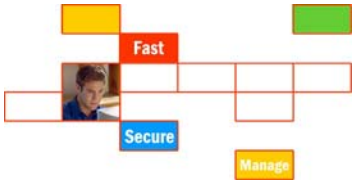
Microsoft
Internet Security
and Acceleration
Server




# ISA Common Criteria Evaluation

## Security Target

*Internet Security and Acceleration Server Team*


| | |
|---|---|
| Author: | Stephan Slabihoud, TÜViT GmbH<br>Anthony Blumfield, Microsoft Corp. |
| Category: | CC Evaluation |
| Status: | Final |
| Version: | 1.2 |
| Revision: | 1 |
| Last Saved: | 8/12/2003 |
| File Name: | MS_ISA_ST_1.2.doc |



### Abstract

This document describes the ST (Security Target) of ISA server 2000 Common Criteria Certification that is the basis for the ISA Server 2000 CC evaluation.



### Keywords

CC, ST, Common Criteria, Firewall, Security Target


### Revision History

| Date | Version | Author | Edit |
|---|---|---|---|
| 10-Mar-03 | 0.1 | Stephan Slabihoud | Created |
| 13-Mar-03 | 0.2 | Stephan Slabihoud | some feedback added |
| 19-Mar-03 | 0.3 | Stephan Slabihoud | new structure concerning information flow control and identification and authentication |
| 25-Mar-03 | 0.4 | Stephan Slabihoud | first draft for the BSI |
| 27-Mar-03 | 0.5 | Stephan Slabihoud | added SFRs and Rationale, first complete draft for release |
| 2-Apr-03 | 0.6 | Stephan Slabihoud | feedback added, FCS_COP.1 added |
| 8-Apr-03 | 0.7 | Stephan Slabihoud | some layout changes |
| 22-Apr-03 | 0.8 | Dr. Patrick Bödeker | feedback from BSI added, figure for TOE demarcation added, FPT_RVM.1 included |
| 28-Apr-03 | 0.9 | Stephan Slabihoud | few corrections |
| 07-May-03 | 1.0 | Stephan Slabihoud | final version |
| 22-May-03 | 1.1 | Stephan Slabihoud | FCS_COP.1 justification more detailed |
| 12-Aug-03 | 1.2 | Stephan Slabihoud | changes according Final Interpretations 058 |

This page intentionally left blank

## Table of Contents

## List of Tables

**Page**



## List of Figures

**Page**

# 1 Introduction

This chapter contains document management and overview information. The Security Target (ST) identification provides the labeling and descriptive information necessary to identify, catalogue, register, and cross-reference a ST. The ST overview summarizes the ST in narrative form and provides sufficient information for a potential user to determine whether the ST is of interest. The overview can also be used as a standalone abstract for ST catalogues and registers.

## 1.1 Identification

The title of this ST is “ISA Common Criteria Evaluation - Security Target”, Version 1.2, Revision 1, dated 13.08.2003.

The Target of Evaluation (TOE) is a dedicated firewall called “Microsoft Internet and Acceleration Server 2000 – Standard Edition”. Its software version is MS ISA 2000 Service Pack 1 with Feature Pack 1.

The Security Target is built in accordance with Common Criteria V2.1 with Final Interpretations [CC].

## 1.2 Overview

This chapter presents a general overview of the Microsoft Internet Security and Acceleration Server 2000[1].

ISA Server is a firewall that helps to provide secure Internet connectivity. ISA Server is an integrated solution optimized for application-layer defense, stateful packet inspection (SPI), and secure web publishing.

ISA Server can be installed as a dedicated (software) firewall that runs on a Windows 2000 Server operating system. It acts as the secure gateway to the Internet for internal clients and protects communication between internal computers and the Internet.

As a multilayered firewall, ISA Server provides security at different levels. IP packet filtering provides security by inspecting individual packets passing through the firewall. Application-level filtering allows ISA Server to intelligently inspect and secure popular protocols (such as HTTP, FTP and others). ISA Server also performs dynamic-filtering using stateful packet inspection (SPI) to open communication ports only when requested by clients and close them when they are no longer needed. This reduces the number of communication ports that are statically open to inbound connections.

With ISA Server’s filtering capabilities, it is possible to create filters that allow or deny traffic on the packet layer and with data-aware filters to determine if packets should be accepted, rejected, redirected, or modified. ISA Server has built in identification and authentication

[1] short: „ISA Server“

capabilities which can be configured separately for incoming and outgoing requests. The firewall features detailed security and access logs. The log files can be configured and enabled for packet and application filters. They are human readable and can be reviewed with additional tools.

## 1.3 Related Documents

Related documents are:

- Microsoft Internet Security and Acceleration Server 2000 manual [MSISA]
- Common Criteria for Information Technology Security Evaluation [CC]

The main chapters of the ST are the TOE Description, TOE Security Environment, Security Objectives, IT Security Requirements, TOE Summary Specification, PP Claims and Rationale.

Chapter 2, the TOE Description, provides general information about the TOE, serves as an aid to understanding the TOE's security requirements, and provides context for the ST's evaluation.

The TOE Security Environment in Chapter 3 describes security aspects of the environment in which the TOE is to be used and the manner in which it is to be employed. The TOE security environment includes:

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

b) Threats relevant to secure TOE operation

Chapter 4 contains the security objectives that reflect the stated intent of the ST. The objectives define how the TOE will counter identified threats and how it will cover identified organizational security policies and assumptions. Each security objective is categorized as being for the TOE or for the environment.

