_SMR.2 which is hierarchical to FMT_SMR.1.
- FMT_MSA.3:1: The management of the respective password can be done by the user (FMT_MSA.1:3) or the administrator (FMT_MSA.1:4). Their roles are defined in FMT_SMR.2 which is hierarchical to FMT_SMR.1.
- FMT_MSA.3:2: The management of the user password can be done by the user (FMT_MSA.1:3) or the administrator (FMT_MSA.1:4). Their roles are defined in FMT_SMR.2 which is hierarchical to FMT_SMR.1.
- FMT_MSA.3:3: The administrative role can be changed by the administrator (FMT_MSA.1:1) and viewed by the auditor (FMT_MSA.1:2). Their roles are defined in FMT_SMR.2 which is hierarchical to FMT_SMR.1.
- FMT_MTD.1:1: The management functions are specified in FMT_SMF.1. The security role administrator is defined in FMT_SMR.2 which is hierarchical to FMT_SMR.1.
- FMT_MTD.1:2: The management functions are specified in FMT_SMF.1. The security role auditor is defined in FMT_SMR.2 which is hierarchical to FMT_SMR.1.
- FMT_SMF.1 has no dependencies.
- FMT_SMR.2: The SFR depends on FIA_UID.1 which is met by FIA_UID.2 which is hierarchical.
- FMT_SMR.3: The security roles are defined in FMT_SMR.2 which is hierarchical to FMT_SMR.1.
- FPT_RCV.2 depends on AGD_OPE.1 which gives the necessary guidance.
- FPT_SST.1EX has no dependencies.
- FPT_STM.1 has no dependencies.
- FPT_TRC.1: The SFR depends on FPT_ITT.1 which requires the protection of the TSF transfer against disclosure (or modification). This requirement is satisfied by the objective **OE.HANET** that requires a physical network for the transfer that prohibits disclosure.
- FPT_UPD.1EX depends on FCS_COP.1, which is used for signature verifcation
- FPT_UPD.2EX has no dependencies

6.3.1 Objectives

This section must show that the SFR address the objectives, and that all dependencies between the SFRs and SARs are met.

O.IDAUTH FIA_AFL.1: This component describes the actions of authentication failure handling.

FIA_ATD.1: This component defines the user attributes.

FIA_SOS.1: This component specifies the used secrets.

FIA_UAU.2: This component requires a user authentication before any action.

FIA_UAU.5EX: This component describes all possible authentication mechanisms.

FIA_UAU.6: This component describes under which circumstances a re authentication is necessary.

FIA_UID.2: This component requires a user identification before any action.

The SFRs are mutually supportive. They are sufficient to meet the objective.

O.MEDIAT FDP_IFC.1:1: This component defines the unauthenticated user SFP that describes the data flow control for users of the firewall.

FDP_IFC.1:2: This component defines the authenticated user SFP that describes the data flow control for users of the firewall that use the smtp-, ssh-, ftp-, or telnetrelay.

FDP_IFC.1:3: This component defines the identified side channel user SFP that describes the data flow control for users of the firewall that use the side channel authentication.

FDP_IFC.1:4: This component defines the authenticated gui user SFP that describes the data flow control for users of the firewall that change their password or register a side channel.

FDP_IFC.1:5: This component defines the authenticated administrator SFP that describes the data flow control for administrators of the firewall.

FDP_IFF.1:1: This component describes the access control for the unauthenticated user SFP.

FDP_IFF.1:2: This component describes the access control for the authenticated user SFP.

FDP_IFF.1:3: This component describes the access control for the identified side channel user SFP.

FDP_IFF.1:4: This component describes the access control for the authenticated gui user SFP.

FDP_IFF.1:5: This component describes the access control for the authenticated administrator SFP.

The SFRs describe all possible access ways to the TOE and their related policies. The SFRs are mutually supportive. They are sufficient to meet the objective.

O.SECSTA FPT_RCV.2: This component describes a recovery after failures. The SFR is sufficient to meet the objective.

O.SELPRO FPT_SST.1EX: This component defines simple self-tests.

O.SECFUN FMT_MOF.1:1: This component defines who can modify the behaviour of the security functions.

FMT_MOF.1:2: This component defines who can read the settings of the security functions.

FMT_MOF.1:3: This component defines who can start and stop the TOE or enter maintenance or normal operation. These actions also modify the behaviour of the security functions.

FMT_MSA.3:1: This component describes that the authenticated user SFP has restrictive default values of the security attributes (the user password).

FMT_MSA.3:2: This component describes that the authenticated gui user SFP has restrictive default values of the security attributes (the user password).

FMT_MSA.3:3: This component describes that the authenticated administrator SFP has restrictive default values of the security attributes (the administrator password).

FMT_MTD.1:1: This component describes who can modify the TSF data.

FMT_MTD.1:2: This component describes who can query the TSF data.

FMT_SMF.1: This component lists the configuration data of the TSF.

FMT_SMR.2: The component defines the security roles.

FMT_SMR.3: This component describe that in order to assume the administrator or the auditor role, an explicit request must be required.

FMT_MSA.1:1: This component defines who can change the administrative role, i.e. who is administrator.

FMT_MSA.1:2: This component defines who can query the administrative role.

FMT_MSA.1:3: This component describes that the users can change their own password.

FMT_MSA.1:4: This component describes that the administrator can change the user and the administrative passwords.

The SFRs describe the security sensitive data on the TOE and the configurable security functions. The SFRs describe who can read/read the data and change the security functions. The SFRs are mutually supportive. They are sufficient to meet the objective.

O.MISUSESSH FDP_IFC.1:1: This component defines the unauthenticated user SFP that describes the data flow control for users of the firewall.

FDP_IFC.1:2: This component defines the authenticated user SFP that describes the data flow control for users of the firewall that use the sshrelay.

FDP_IFF.1:1: This component describes the access control for the unauthenticated user SFP.

FDP_IFF.1:2: This component describes the access control for the authenticated user SFP.

The SFRs describe all possible access ways to the TOE and their related policies. The SFRs are mutually supportive. They are sufficient to meet the objective.

O.AUDREC FAU_ARP.1: This component detects potential security violations.

FAU_GEN.1EX: This component describe the data generated for the audit.