Chapter 5 contains the applicable security requirements taken from the Common Criteria, with appropriate refinements. The requirements are provided in separate subsections for the TOE and its environment. The IT security requirements are subdivided as follows:

a) TOE Security Functional Requirements

b) TOE Security Assurance Requirements

The TOE summary specification in chapter 6 defines the security functions, the assurance measures and the security function policies are defined in the ST as property of this specific TOE.

The security target does not claim for compliance with any existing protection profile (see Chapter 7).

The Rationale in Chapter 8 presents evidence that the ST is a complete set of requirements and that the TOE provides an effective set of IT security countermeasures within the security environment.

The rationale is divided in three main parts:

- a security objectives rationale demonstrates that the stated security objectives are traceable to all of the aspects identified in the TOE security environment and are suitable to cover them,
- a security requirements rationale demonstrates that the security requirements (TOE and environment) are traceable to the security objectives and are suitable to meet them, and
- the TOE summary specification rationale consists of a TOE security functions rationale and an assurance measures rationale.

A glossary of acronyms and terms used in the ST as well as references is provided in the Appendix in chapter 9.

## 1.4 Common Criteria Conformance

This ST has been built with Common Criteria (CC) Version 2.1 (ISO/IEC 15408 Evaluation Criteria for Information Technology Security; Part 1: Introduction and general model, Part 2: Security functional requirements, and Part 3: Security assurance requirements). The TOE itself is conformant with Common Criteria Version 2.1, part 2 and part 3.

This security target does not claim for compliance with any existing protection profile.

The assurance level for the TOE is **EAL 2**.

The strength of function is **SOF Basic**.

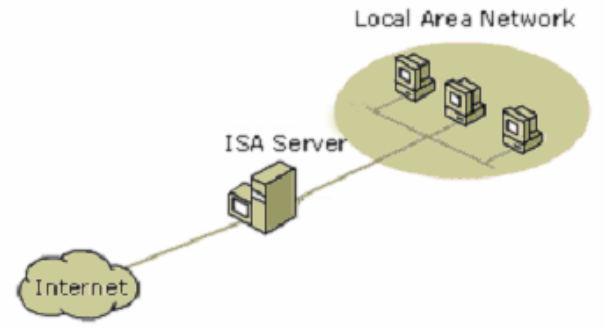
# 2 TOE Description

The TOE description helps to understand the specific security environment and the security policy. In this context the assets, threats, security objectives and security functional requirements can be employed. After some general remarks about ISA Server in chapter 2.1, chapter 2.2 presents a more detailed description of the TOE as it refers to this particular TOE implementation.

## 2.1 ISA Server - Some general remarks

ISA Server integrates firewall, cache features, reporting, alerting and an administration GUI to secure a network, to improve its performance and to maintenance the functionalities (see Figure 2.1 and Figure 2.2).

**Figure 2.1 – ISA Server environment**




There are two versions of ISA Server available: Standard Edition (single machine support only) and Enterprise Edition (can be member of a firewall cluster).

The Enterprise edition is designed for large-scale deployments with high-volume Internet traffic environments. It supports multi-server arrays with centralized management as well as enterprise-level and array-level security policy. Enterprise Edition has no hardware limits. ISA Server Standard Edition shares the feature set of Enterprise Edition, but it is intended for small businesses, workgroups, and departmental environments. Standard Edition provides local policy only, and supports up to four processors.

For the Standard Edition security policy configuration data is stored in the local Windows registry, for the Enterprise Edition security policy configuration data can be optionally centrally stored in the Active Directory database. Both versions - Standard and Enterprise - can be treated the same way because the storage of policy configuration data is not part of the evaluation (Windows Registry and Active Directory are outside the scope of the TOE) and also scalability is not part of the evaluation. As a result, the Standard Edition with local administration and without Active Directory integration has been chosen as TOE.

ISA Server can be installed in one out of three modes: firewall, cache, or integrated.

The firewall mode allows secure network communication by configuring rules that control communication for all Internet protocols. ISA Server routes requests and responses between the Internet and the computers placed in the local area network after determining if the communication is allowed. It is also possible to publish services of internal servers.

In cache mode network performance is improved and bandwidth is saved by storing frequently-requested objects in a cache so that the objects can be quickly accessed by clients directly from the ISA Server computer, rather than from the Internet. Protocols supported in cache mode are HTTP, HTTPS and FTP[2].

In integrated mode all the features of firewall and cache mode are available.

For evaluation the firewall mode is chosen because the caching functionality does not include any security related features.

ISA Server blocks unauthorized Internet users from accessing the local network. It examines incoming data and blocks packets that do not meet the predefined filter criteria. It can also be configured to listen for requests from internal clients for objects on the Internet. These requests can be restricted to selected computers and/or users (using authentication).

ISA Server works at various communication layers to protect the internal network. At packet layer ISA Server can control data on the internal and external interface:

- Inbound traffic is evaluated before it can reach any resource in the internal network. If the data is allowed to pass the packet filtering layer, it is passed to the Firewall and/or Web Proxy service, where ISA Server rules are processed to determine if the request should be serviced.
- Outbound traffic is checked against IP packet filter rules, site and content rules, and protocol rules to determine if access is allowed. A request is allowed only if both a protocol rule and a site and content rule each allow the request and if there is no rule that explicitly denies the request.

## 2.2 TOE overview

ISA Server is a dedicated firewall that acts as the secure gateway to the Internet for internal computers. ISA Server protects all communication between internal computers and the Internet.

For administration ISA Server includes graphical taskpads and wizards. These simplify navigation and configuration for common tasks. These features are embedded in the MMC[3] and do not belong to the TOE. They are implemented in the environment.

The operation system Windows 2000 maintains security attributes for all administrators. Windows 2000 stores the identification and authentication data for all known administrators and maintains a method of associating human users with the authorized administrator role.

---

[2] exact: FTP over HTTP, regular FTP is not cached

[3] Microsoft Management Console

The TOE itself offers no additional identification and authentication methods for firewall administrators.

The next chapters describe the physical scope and boundary and the basic functionalities of the TOE.

### 2.2.1 Physical scope and boundary

The TOE configuration consists of:

- the software package “Microsoft Internet and Acceleration Server 2000 - Standard Edition” with service pack 1 and feature pack 1 installed running on a single machine in firewall mode

The TOE is delivered on CD-ROM and running on an

- evaluated Windows 2000 operating system with service pack 3 installed (same installation that has been used for Windows 2000 Common Criteria EAL 4+ Evaluation; Validation Report Number CCEVS-VR-02-0025),
- HP/Compaq ProLiant ML330 G2 hardware as the same servers that have been used for Windows 2000 Common Criteria Evaluation.

### 2.2.2 Logical scope and boundary

The logical scope and boundary of the TOE is subdivided into the following major functions of the TOE:

- Identification and Authentication,
- Filtering (Information Flow Control) and
- Audit Generation.

#### 2.2.2.1 Identification and Authentication

The access policy[4] and publishing rules[5] of the TOE can be configured to allow or deny a set of computers[6] or a group of users to access specific servers. If the rule applies specifically to users, the TOE checks how the user should be authenticated. It is possible to configure incoming and outgoing Web request settings so that users must always be authenticated by the TOE. This ensures that requests are allowed only if the user making the request is authenticated. It is possible to choose between different authentication methods also separately for incoming and outgoing requests.

---

[4] see chapter 2.2.2.2 and glossary

[5] see chapter 2.2.2.2 and glossary

[6] „client address set“ or „client set“

ISA Server supports a widely range of authentication methods:

- Basic authentication

  The standard method of authentication for Hypertext Transfer Protocol (HTTP) transmissions is basic authentication. Basic authentication sends and receives user information as text characters. No encryption is used with basic authentication.

- Digest authentication

  Digest authentication offers the same features as basic authentication but uses hashing before transmitting the user data to the server. So it is not possible to decipher the original data from the hash. To prevent capturing and using of the password by a third user additional information is added to the password before hashing. So it is possible to authenticate the user and the user's computer and the domain.

- Integrated Windows authentication

  Integrated Windows authentication does not send the user name and password across the network. It is possible to use either the Kerberos V5[7] authentication protocol or NTLM challenge/response authentication protocol.

- RSA SecurID Authentication

  ISA Server allows strong, RSA SecurID authentication for Web servers and servers running OWA at ISA Server. SecurID authentication is based on something you know (a password or personal identification number) and something you have (an authenticator).

- Client certificates and server certificates

  ISA Server allows using SSL security features for authentication. Certification is used in two ways, when a client requests an object from a server:

  - The server authenticates itself by sending a server certificate to the client.
  - The server requests that the client authenticate itself. In this case, the client must present an appropriate client certificate to the server.

  Server certificates contain identifying information about the server. Client certificates usually contain identifying information about the user and the organization that issued the certificate.

Delegation of authentication helps increase security by enabling ISA Server to authenticate Internet clients instead of passing the pre-authentication to the published server. This delegation also eliminates multiple login prompts. Delegation is possible with SecurID and Basic (user name and password) authentication and can be enabled for each Web publishing rule.

The TOE is configured that it supports Basic authentication only, which is the standard method of authentication for Hypertext Transfer Protocol (HTTP) transmissions. Basic

[7] http://www.ietf.org/rfc/rfc1510.txt

authentication for web requests can be secured using an SSL channel, so user identification and authentication credentials are encrypted during transmission.

#### 2.2.2.2 Filtering (packet and application level filtering)

The TOE combines several security mechanisms to enforce the security policies at different network layers: a rule base for incoming and outgoing requests, application filters, and system security configuration options.

The TOE Server distinguishes between the following filters and rules:

**Packet filters**

Packet filters control the flow of IP packets to and from the TOE. All packets on the external interface are dropped unless they are explicitly allowed, either statically by “IP packet filters” or dynamically by publishing rules (e.g. Web publishing). All packets on the internal interface are dropped unless they are explicitly allowed statically by “IP packet filters” and the access policy (“protocol rules” and “site and content rules”).

**IP packet filters** allow or block packets destined for specific computers on the internal network. It is possible to configure two types of static IP packet filters: allow filters and block filters. Allow filters are exception filters — all packet types are blocked except for those that are specified. When no packet filter is activated for a specific port, then the service cannot listen on that port unless the port is opened dynamically.

Block filters close the specified ports. They are used to further define the traffic allowed through the TOE. Allow filters that open a specific port for all clients can be restricted by block filters for specific clients.

**Protocol rules** define which protocols can be used for communication between the local network and the Internet. Protocol rules are processed at the application level. For example, a protocol rule might allow clients to use the HTTP protocol. The TOE includes a list of preconfigured protocol definitions, including the Internet protocols which are most widely used. It is possible to add additional protocols.

**Site and content rules** define what content on which Internet sites can be accessed by clients behind the TOE. Site and content rules are processed at the application level. For example, a site and content rule might allow clients to access any destination on the Internet.

IP packet filters open the ports statically. Access policy rules (and also publishing rules) open the ports dynamically (as a request arrives).