FAU_SAA.1: The component describes the security violation analysis.

FAU_SAR.1: The component requires an audit review.

FAU_SAR.2: This component assigns who can view the audit log.

FAU_SAR.3: This component allows the searching of the audit log.

FAU_STG.1:1, FAU_STG.1:2: This component makes sure that the audit log is protected.

FAU_STG.4:1, FAU_STG.4:2: This component requires a prevention of audit data loss.

FPT_STM.1: This component provides reliable time stamps.

The SFRs are mutually supportive. They are sufficient to meet the objective.

O.ACCOUN FAU_GEN.1EX: This component describes the data generated for the audit.

FIA_UID.2: This component requires a user identification before any action.

FIA_UAU.2: This component requires a user authentication before any action.

The SFRs are mutually supportive. They are sufficient to meet the objective.

O.AVAIL FPT_TRC.1: This component requires that replicated data is consistent between parts of the TOE and that they check the consistency of the replicated data before accepting user connections.

O.PATCH FCS_COP.1: This component defines the cryptographic operations for patch signature verification.

FPT_UPD.1EX: This component defines the checks for authentic patches.

FPT_UPD.2EX: This component defines unique patch levels and its display.

The SFRs are mutually supportive. They are sufficient to meet the objective.

The following table 5 shows that all SFR contribute to (at least one objective) and all objectives are met by (at least) one SFR.
The following text shows how the SFR help to maintain the objectives.

FAU_ARP.1 This component detects potential security violations and aids in meeting the objective **O.AUDREC**.

FAU_GEN.1EX This component describes the data generated for the audit and aids in meeting the objective **O.AUDREC**. It also aids in meeting **O.ACCOUN**.

FAU_SAA.1 The component describes the security violation analysis and aids in meeting the objective **O.AUDREC**.

FAU_SAR.1 The component requires an audit review and contributes to the objectives **O.AUDREC**.

FAU_SAR.2 This component assigns who can view the audit log and contributes to **O.AUDREC**.

FAU_SAR.3 This component allows the searching of the audit log and contributes to **O.AUDREC**.

FAU_STG.1:1 This component makes sure that the audit log can be written and contributes to **O.AUDREC**.

FAU_STG.1:2 This component requires a prevention of audit data loss and contributes to **O.AUDREC**.

FAU_STG.4:1 This component makes sure that the audit log can be written and contributes to **O.AUDREC**.

FAU_STG.4:2 This component requires a prevention of audit data loss and contributes to **O.AUDREC**.

FCS_COP.1 This component defines the cryptographic methods for patch signature verification. The component contributes to **O.PATCH**.

FDP_IFC.1:1 This component defines the unauthenticated user SFP that describes the data flow control for users of the firewall. The component aids in meeting **O.MEDIAT** and **O.MISUSESSH**.

FDP_IFC.1:2 This component defines the authenticated user SFP that describes the data flow control for users of the firewall that use the smtp- (if configured), ssh-, ftp, or telnetrelay. The component aids in meeting **O.MEDIAT** and **O.MISUSESSH**.

Table 5: SFR coverage

SFR	O.IDAUTH	O.MEDIAT	O.SECSTA	O.SELFPRO	O.SECFUN	O.MISUSESSH	O.AUDREC	O.ACCOUN	O.AVAIL	O.PATCH
FAU_ARP.1							X			
FAU_GEN.1EX							X	X		
FAU_SAA.1							X			
FAU_SAR.1							X			
FAU_SAR.2							X			
FAU_SAR.3							X			
FAU_STG.1:1							X			
FAU_STG.1:2							X			
FAU_STG.4:1							X			
FAU_STG.4:2							X			
FCS_COP.1										X
FDP_IFC.1:1		X				X				
FDP_IFC.1:2		X				X				
FDP_IFC.1:3		X								
FDP_IFC.1:4		X								
FDP_IFC.1:5		X								
FDP_IFF.1:1		X				X				
FDP_IFF.1:2		X				X				
FDP_IFF.1:3		X								
FDP_IFF.1:4		X								
FDP_IFF.1:5		X								
FIA_AFL.1	X									
FIA_ATD.1	X									
FIA_SOS.1	X									
FIA_UAU.2	X							X		
FIA_UAU.5EX	X									
FIA_UAU.6	X									
FIA_UID.2	X							X		

FDP_IFC.1:3 This component defines the identified side channel user SFP that describes the data flow control for users of the firewall that use the side channel authentication. The component aids in meeting **O.MEDIAT**.

FDP_IFC.1:4 This component defines the authenticated gui user SFP that describes the data flow control for users of the firewall that change their password or register a side channel. The component aids in meeting **O.MEDIAT**.

Table 5: SFR coverage (continued)

SFR	O.IDAUTH	O.MEDIAT	O.SECSTA	O.SELFPRO	O.SECFUN	O.MISUSESSH	O.AUDREC	O.ACCOUN	O.AVAIL	O.PATCH
FMT_MOF.1:1					X					
FMT_MOF.1:2					X					
FMT_MOF.1:3					X					
FMT_MSA.1:1					X					
FMT_MSA.1:2					X					
FMT_MSA.1:3					X					
FMT_MSA.1:4					X					
FMT_MSA.3:1					X					
FMT_MSA.3:2					X					
FMT_MSA.3:3					X					
FMT_MTD.1:1					X					
FMT_MTD.1:2					X					
FMT_SMF.1					X					
FMT_SMR.2					X					
FMT_SMR.3					X					
FPT_RCV.2			X							
FPT_SST.1EX				X						
FPT_STM.1							X			
FPT_TRC.1									X	
FPT_UPD.1EX										X
FPT_UPD.2EX										X

FDP_IFC.1:5 This component defines the authenticated administrator SFP that describes the data flow control for administrators of the firewall. The component aids in meeting **O.MEDIAT**.

FDP_IFF.1:1 This component describes the access control for the unauthenticated user SFP and contributes to **O.MEDIAT** and **O.MISUSESSH**.

FDP_IFF.1:2 This component describes the access control for the authenticated user SFP and contributes to **O.MEDIAT** and **O.MISUSESSH**.

FDP_IFF.1:3 This component describes the access control for the identified side channel user SFP and contributes to **O.MEDIAT**.