**Server publishing**

Server publishing allows any computer on an internal network to publish to the Internet. Security is not compromised because all incoming requests and outgoing responses pass through the TOE. When a server is published by the TOE, the IP addresses that are published are actually the IP addresses of the TOE. Users who request objects think that they are communicating with the TOE — whose name or IP address they specify when requesting the object — while they are actually requesting the information from the actual publishing server.

**Web publishing**

The TOE uses Web publishing rules to relieve the concerns associated with publishing Web content to the Internet without compromising internal network security. Web publishing rules determine how the TOE should intercept incoming requests for Hypertext Transfer Protocol (HTTP) objects on an internal Web server and how the TOE should respond on behalf of the Web server. Requests are denied or forwarded downstream to an internal Web server, located behind the TOE.

**Application filters**

The TOE offers a set of application filters that can access the data associated with a session within the Firewall service. Application filters work with some or all application-level protocols (see IP packet filters: protocol rules). An application filter can perform protocol-specific tasks. In addition, application filters may also include protocol definitions. They can be included when ISA Server is installed, or they can be installed later.

The application filter evaluated with the TOE is the RPC filter that enables publishing of Exchange RPC servers.

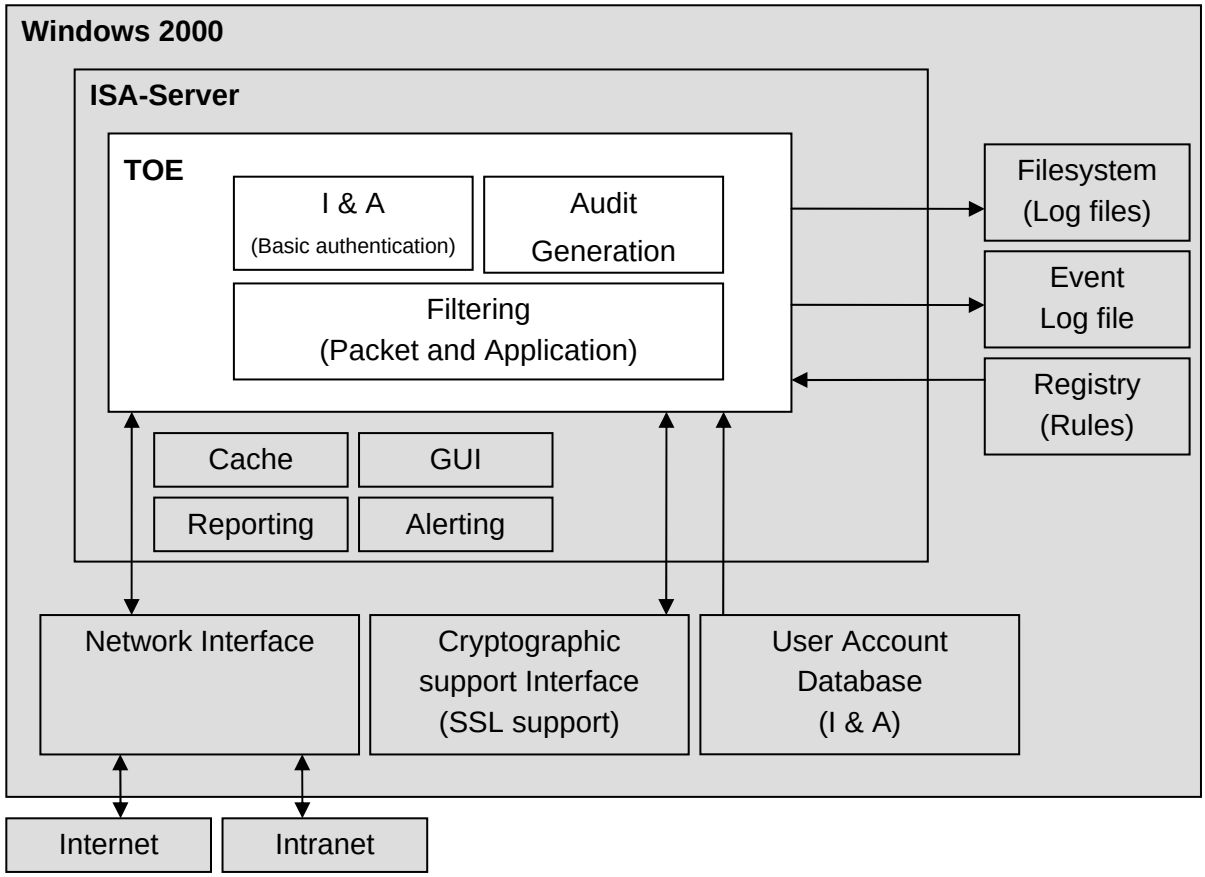
#### 2.2.2.3 Audit Generation

The TOE features detailed security and access logs (packet filter log file, firewall service log file and web proxy log file), which can be generated in standard data formats like W3C[8]. The log files are stored locally in human readable text files[9]. It is possible to change the destination folder the log files are created in. The TOE offers no additional access protection for the log files. Access protection is granted by the filesystem of the underlying operation system.

[8] http://www.w3.org/Daemon/User/Config/Logging.html

[9] ISA Server can store log files locally or remote in a database. The ISA Server reporting system centralizes the logs, collecting data from all the servers into a single report. These features are not part of the TOE.