FDP_IFF.1:4 This component describes the access control for the authenticated gui user SFP and contributes to **O.MEDIAT**.

FDP_IFF.1:5 This component describes the access control for the authenticated administrator SFP and contributes to **O.MEDIAT**.

FIA_AFL.1 This component describes the actions of authentication failure handling and contributes to **O.IDAUTH**.

FIA_ATD.1 This component defines the user attributes and aids in meeting the objective **O.IDAUTH**.

FIA_SOS.1 The verification of secrets contributes to **O.IDAUTH**.

FIA_UAU.2 This component requires a user authentication before any action. It contributes to **O.IDAUTH**. It also aids in meeting **O.ACCOUN**, as the users are authenticated.

FIA_UAU.5EX This component describes all possible authentication mechanisms and helps to meet **O.IDAUTH**.

FIA_UAU.6 This component describes under which circumstances a re-authentication is necessary and contributes to **O.IDAUTH**.

FIA_UID.2 This component requires a user identification before any action. It contributes to **O.IDAUTH**. It also aids in meeting **O.ACCOUN**, because log entries can be associates with users.

FMT_MOF.1:1 This component defines who can modify the behaviour of the security functions. It contributes to **O.SECFUN**.

FMT_MOF.1:2 This component defines who can read the settings of the security functions. It contributes to **O.SECFUN**.

FMT_MOF.1.3 This component defines who can start and stop the TOE or enter maintenance or normal operation. These actions also modify the behaviour of the security functions. The component contributes to **O.SECFUN**.

FMT_MSA.1:1 This component defines who can change the administrative role, i.e. who is administrator. The component contributes to **O.SECFUN**.

FMT_MSA.1:2 This component defines who can query the administrative role. It contributes to **O.SECFUN**.

FMT_MSA.1:3 This component describes that the users can change their own password. It contributes to **O.SECFUN**.

FMT_MSA.1:4 This component describes that the administrator can change the user and the administrative passwords. It contributes to **O.SECFUN**.

FMT_MSA.3:1 This component describes that the authenticated user SFP has restrictive default values of the security attributes. The component contributes to **O.SECFUN**.

FMT_MSA.3:2 This component describes that the authenticated gui user SFP has restrictive default values of the security attributes. The component contributes to **O.SECFUN**.

FMT_MSA.3:3 This component describes that the authenticated administrator SFP has restrictive default values of the security attributes. The component contributes to **O.SECFUN**.

FMT_MTD.1:1 This component describes who can modify the TSF data. It contributes to **O.SECFUN**.

FMT_MTD.1:2 This component describes who can query the TSF data. It contributes to **O.SECFUN**.

FMT_SMF.1 This component lists the configuration data of the TSF. It contributes to **O.SECFUN**.

FMT_SMR.2 The component defines the security roles. It contributes to **O.SECFUN**.

FMT_SMR.3 This component describes that in order to assume the administrator or the auditor role, an explicit request must be required. This component contributes to **O.SECFUN**.

FPT_RCV.2 This component describes a recovery after failures and contributes to **O.SECSTA**.

FPT_SST.1EX This component defines simple self-tests. It contributes to **O.SELPRO**.

FPT_STM.1 This component provides reliable time stamps and contributes to **O.AUDREC**.

FPT_TRC.1 This component requires consistency in the TSF data when it is replicated internal to the TOE. It avoids inconsistent states in the takeover case and aids to meet **O.AVAIL**.

FPT_UPD.1EX This component defines the checks for authentic patches. The component contributes to **O.PATCH**.

FPT_UPD.2EX This component defines unique patch levels and its display. The component contributes to **O.PATCH**.

6.3.2 New or tailored SFR

The following rationale justifies the introduction of new SFR components and families.

FAU_GEN.1EX: This component is derived from FAU_GEN.1, but omits the audit events on start-up and shutdown of the audit functions. The replacement can be used if the omitted functionality is not supported. All other requirements are taken literally from FAU_GEN.1. The SFR that depend on FAU_GEN.1, usually require only the still supported security functions. FAU_GEN.1EX can therefore be used as a replacement for FAU_GEN.1. The dependency on FAU_GEN.1 of other SFRs can be substituted by FAU_GEN.1EX. Because FAU_GEN.1EX is close connected to FAU_GEN.1, it has been added to the same family.

FIA_UAU.5EX: This component is derived from FIA_UAU.5, with the clarification that the SFR itself does not implement authentication methods, but uses methods outside of the TOE. This component is introduced only in order to clearly state the situation to the reader. As FIA_UAU.5EX provides the same functionality as FIA_UAU.5, it can be used as a replacement for FIA_UAU.5. The dependency on FIA_UAU.5 of other SFRs can be substituted by FIA_UAU.5EX. Because FIA_UAU.5EX is close connected to FIA_UAU.5, it has been added to the same family.

FPT_SST.1EX: The single component of this new family FPT_SST is modelled after component FPT_TST.1. The component FPT_TST.1 has a dependency on FPT_AMT.1. Self-tests can, however, also be performed without having a formal abstract state machine. In order to avoid any associations with these concept, a new family has been introduced. In addition, the tests do not just check the TSFs, but perform tests that can also check any other targets. Therefore, a new family seems justified.

FPT_UPD.1EX This is a new component in the new family FPT_UPD that handles software updates. This functionality is currently not covered in common criteria and therefore must be a new family and component.

FPT_UPD.2EX This is a new component in the new family FPT_UPD that handles software updates. This functionality is currently not covered in common criteria and therefore must be a new family and component.

6.4 Security Assurance Requirements Rationale

The overall security claim of this Security Target is aimed at EAL4.
The attack potential of the anonymous users is high. The firewall components are exposed to unrestricted attackers, simply because they are exposed to the Internet. Therefore the vulnerability analysis has been augmented to AVA_VAN.5 in order to match the resistance to attackers with a high attack potential.
For the same reason the TOE summary specification has been augmented to ASE_TSS.2. This augmentation explains the security architecture of the product.
The life cycle support has been augmented by ALC_FLR.2 to demonstrate genua's flaw handling procedures.
The component ALC_PAM.1 has been included in order to have a well defined, secure and correct patch generation process.
The new components are necessary, because application of patches has not been addressed by Common Criteria.
The following table 6 shows that all dependencies are met.

Table 6: SAR Dependencies

ID	Requirement	Dependency	Solution
R01	ADV_ARC.1	ADV_FSP.1	R02
		ADV_TDS.1	R04
R02	ADV_FSP.4	ADV_TDS.1	R04
R03	ADV_IMP.1	ADV_TDS.3	R04
		ADV_TAT.1	R14
R04	ADV_TDS.3	ADV_FSP.4	R02
R05	AGD_OPE.1	ADV_FSP.1	R02
R06	AGD_PRE.1	-	-
R07	ALC_CMC.4	ALC_CMS.1	R08
		ALC_DVS.1	R10
		ALC_LCD.1	R13
R08	ALC_CMS.4	-	-
R09	ALC_DEL.1	-	-
R10	ALC_DVS.1	-	-
R11	**ALC_PAM.1**	ALC_FLR.2	R12
R12	**ALC_FLR.2**	-	-
R13	ALC_LCD.1	-	-
R14	ALC_TAT.1	ADV_IMP.1	R03
R15	ASE_CCL.1	ASE_INT.1	R17
		ASE_ECD.1	R16
		ASE_REQ.1	R19
R16	ASE_ECD.1	-	-
R17	ASE_INT.1	-	-
R18	ASE_OBJ.2	ASE_SPD.1	R20
R19	ASE_REQ.2	ASE_OBJ.2	R18
		ASE_ECD.1	R16
R20	ASE_SPD.1	-	-
R21	**ASE_TSS.2**	ASE_INT.1	R17
		ASE_REQ.1	R19
		ADV_ARC.1	R01
R22	ATE_COV.2	ADV_FSP.2	R02
		ATE_FUN.1	R24
R23	ATE_DPT.1	ADV_ARC.1	R01
		ADV_TDS.2	R04
		ATE_FUN.1	R24
R24	ATE_FUN.1	ATE_COV.1	R22

Table 6: SAR Dependencies (continued)

ID	Requirement	Dependency	Solution
R25	ATE_IND.2	ADV_FSP.2	R02
		AGD_OPE.1	R05
		AGD_PRE.1	R06
		ATE_COV.1	R22
		ATE_FUN.1	R24
R26	**AVA_VAN.5**	ADV_ARC.1	R01
		ADV_FSP.2	R02
		ADV_TDS.3	R04
		ADV_IMP.1	R03
		AGD_OPE.1	R05
		AGD_PRE.1	R06

7 TOE Summary

7.1 TOE Summary Specification

7.1.1 SF_SA: Security audit

SF_SA.1: The TOE generates log data whenever important events occur. This includes starting and stopping of the system, and changing from normal to the maintenance mode. Starting and stopping or reconfiguration of the relays generate log data. Loading of packet filter rules for ALG and PFL generate log data.

SF_SA.2: All relays generate log data when the connection state changes. Log data includes the IP address of source and destination, Ports for TCP and UDP-based protocols, the time stamps for connection and disconnection and the amount of data transferred in both directions for the source and the destination side. The protocol specific relays log part of the protocol data (e.g. URLs, SMTP-Envelope-lines, ...). The telnet-, ftp, smtp-, and sshrelay (if configured) log information about authentication. All unsuccessful connection attempts are logged.

SF_SA.3: All administration through the admin web generates log data. The administration action is logged together with the administrative role. Successful and unsuccessful login attempts are logged. The log contains a time stamp.

SF_SA.4: The log data is analysed by automated tools that look for pattern in the log data. The pattern include packet filter violations, daemon messages, relay messages, kernel messages, ARP spoofing messages, failure of time synchronization, usage of duplicate IP addresses, and messages from other processes, e.g. the processes that implement the self-tests. If a pattern matches, a security event is generated. The actions include logging of the event, adding the event to an event digest, use of 'wall' to show the event on the consoles, mail the event to the administrators, create an process master event, shut down network interfaces, and system halt. The extracted log data is written to the audit log. In normal operation mode the audit log is protected by file system append-only flag. It can only be changed in maintenance mode (e.g. rotated).

SF_SA.5: The log data can be transformed into a human readable form and can be searched by all administrators and auditors. Other roles are not allowed to read the log. The possible search criteria are: time, date, process id and additional log data. For relays the log data contains: the relay type, connection state, IP addresses and ports, bytes transferred.

SF_SA.6: The application level audit trail is divided into two parts, the automatically rotated audit logs and the flagged audit logs. The log data for the automatically rotated audit logs will be deleted after multiple rounds of rotation. The flagged audit logs can only be rotated in maintenance mode with the approval of an administrator. The time span between the rotation passes is large enough so that the security audit can extract relevant log entries and write them to the flagged audit log.

The system monitors the application level audit trail. If it fills beyond a threshold, a configurable action is executed.

The process master receives an event from the kernel if the kernel audit trails is filled beyond a threshold or is totally filled. It then executes a configurable action which can range from ignoring the event to halting the system. If the process master does not react, the kernel will panic the system.

This Security Function addresses the following SFRs: FAU_GEN.1EX (runtime audit data generation); FAU_ARP.1 (automatic response); FAU_SAA.1 (audit analysis); FAU_STG.1:1, FAU_STG.1:2, FAU_STG.4:1, and FAU_STG.4:2 (audit storage); FAU_SAR.1, FAU_SAR.2, FAU_SAR.3 (audit review) and FPT_STM.1 (time stamps).

7.1.2 SF_DF: Data flow control

SF_DF.1: The packet filter at the ALG and PFL implement the flow control at the network layer (IP) and transport layer (TCP/UDP). The filter rules take the information from the IP and TCP/UDP-Header (where applicable) in order to apply the filter rules.

Packets with spoofed source- or destination-IP addresses are dropped. Packets with source routing are dropped. Packets are not forwarded at the ALG; so that packets that cannot be transmitted to the socket layer are dropped.

The packet filter of the PFL has a restrictive default filter set. Any TCP-connections (or UDP packets) from the ALG into the internal net have to be activated by an administrator.