**Figure 2.2 – TOE demarcation**




For better understanding the boundaries of the TOE are summarized in Figure 2.2. It shows the TOE with its three main security functionalities: filtering, identification & authentication and audit generation, the additional features of the ISA-Server which are not part of the evaluation: cache, GUI, reporting and alerting, and the used functionalities of the underlying operating system Windows 2000. The arrows show the interfaces between the TOE and the operating system, the arrowheads show the direction of information flow. The TOE uses the file system and the event log file to store the audit data. From the registry the filter rules are read. The user account database provides the information required by the I&A functionality of the TOE. The cryptographic support interface supports the SSL functionality. The network interface is needed for transmitting data to the internet respectively the intranet.

# 3 TOE Security Environment

This chapter aims to clarify the security problems that the ISA Server is intended to solve, by describing any assumptions about the security aspects of the environment and/or of the manner in which the TOE is intended to be used and any known or assumed threats to the assets against which protection within the TOE or its environment is required. This is done considering the attack potential of attackers aiming to discover exploitable vulnerabilities to be low.

## 3.1 Assumptions

Table 3.1 lists the TOE Secure Usage Assumptions for the IT environment and intended usage.

**Table 3.1 – Assumptions for the IT Environment and intended usage**

| # | Assumption Name | Description |
|---|---|---|
| 1 | A.PHYSEC | The TOE is physically secure. Only authorized personal has physical access to the TOE. |
| 2 | A.GENPUR | The TOE stores and executes security-relevant applications only. It stores only data required for its secure operation. |
| 3 | A.NOEVIL | Authorized administrators are non-hostile and follow all administrator guidance. |
| 4 | A.SINGEN | Information can not flow among the internal and external networks unless it passes through the TOE. |
| 5 | A.DIRECT | The TOE is available to authorized administrators only. Personal who has physical access to the TOE and can log in the operating system is assumed to act as an authorized TOE administrator. |
| 6 | A.SECINST | Required certificates and user identities are installed using a confidential path. |
| 7 | A.OS | The operating system implements following functions which are used by the TOE security functions:<br>reliable time stamp (log file audit), file protection (for log file access protection), tools for audit review, and administration access control. |

## 3.2 Organisational Security Policies

Security policies to be fulfilled by the TOE are defined in Table 3.2 below.

**Table 3.2 – Security Policies addressed by the TOE**

| # | Policy Name | Description |
|---|---|---|
| 1 | P.AUDACC | Persons must be accountable for the actions that they conduct. Therefore audit records must contain sufficient information to prevent an attacker to escape detection. |

## 3.3 Threats

Threats to the TOE are defined in Table 3.3 below. The asset under attack is the information transiting the TOE. In general, the threat agent (attacker) includes, but is not limited to:

1) not authorized persons or

2) external IT entities not authorized to use the TOE itself.

**Table 3.3 – Threats**

| # | Threat | Description |
|---|---|---|
| 1 | T.NOAUTH | An attacker may attempt to bypass the security of the TOE so as to access and use security functions and/or non-security functions provided by the TOE. |
| 2 | T.ASPOOF | An attacker may carry out spoofing in information flows mediated by the TOE between clients and servers located on internal and external networks governed by the TOE, by using a spoofed source address to hide his identity. |
| 3 | T.MEDIAT | An attacker may send impermissible information through the TOE, which results in the exploitation of resources on the internal network and gathering of information he is not authorized for. |
| 4 | T.OLDINF | Because of a flaw in the TOE functioning, an attacker may gather residual information from a previous information flow or internal TOE data by monitoring the padding of the information flows from the TOE. |
| 5 | T.AUDFUL | An attacker may cause audit records to be lost or prevent future records from being recorded by taking actions to exhaust audit storage capacity, thus masking an attackers actions. |

# 4 Security Objectives

## 4.1 Security Objectives for the TOE

TOE security objectives are defined in Table 4.1, below.

**Table 4.1 – Security Objectives for the TOE**

| # | Objective | Description |
|---|---|---|
| 1 | O.IDAUTH | The TOE must uniquely identify and authenticate the claimed identity of all users, before granting a user access to TOE functions or, for certain specified services, to a connected network. |
| 2 | O.MEDIAT | The TOE must mediate the flow of all information from users on a connected network to users on another connected network, and must ensure that residual information from a previous information flow is not transmitted in any way. |
| 3 | O.SECSTA | Upon initial start-up of the TOE or recovery from an interruption in TOE service, the TOE must not compromise its resources or those of any connected network. |
| 4 | O.AUDREC | The TOE must provide a means to record a readable audit trail of security-related events, with accurate dates and times and makes sure that no records are left because of not enough storage capacity. |
| 5 | O.ACCOUN | The TOE must provide user accountability for information flows through the TOE. |

## 4.2 Security Objectives for the Environment

Table 4.2 lists security objectives for the IT-Environment.

Table 4.2 – Security Objectives for the IT-Environment

| # | Objective Name | Objective Description |
|---|---|---|
| 1 | OE.PHYSEC | The TOE should be physically secure. |
| 2 | OE.GENPUR | The TOE should store and execute security-relevant applications only and should store only data required for its secure operation. |
| 3 | OE.NOEVIL | Authorized administrators should be non-hostile and should follow all administrator guidance. |
| 4 | OE.SINGEN | Information should not flow among the internal and external networks unless it passes through the TOE |
| 5 | OE.DIRECT | The TOE should be available to authorized administrators only. |
| 6 | OE.SECINST | The required certificates and user identities should be stored using a confidential path. |

| | | |
|---|---|---|
| 7 | OE.OS | The operating system should implement following functions:<br>reliable time stamp (log file audit), file protection (for log file access protection), tools for audit review, cryptographic support (for SSL encryption), and administration access control. |

There are no security objectives for the non-IT environment.