SF_DF.2: The relays check the following attributes:

- The header information of network packets, depending on their type:

  TCP: IP and TCP header;

  UDP: IP and UDP header;

  ICMP: IP header and ICMP message;

  IGMP: IP header and IGMP message;

  IP: IP header;
- The incoming and outgoing interfaces.
- The actual date and time.
- Additional information depending on the handling relay:
  - iprelay: none;
  - pingrelay: none;
  - udprelay: protocol conformance by applying regular expressions at the start of the communication if the filter is activated.
  - tcprelay: protocol conformance by applying regular expressions at the start of the communication if the filter is activated.
  - smtprelay: protocol and application data;
  - smtp2smtprelay: protocol and application data;
  - imaprelay: protocol and application data;
  - poprelay: protocol and application data;
  - siprelay: protocol and application data;
  - sshrelay: protocol data;
  - wwwrelay: protocol and application data;
  - httprelay: protocol and application data.
  - ftprelay: protocol data;
  - telnetrelay: protocol data;

- mcastudprelay: IGMP and multicast UDP packets;

A virus scanner can be used to scan the application data of smtprelay, poprelay, ftprelay, wwwrelay, WWWserver-relay, smtp2smtprelay, and imaprelay.

SF_DF.3: The smtprelay can block mails depending on the mail data (virus, blocked extension type of a MIME part). The mail stays on the TOE and must be handled by an administrator.

SF_DF.4: wwwrelay: For data of the content-type text/html a filter can remove the following tags that imply active content: `<applet>`, `<embed>`, `<object>`, `<script>`, and comments. Typical JavaScript-fragments, like event handler (on-tags) can also be removed.

SF_DF.5: MIME-encoded messages are (recursively) parsed. Their parts are checked like non encoded messages.

SF_DF.6: The sshrelay can block the following SSH protocol messages: shell spawning, command execution and file transfer with scp, local port forwarding, remote port forwarding, X11 forwarding, authentication agent forwarding, and subsystem execution.

SF_DF.7: The siprelay can block connections that do not use the configured internal and external domains or use RTP ports outside the configured port range. The protocol methods can be filtered.

SF_DF.8: The webservice policy of the httprelay can validate the application data against configurable XML schemas and use only configurable transport protocols.

SF_DF.9: An authenticated administrator can terminate connections in the traffic monitor section at the genugate administration interface or add an IP address to a list of blocked IP addresses.

This Security Function addresses the SFRs: FDP_IFC.1:1, FDP_IFC.1:2, FDP_IFC.1:3, FDP_IFC.1:4, and FDP_IFC.1:5 (information flow control policy); FDP_IFF.1:1, FDP_IFF.1:2, FDP_IFF.1:3, FDP_IFF.1:4, and FDP_IFF.1:5 (information flow control functions). They cover the policies ***unauthenticated user SFP***, ***authenticated user SFP***, ***identified side channel user SFP***, ***authenticated gui user SFP***, and ***authenticated administrator SFP***.

7.1.3 SF_IA: Identification and Authentication

SF_IA.1: All IP packets are identified at the network layer by their source and destination IP addresses (and ports if applicable).

SF_IA.2: The TCP-based relays are already connection oriented. The UDP- and IP-related relays introduce a UDP-association or IP-association respectively. Packages with the same destination IP, (destination port,) source IP, (source port,) and packets where source and destination are reversed are treated as belonging to a connection if they appear within a short timespan one after the other. The connections time out after an idle time with no traffic. As with TCP connections, the connection establishment can be configured to be initiated only by one side. For the iprelay, the IP protocol takes the role of the port.

SF_IA.3: For the ftp-, telnet-, smtp-, ssh-, and wwwrelay a user authentication at the TOE can be configured by the administrator. The authentication method can be configured and either be

password, RADIUS, LDAP, or password file. An additional method for the telnet- and ftprelay is crypto card.

The password can be changed by the users themselves, but a minimum quality is checked by the TOE:

- the password length is between 10 and 128 characters
- the password contains at least one number
- the password contains at least one lower case letter
- the password contains at least one upper case letter
- the password contains at least one symbol

For the password file authentication method, the password can not be changed by the users.

The telnet- and ftprelay capture the eventual option-negotiation commands sent before the authentication proceeds, and replay them to the destination, if the authentication completes successfully.

SF_IA.4: The side channel authentication allows users to activate configurable tcprelays after a successful authentication at the side channel web site. The authentication method can be configured by the administrators and either be password, RADIUS, LDAP, or crypto card. The password can be changed by the users themselves, but a minimum quality is checked by the TOE:

- the password length is between 10 and 128 characters
- the password contains at least one number
- the password contains at least one lower case letter
- the password contains at least one upper case letter
- the password contains at least one symbol

SF_IA.5: The user web allow the gui users to change their passwords after a successful authentication. The authentication method is password. The password can be changed by the users themselves, but a minimum quality is checked by the TOE:

- the password length is between 10 and 128 characters
- the password contains at least one number
- the password contains at least one lower case letter
- the password contains at least one upper case letter
- the password contains at least one symbol

SF_IA.6: Administration is only possible after successful authentication at the admin web server. Auditors (administrators with read-only rights) can view the configuration after successful authentication at the admin web server. Connections to the admin web server are only accepted from the administration network. The authentication method is password. The password can be changed by the respective administrators themselves, but a minimum quality is checked by the TOE.

- the password length is between 10 and 128 characters
- the password contains at least one number
- the password contains at least one lower case letter
- the password contains at least one upper case letter
- the password contains at least one symbol

SF_IA.7: All of the different authentication methods disable a user/administrator account after a configurable number of unsuccessful attempts. The default value is 5. An administrator has to reactivate the user account.

SF_IA.8: The side channel, user and the admin web server have a timeout for inactivity, after which the user/administrator have to re-authenticate. The default timeout is 10 minutes.

SF_IA.9: To gain interactive access (shell access) to the console, the administrator has to authenticate. Other interactions at the console require administrator input. On (re)boot the system waits for keyboard input but does not require a password. The application of boot install scripts in maintenance mode continue without applying the scripts, if the password is not entered during the timeout period. Changing the kernel requires keyboard input but does not require a password.

This Security Function addresses the SFRs: FIA_AFL.1 (authentication failures), FIA_SOS.1 (specification of secrets), FIA_UAU.2, FIA_UAU.5EX, FIA_UAU.6 (user authentication), FIA_UID.2 (user identification); FDP_IFC.1:2, FDP_IFC.1:3, FDP_IFC.1:4, and FDP_IFC.1:5 (Information flow control policy); FDP_IFF.1:2, FDP_IFF.1:3, FDP_IFF.1:4, and FDP_IFF.1:5 (Information flow control functions), FMT_MOF.1:3 (management of functions in TSF), FMT_SMR.2 and FMT_SMR.3 (security management roles). They cover the policies ***authenticated user SFP***, ***identified side channel user SFP***, ***authenticated gui user SFP***, and ***authenticated administrator SFP***.

7.1.4 SF_SM: Security management

SF_SM.1: The security management can be divided into three different roles: normal users do not have any rights, auditors (administrators with read-only rights) can view the configuration, and (normal) administrators can change the configuration. All users have the security attributes administrative role and password.

SF_SM.2: The configuration is divided into the following fields:

- System
- Connections
- Users
- Packet Filter
- HA
- Statistics
- Logging

SF_SM.3: Only administrators can change the password and security role of users, auditors and administrators. The auditors can view the settings. All security attributes for new users and administrators are set to a restrictive default. The user can change their passwords at the user web server.

SF_SM.4: Only administrators can change the timeouts for the administrator, user and side channel web server. The auditors can view the settings.

SF_SM.5: Only administrators can change the log details and authentication methods. The auditors can view the settings.

SF_SM.6: The attributes synchronized between HA peers are

1. user configuration (but not their blocked status);
2. network configuration;
3. relay configuration;
4. dns server configuration;
5. mail server configuration;
6. packet filter rule configuration;
7. http-proxy squid configuration;
8. virus scanner configuration;
9. audit configuration;
10. snmp server configuration;
11. igmpproxy configuration (on the PFL).

This Security Function addresses the SFRs: FIA_ATD.1 (user attribute definition); FMT_SMR.2 and FMT_SMR.3 (security management roles); FMT_MTD.1:1 and FMT_MTD.1:2 (management of TSF data); FMT_SMF.1 (specification of management functions); FMT_MSA.1:1, FMT_MSA.1:2, FMT_MSA.1:3, FMT_MSA.1:4, FMT_MSA.3:1, FMT_MSA.3:2, and FMT.MSA.3:3 (management of security attributes); FMT_MOF.1:1 and FMT_MOF.1:2 (management of functions in TSF).

7.1.5 SF_PT: Protection of the TSF

SF_PT.1: After a shutdown due to a failure or service discontinuity, the TOE does not reboot automatically, but requires an administrator interaction at the console.

For the high availability system this stop of service is not desired. Therefore a peer will take over the services of the failed system. The HA peers synchronize the attributes given in SF_SM.6.

SF_PT.2: In maintenance mode, system flags can be modified and therefore protected files can be manipulated. To allow an interactive session at the TOE only for the administrator at the console, all network packets (and Ethernet frames) are dropped silently in maintenance mode.

SF_PT.3: The TOE executes self tests regularly. The self tests consist of checking that (a configurable number) of processes are running, the file system usage is below a configurable threshold, and of tests for the file system consistency (file system permissions and flag settings). Administrators and auditors (the authorized users) can view the results of the self tests.

SF_PT.4: During normal operation the packet filter rules of the PFL cannot be modified. They are sealed when changing into normal operation mode.

This Security Function addresses the SFRs: FPT_SST.1EX (simple self test); FPT_RCV.2 (trusted recovery); FPT_TRC.1 (internal TOE TSF data replication consistency).

7.1.6 SF_PI: Patch installation

SF_PI.1: The TOE shall verify the integrity of patches using RSA signatures with a key size of 4096 bit according to PKCS#1, v2.1 using RSASSA-PKCS1-v1_5 and SHA-512 (default), SHA-384 or SHA-256. The patches are signed during the patch generation process and the signature is checked during patch installation.

SF_PI.2: During installation of the patch the current patch level is stored on the system in a defined way.

This Security Function addresses the SFR: FCS_COP.1 Cryptographic operation. It also adresses the SFRs FPT_UPD.1EX and FPT_UPD.2EX.

7.2 Self-Protection against Interference and Logical Tampering

The product takes the following self-protection measures, supplied by the TOE:

- The system is a two-tiered firewall. Both systems have to be overcome to gain unauthorised access from the external network on the internal network.
- On the ALG all connections are accepted by relay which are located in a reduced runtime environment (cages). An attacker has only limited capabilities.
- The ALG has a hardened kernel, some system calls are modified and deviate from their POSIX-conformant behaviour. This prevents attackers from escape out of the cages. The system calls are `chroot`, `mknod`, `ktrace`, and `strace`.
- All central processes of the ALG are controlled by the process master. In case of strange behaviour the process master can take actions.
- The ALG uses the BSD file system flags and runs at `securelevel=2`. The flags are used to mark most files as read-only and log files as append-only. The securelevel prevents changing the flags without going through single user mode.
- A reboot requires a manual interaction at the console. An attacker cannot modify the flags by going through single user mode.
- The PFL runs at `securelevel=3`. This means that the packet filter rules are immutable.

The following self-protection measures are supplied by the environment:

- The OpenBSD kernel uses a randomized stack top, a stack canary to detect stack overflow, and exclusive write or executable memory segments (`W^X`) to mitigate exploits.
- The OpenBSD applications use a randomized stack top, a stack canary to detect stack overflow, and exclusive write or executable memory segments (`W^X`) to mitigate exploits. Further, they use random library memory locations, random `mmap` and `malloc` function results, a read-only data segment `.rodata` for constant data to mitigate exploits.
- The OpenBSD daemons use either privilege revocation or privilege separation if they temporary need enhanced privileges.

- Both the OpenBSD kernel and the core OpenBSD applications use the functions `strlcat` and `strlcpy` to replace `strncat` and `strncpy` that guarantee to null-terminate the result.
- The OpenBSD daemons use `pledge` and`unveil` to prohibit system calls that the deamons do not need to call.
- The OpenBSD applies mitigations for the spectre/meltdown class of vulnerabilities. Especially, hyperthreading is disabled.

The measures together build up a multi-layered security barrier that results in a sufficient level of self-protection:

- The low level `strlcat` and `strlcpy` functions prohibit overwriting the allocated memory.
- The stack and memory protection mechanisms make it difficult to insert shell code.
- The privilege reduction functions inhibit a successful attacker to gain further privileges.