# 5 IT Security Requirements

## 5.1 Introduction

This chapter defines the TOE security functional requirements and assurance requirements. All requirements are taken from the CC Parts 2 and 3. Selections, assignments, and refinements performed are indicated by *italics* and stated which operation is used.

## 5.2 TOE Security Functional Requirements

This chapter defines the TOE security functional requirements. A list of the requirements is provided in Table 5.1. The full text of the security functional requirements is contained below. Certain security functional requirements have multiple iterations in the text. Iterations are indicated by the use of parentheses "()" in the component identification and by parentheses "()" and an abbreviation in the component name.

**Table 5.1 – TOE Security Functional Requirements**

| # | Functional Requirement | Title | Dependencies |
|---|---|---|---|
| **Audit Generation** | | | |
| 1 | FAU_GEN.1 | Audit data generation | FPT_STM.1 |
| 2 | FAU_SAR.1 | Audit review | FAU_GEN.1 |
| 3 | FAU_STG.3 | Action in case of possible audit data loss | FAU_STG.1 |
| **Identification and Authentication** | | | |
| 4 | FIA_AFL.1 | Authentication failure handling | FIA_UAU.1 |
| 5 | FIA_ATD.1 | User attribute definition | |
| 6 | FIA_UID.2 | User identification before any action | |
| 7 | FIA_UAU.2 | User authentication before any action | FIA_UID.1 |
| **Information Flow Control** | | | |
| 8 | FDP_IFC.1 (1) | Subset information flow control (1) - UNAUTHENTICATED SFP | FDP_IFF.1 (1) |
| 9 | FDP_IFC.1 (2) | Subset information flow control (2) - UNAUTHENTICATED_APPL SFP | FDP_IFF.1 (2) |
| 10 | FDP_IFC.1 (3) | Subset information flow control (3) - AUTHENTICATED SFP | FDP_IFF.1 (3) |
| 11 | FDP_IFF.1 (1) | Simple security attributes (1) - UNAUTHENTICATED SFP | FDP_IFC.1 (1) FMT_MSA.3 |
| 12 | FDP_IFF.1 (2) | Simple security attributes (2) - UNAUTHENTICATED_APPL SFP | FDP_IFC.1 (2) FMT_MSA.3 |
| 13 | FDP_IFF.1 (3) | Simple security attributes (3) - | FDP_IFC.1 (3) |

| | | | |
|---|---|---|---|
| | | AUTHENTICATED SFP | FMT_MSA.3 |
| 14 | FDP_RIP.1 | Subset residual information protection | |
| 15 | FMT_MSA.3 | Static attribute initialization | FMT_MSA.1<br>FMT_SMR.1 |
| 16 | FPT_RVM.1 | Non-bypassability of the TSP | |

Note: FPT_STM.1, FAU_STG.1, FMT_MSA.1, and FMT_SMR.1 are considered in the IT environment (see chapter 8.2.4).

## 5.2.1 Class FAU – Security audit

**FAU_GEN.1 Audit data generation**

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

a) Start-up and shutdown of the audit functions;

b) All auditable events for the [*selection: not specified*] level of audit; and

c) [*assignment: the events specified in* Table 5.2].

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

a) Date and time of the event, type of event, subjects identities, 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: information specified in column four of* Table 5.2].

**Table 5.2 – Auditable Events**

| Functional Component | Level | Auditable Event | Additional Audit Record Contents |
|---|---|---|---|
| FIA_UID.2 | basic | All use of the user identification mechanism. | The user identities provided to the TOE |
| FIA_UAU.2 | basic | All use of the user authentication mechanism. | The user identities provided to the TOE |
| FIA_AFL.1 | minimal | The reaching of the threshold for unsuccessful authentication attempts. | The user identities provided to the TOE |
| FDP_IFF.1 (1) | basic | All decisions on requests for information flow. | The presumed addresses of the source and destination subject. |
| FDP_IFF.1 (2) | basic | All decisions on requests for information flow. | The presumed addresses of the source and destination subject. |
| FDP_IFF.1 (3) | basic | All decisions on requests for information flow. | The presumed addresses of the source and destination subject. |
| FPT_STM.1 | Detailed | Providing a timestamp | Timestamp for use in audit log files |

Application Note:

The timestamp is provided by the underlying operating system and used for logging. FPT_STM.1 is part of the environment.

The auditable event FMT_SMR.1 “Minimal: modifications to the group of users that are part of a role” is not part of the TOE (the functional component FMT_SMR.1 is part of the environment). User accounts are managed by the underlying operating system.

The auditable event FCS_COP.1 “Minimal: Success and failure, and the type of cryptographic operation” is not part of the TOE (the functional component FCS_COP.1 is part of the environment). The underlying operating system logs cryptographic operation failures.

**FAU_SAR.1 Audit review**

| | |
|---|---|
| FAU_SAR.1.1 | The TSF shall provide [*assignment: an authorized administrator*] with the capability to read [*assignment: all audit trail data*] from the audit records. |
| FAU_SAR.1.2 | The TSF shall provide the audit records in a manner suitable for the user to interpret the information. |

**FAU_STG.3 Action in case of possible audit data loss**

| | |
|---|---|
| FAU_STG.3.1 | The TSF shall take [*assignment: alerting the administrator*] if the audit trail exceeds [*assignment: a defined capacity limit*]. |

## 5.2.2 Class FIA – Identification and authentication

**FIA_AFL.1 Authentication failure handling**

| | |
|---|---|
| FIA_AFL.1.1 | The TSF shall detect when [*assignment: one*] unsuccessful authentication attempts occur related to [*assignment: failed Basic authentication*]. |
| FIA_AFL.1.2 | When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall [*assignment: create a log file entry*]. |

**FIA_ATD.1 User attribute definition**

| | |
|---|---|
| FIA_ATD.1.1 | The TSF shall maintain the following list of security attributes belonging to individual users: [*assignment: identity*] |

**FIA_UID.2 User identification before any action**

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

**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.

## 5.2.3 Class FDP – User Data Protection

**FDP_IFC.1 Subset information flow control (1) – UNAUTHENTICATED SFP**

FDP_IFC.1.1 The TSF shall enforce the [*assignment: UNAUTHENTICATED SFP*] on

[*assignment:*

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

*b) information: packet traffic sent through the TOE from one subject to another;*

c) *operation: pass information*].