The TOE supplies a configuration GUI that check the parameters entered in the HTML forms. This helps to mitigate misconfiguration by administrators. It also gives a clear user interface for the administrators and auditors.

7.3 Self-Protection against Bypass

As the TOE is a firewall system, there can be no bypassing if it is installed properly. The assumption **A.SINGEN** reflects this.

8 Use of Cryptographic Functions

The use of cryptographic functions is summarised in table 7.

Table 7: Cryptographic functions

No	Purpose	Cryptographic Mechanism	Standard of Implementation	Key Size in Bits	Security Level above 100 Bits
Signature Verification					
1	Integrity	SHA-512 (default)	FIPS PUB 180-4	N/A	yes
2	Integrity	SHA-384 (alternative)	FIPS PUB 180-4	N/A	yes
3	Integrity	SHA-256 (alternative)	FIPS PUB 180-4	N/A	yes
4	Authentication	RSA using RSASSA-PKCS1-v1_5	PKCS#1, v2.1	|k| = 4096	yes

A Evaluation Methology for ALC_PAM

A.1 Objectives

The objective of this sub-activity is to determine if the patch release process is sufficiently documented.

A.2 Input

The evaluation evidence for this sub-activity is:

a) the ST;

b) the operational user guidance;

c) documentation of the patch release process of the developer;

d) developer evidence documentation;

A.3 Action ALC_PAM.1.1E

ALC_PAM.1.1C ***The developer's patch management policies shall describe what is the criteria used for the decision that a patch has to be released.***

ALC_PAM.1-1 The evaluator shall check for the definition of criteria and check for the implementation as a policy. Example of a list of criteria:

- Complexity of backports
- Operational stability, development teams is able to estimate effect for operational stability
- security impact
- customer impact (i.e. practical problems, theoretical problems)
- timely impact, i.e. customer expect patches each quarter of a year, i.e. also minor security problems have to be fixed

ALC_PAM.1-2 The evaluator shall check the status of the implementation of the policies for patch releases and verify if the policies for patch releases have the same level of detail as other developer evidences provided for ALC.

ALC_PAM.1-3 The evaluator shall verify if the following mandatory policy content was implemented as policy:

- responsible roles for the final decision to release a patch
- unique label for each patch to identify all release items

ALC_PAM.1.2C ***The Security Target shall contain the estimated end-of-life of the TOE.***

ALC_PAM.1-4 The evaluator shall check for estimated TOE end-of-life information in the ST and for estimated TOE end-of-life information in user information material, i.e. the guidance, release notes, product (support) website.

ALC_PAM.1.3C ***The developer's patch management policies shall describe how to self-assess the security relevance of a patch (i.e. Security Impact Analysis Report, S-IAR) and which procedures have to apply due to which assessment result.***

ALC_PAM.1-5 The evaluator shall check if at least two assessment result categories were defined for patch management. For example:

- Category 1: no patch required

- Category 2: patch is required

ALC_PAM.1.4C ***The developer's patch management policies shall describe how to update the evidence documentation used in the base evaluation.***

ALC_PAM.1-6 The evaluator shall check if the patch management policies describe how to update the evidence documentation in a consistent way with the evaluation assurance level.

ALC_PAM.1.5C ***The developer's patch management policies shall describe how unhandled (potential) flaws are documented.***

ALC_PAM.1-7 The evaluator shall check the status of the implementation of the unhandled flaw documentation policy.

ALC_PAM.1.6C ***The developer's patch management policies shall describe which organizational role (or group) is responsible for the patch development.***

ALC_PAM.1-8 The evaluator shall check the organizational definitions and responsibilities of all roles involved in the patch development process. Examples for definitions of patch development responsibilities:

- patch development tasks as part of RACI matrix
- patch development tasks as function of a product development team

ALC_PAM.1.7C ***The developer's patch management policies shall describe which policies have to be applied until the end of life of the TOE during the patch management.***

ALC_PAM.1-9 The evaluator shall check for the implementation of internal policies that have to be applied during TOE maintenance. Examples for policies regarding 3rd Party Libraries:

- update only libraries that are still supported as well
- backport latest changes to used library version
- upgrade to latest library version

ALC_PAM.1.8C ***Each tool used for the patch management shall be documented.***

ALC_PAM.1-10 The evaluator shall check the list of tools the developer uses for patch management.

ALC_PAM.1.9C ***The patch management policies shall describe the mandatory structure and content of the S-IAR.***

ALC_PAM.1-11 The evaluator shall check the format of the S-IAR used by the developer. Mandatory elements of the S-IAR are:

- Description how to use bug tracker information for the S-IAR
- Security relevance criteria: e.g. remote execution, only product type specific
- Category criteria: e.g. CWE (common weakness enumeration)

ALC_PAM.1.10C ***Each type of documentation used to record decisions in the patch management process shall be documented.***

ALC_PAM.1-12 The evaluator shall check if the patch management policies describe how to record decisions.

ALC_PAM.1.11C ***The patch management policies shall describe the mandatory content of patch release notes.***

ALC_PAM.1-13 The evaluator shall check if the patch management policies contain the elements that shall be mandatory in a developer's patch release notes.

ALC_PAM.1.12C ***The patch management policies shall describe the mandatory content for the guidance documents which have to be fulfilled to support the installation of the patch.***

ALC_PAM.1-14 The evaluator shall check the developer's patch management policies for release note or update guidance requirements (e.g. checklist for steps to describe during patch installation).

ALC_PAM.1.13C ***The patch management policies shall describe the mandatory procedures during patch release.***

ALC_PAM.1-15 The evaluator shall check the developer's patch management policies for mandatory patch release procedures. Examples:

- procedure steps for (patch) release: Build → QA test → HW integration test → Release
- process definition should contain the failure of test/validation steps and how to handle these cases

ALC_PAM.1.14C ***The patch management policies shall contain rules in which case the evaluation facility has to perform additional tests before the patch is released.***

ALC_PAM.1-16 The evaluator shall check the developer's patch management policies for rules that require testing of the evaluation facility.

- e.g. ruleset for different acting roles in the (patch) release procedure
- relevant roles: development, QA department, product owner, etc.
- hardware release decisions that require software updates in the drivers for the new hardware.

ALC_PAM.1.15C ***The patch management policies shall describe how each of the patch management Security Objectives for the Operational Environment are fulfilled until the end of life of the TOE.***

ALC_PAM.1-17 The evaluator shall check the developer's patch management policies for a description of how the Patch Management Security Objectives are fulfilled.

ALC_PAM.1-18 The evaluator shall verify if the patch management processes address the following requirements:

- How the cryptographic keys involved in signing and/or distributing patches are generated and managed during its entire life-cycle so they have enough strength to protect the authenticity of the updates.
  - How the cryptographic keys are created
  - How the cryptographic keys are securely stored
  - The process for revocation and loading of a new cryptographic key if it is compromised
  - How the cryptographic keys are destroyed or archived at the end-of-life of the product
- The process for approving, signing and releasing new updates in a secure and audited environment.
  - Who approves the releasing of updates
  - Who can access the cryptographic keys used for signing updates
  - How the update is moved from the development environment to the signing environment so that it is not tampered
  - How this process generates logs
  - How this logs are audited
- How the user is notified of the availability of a new patch due to a security issue:
  - Through email
  - Through automatic checks to a website handled by the product
- How the patches are made available and securely distributed to the end user
  - Uploaded to a website by the developer and automatically downloaded by the TOE by using an appropriate and declared security protocol
- Sent to the end-user using delivery services and providing installation instructions where administrator rights must be implemented using password/authentication codes and/or cryptographic authentication techniques

A.4 Implied evaluator action ALC_PAM.1.2D

ALC_PAM.1.2D ***The developer shall self-assess and confirm the application of existing policies on a regular basis saving records of its application.***

ALC_PAM.1-19 The evaluator shall verify evidences of developer's self-assessment procedures.

ALC_PAM.1-20 The evaluator shall check if results or evidences for the self-assessment can be presented. For example:

- publication of developer self-declaration with reference to product certification ID
- internal (or external) audit report, in general annual audit

ALC_PAM.1-21 The evaluator shall check if existing unhandled flaw documentation exists and if these fulfil the policy requirements.

ALC_PAM.1-22 The evaluator shall check if decisions in the patch management process were documented.

ALC_PAM.1-23 The evaluator shall check the patch release notes for the content elements required by the patch management policies.

ALC_PAM.1-24 The evaluator shall select and examine a sample of evidence covering each type of relevant event (e.g. signing logs, approval of updates, S-IAR, fulfilled checklists, bug tracker evidence...) to confirm that all operations of the patch management policies and procedures are carried out in line with the documentation. The evaluator may choose to sample the evidence.

- For guidance on sampling see ISO/IEC 18045, A.2, Sampling.
- Further confidence in the correct operation of the patch management policies and procedures may be established by means of interviews with selected development staff. Note that such interviews should complement rather than replace the examination of documentary evidence, and may not be necessary if the documentary evidence alone satisfies the requirement.
- The evaluator may visit the development site in support of this activity.
- For guidance on site visits see ISO/IEC 18045, A.4, Site Visits.

A.5 Implied evaluator action ALC_PAM.1.3D

ALC_PAM.1.3D ***The developer shall provide security patches using the defined policies and procedures at least until the estimated end-of-life of the TOE.***

ALC_PAM.1-25 The evaluator shall examine aspects of the patch management procedure to determine that the patch management procedures are being used.

- In addition to examination of the procedures themselves, the evaluator seeks some assurance that they are applied in practise. Some possible approaches are:
  - a visit to the development site(s) where practical application of the procedures may be observed;
  - observing that the process is applied in practise when the evaluator obtains new updates solving the vulnerabilities found during the Vulnerability Analysis.
- If a Site Visit is already included in the evaluation plan, the evaluator shall apply option (a) to check that the processes are applied in practice.
- For guidance on site visits see A.4, Site Visits.

B References

[1] Common Criteria. Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components, Version 3.1 Revision 5. Technical report, Common Criteria, April 2017. CCMB-2017-04-002.

[2] Common Criteria for Information Technology Security Evaluation. Part 1: Introduction and general model. Version 3.1, Revision 5, September 2017.

[3] Common Criteria for Information Technology Security Evaluation. Part 2: Security functional components. Version 3.1, Revision 5, September 2017.

[4] Common Criteria for Information Technology Security Evaluation. Part 3: Security assurance components. Version 3.1, Revision 5, September 2017.

[5] T. Ylonen and C. Lonvick. The Secure Shell (SSH) Protocol Architecture. RFC 4251, Internet Engineering Task Force, January 2006. `http://www.ietf.org/rfc/rfc4251.txt`.

C Acronyms

ALG Application Level Gateway

AWC Advanced Web Categories

BGP Border Gateway Protocol

BSD Berkeley Software Distribution

CARP Common Address Redundancy Protocol

DMZ Demilitarized Zone

DNS Domain Name Server

FTP File Transfer Protocol

HTTPS Hypertext Transfer Protocol Secure

ICMP Internet Control Message Protocol

IGMP Internet Group Management Protocol

IMAP Internet Message Access Protocol

IMAPS IMAP over SSL

IP Internet Protocol

LDAP Lightweight Directory Access Protocol

MSSQL Microsoft SQL Server relational database management system

MTA Mail Transfer Agent

MySQL a relational database management system

NTP Network Time Protocol

OPC UA Open Platform Communications Unified Architecture

OWASP Open Web Application Security Project

PAP packet filter - application level gateway - packet filter

PFL Packet Filter

POP Post Office Protocol

POP3 Post Office Protocol, version 3

PostgreSQL PostgreSQL object-relational database management system

PPTP Point-to-Point Tunneling Protocol

RDP Remote Desktop Protocol

RTP Real-time Transport Protocol

SIP Session Initiation Protocol

SMB Server Message Block

SMTP Simple Mail Transfer Protocol

SNMP Simple Network Management Protocol

SOAP Simple Object Access Protocol

SSH Secure Shell, siehe [5], `http://www.openssh.org`

SSL Secure Sockets Layer

TCP Transmission Control Protocol

TLS Transport Layer Security

UDP User Datagram Protocol

WAF Web Application Firewall

WSDL Web Service Description Language