**FDP_IFC.1 Subset information flow control (2) – UNAUTHENTICATED_APPL SFP**

FDP_IFC.1.1 The TSF shall enforce the [*assignment: UNAUTHENTICATED_APPL SFP*] on

[*assignment:*

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

*b) information: RPC, HTTP, HTTPS traffic sent through the TOE from one subject to another;*

c) *operation: pass information*].

**FDP_IFC.1 Subset information flow control (3) – AUTHENTICATED SFP**

FDP_IFC.1.1 The TSF shall enforce the [*assignment: AUTHENTICATED SFP*] on

[*assignment:*

*a) subjects: an external IT entity that sends and receives application level traffic information through the TOE to one another, only after the user initiating the information flow has authenticated at the TOE per FIA_UAU.2,*

*b) information: HTTP, HTTPS traffic sent through the TOE from one subject to another;*

c) *operation: initiate service and pass information.*]

**FDP_IFF.1 Simple security attributes (1) – UNAUTHENTICATED SFP**

FDP_IFF.1.1 (1) The TSF shall enforce the [*assignment: UNAUTHENTICATED SFP*] based on the following types of subject and information security attributes:

[*assignment:*

a) *subject attributes:*

*presumed address;*

b) *information attributes:*

a. *presumed address of source subject;*

b. *presumed address of destination subject;*

c. *protocol type;*

d. *direction of connection establishment;*

e. *port numbers*].

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:

[*assignment:*

*a) Subjects on an internal network can cause information to flow through the TOE to another connected network if:*

*a. all the information security attribute values are unambiguously permitted by the information flow security policy rules, where such rules may be composed from*

*all possible combinations of the values of the information flow security attributes, created by the authorized administrator;*

*b. the presumed address of the source subject, in the information translates to an internal network address;*

*c. and the presumed address of the destination subject, in the information, translates to an address on the other connected network.*

*b) Subjects on the external network can cause information to flow through the TOE to another connected network if:*

*a. all the information security attribute values are unambiguously permitted by the information flow security policy rules, where such rules may be composed from all possible combinations of the values of the information flow security attributes, created by the authorized administrator;*

*b. the presumed address of the source subject, in the information translates to an external network address;*

c. *and the presumed address of the destination subject, in the information, translates to an address on the other connected network*.]

FDP_IFF.1.3 (1) The TSF shall enforce the [*assignment: none*].

FDP_IFF.1.4 (1) The TSF shall provide the following [*assignment: none*].

FDP_IFF.1.5 (1) The TSF shall explicitly authorize an information flow based on the following rules: [*assignment: none*].

FDP_IFF.1.6 (1) The TSF shall explicitly deny an information flow based on the following rules:

[*assignment:*

*a) The TOE shall reject requests for access or services where the information arrives on an external TOE interface, and the presumed address of the source subject is an external IT entity on an internal network;*

*b) The TOE shall reject requests for access or services where the information arrives on an internal TOE interface, and the presumed address of the source subject is an external it entity on the external network:*

*c) The TOE shall reject requests for access or services where the information arrives on either an internal or external TOE interface, and the presumed address of the source subject is an external IT entity on a broadcast network;*

d) *The TOE shall reject requests for access or services where the information arrives on either an internal or external TOE interface, and the presumed address of the source subject is an external IT entity on the loopback network*].

**FDP_IFF.1 Simple security attributes (2) – UNAUTHENTICATED_APPL SFP**

FDP_IFF.1.1 (2) The TSF shall enforce the [*assignment: UNAUTHENTICATED_APPL SFP*] based on the following types of subject and information security attributes:

[*assignment:*

a) *subject attributes:*

*presumed address;*

b) *information attributes:*

a. *presumed address of source subject;*

b. *presumed address of destination subject;*

c. *transport layer protocol;*

d. *direction of connection establishment;*

e. *services: RPC, HTTP, HTTPS*].

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

[*assignment:*

*a) Subjects on an internal network can cause information to flow through the TOE to another connected network if:*

*a. all the information security attribute values are unambiguously permitted by the information flow security policy rules, where such rules may be composed from all possible combinations of the values of the information flow security attributes, created by the authorized administrator;*

*b. the presumed address of the source subject, in the information translates to an internal network address;*

*c. and the presumed address of the destination subject, in the information, translates to an address on the other connected network.*

*b) Subjects on the external network can cause information to flow through the TOE to another connected network if:*

*a. all the information security attribute values are unambiguously permitted by the information flow security policy rules, where such rules may be composed from all possible combinations of the values of the information flow security attributes, created by the authorized administrator;*

*b. the presumed address of the source subject, in the information translates to an external network address;*

c. *and the presumed address of the destination subject, in the information, translates to an address on the other connected network.*]

FDP_IFF.1.3 (2) The TSF shall enforce the [*assignment: none*].

FDP_IFF.1.4 (2) The TSF shall provide the following [*assignment: none*].

FDP_IFF.1.5 (2) The TSF shall explicitly authorize an information flow based on the following rules: [*assignment: none*].

FDP_IFF.1.6 (2) The TSF shall explicitly deny an information flow based on the following rules:

[*assignment:*

*a) The TOE shall reject requests for access or services where the information arrives on an external TOE interface, and the presumed address of the source subject is an external IT entity on an internal network;*

*b) The TOE shall reject requests for access or services where the information arrives on an internal TOE interface, and the presumed address of the source subject is an external it entity on the external network:*

*c) c) The TOE shall reject requests for access or services where the information arrives on either an internal or external TOE interface, and the presumed address of the source subject is an external IT entity on a broadcast network;*

*d) The TOE shall reject requests for access or services where the information arrives on either an internal or external TOE interface, and the presumed address of the source subject is an external IT entity on the loopback network*]

**FDP_IFF.1 Simple security attributes (3) – AUTHENTICATED SFP**

FDP_IFF.1.1 (3) The TSF shall enforce the [*assignment: AUTHENTICATED SFP*] based on the following types of subject and information security attributes:

[*assignment:*

a) *subject attributes:*

   a. *presumed address;*

b) *information attributes:*

   a. *user identity*

   b. *presumed address of source subject;*

   c. *presumed address of destination subject;*

   d. *protocol type;*

   e. *direction of connection establishment;*

   f. *services: HTTP, HTTPS*].

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

[*assignment:*

*a) Subjects on an internal network can cause information to flow through the TOE to another connected network if:*

   *a. the human user initiating the information flow authenticates according to FIA_UAU.2;*

   *b. all the information security attribute values are unambiguously permitted by the information flow security policy rules, where such rules may be composed from all possible combinations of the values of the information flow security attributes, created by the authorized administrator;*

   *c. the presumed address of the source subject, in the information translates to an internal network address;*

   *d. and the presumed address of the destination subject, in the information, translates to an address on the other connected network.*

*b) Subjects on the external network can cause information to flow through the TOE to another connected network if:*

a. *the human user initiating the information flow authenticates according to FIA_UAU.2;*

b. *all the information security attribute values are unambiguously permitted by the information flow security policy rules, where such rules may be composed from all possible combinations of the values of the information flow security attributes, created by the authorized administrator;*

c. *the presumed address of the source subject, in the information translates to an external network address;*

d. *and the presumed address of the destination subject, in the information, translates to an address on the other connected network.*]

FDP_IFF.1.3 (3) The TSF shall enforce the [*assignment: none*].

FDP_IFF.1.4 (3) The TSF shall provide the following [*assignment: none*].

FDP_IFF.1.5 (3) The TSF shall explicitly authorize an information flow based on the following rules: [*assignment: none*].

FDP_IFF.1.6 (3) The TSF shall explicitly deny an information flow based on the following rules:

[*assignment:*

*a) The TOE shall reject requests for access or services where the information arrives on an external TOE interface, and the presumed address of the source subject is an external IT entity on an internal network;*

*b) The TOE shall reject requests for access or services where the information arrives on an internal TOE interface, and the presumed address of the source subject is an external it entity on the external network:*

*c) The TOE shall reject requests for access or services where the information arrives on either an internal or external TOE interface, and the presumed address of the source subject is an external IT entity on a broadcast network;*

d) *The TOE shall reject requests for access or services where the information arrives on either an internal or external TOE interface, and the presumed address of the source subject is an external IT entity on the loopback network*]

**FDP_RIP.1 Subset residual information protection**

| | |
|---|---|
| FDP_RIP.1.1 | The TSF shall ensure that any previous information content of a resource is made unavailable upon the [*selection: allocation of the resource to*] the following objects: [*assignment: resources that are used by the subjects of the TOE to communicate through the TOE to other subjects*]. |

## 5.2.4 Class FMT – Security Management

Application Note:

The TOE does not maintain the role “authorized administrator”. Access control to the TOE is granted by the underlying operating system which also maintains the role “authorized administrator”. So FMT_SMR.1 has been placed in the environment.

FMT_MSA.3 has been chosen because of dependencies of FMT_MSA.3.1 with FDP_IFF.1. FMT_MSA.3.2 is not applicable because the TOE has unchangeable default rules (deny all).

**FMT_MSA.3 Static attribute initialization**

| | |
|---|---|
| FMT_MSA.3.1 | The TSF shall enforce the [*assignment: information flow UNAUTHENTICATED SFP, UNAUTHENTICATED_APPL SFP, and AUTHENTICATED SFP,*] to provide [*selection: restrictive*] default values for information flow security attributes that are used to enforce the SFP. |
| FMT_MSA.3.2 | The TSF shall allow an [*assignment: authorized administrator*] to specify alternative initial values to override the default values when an object or information is created. |

## 5.2.5 Class FPT – Protection of the TSF

**FPT_RVM.1 Non-bypassability pf the TSP**

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

## 5.2.6 Minimum strength of function

The threat level for the TOE authentication function is assumed to be SOF-Basic. The minimum level is SOF-Basic. The strength of cryptographic algorithms is outside the scope of the CC. Strength of function only applies to non-cryptographic, probabilistic or

permutational mechanisms. The SOF requirement applies to the identification and authentication functionality within the TOE.

## 5.3 TOE Security Assurance Requirements

The Security Assurance Requirements for the TOE are the assurance components of Evaluation Assurance Level 2 (EAL2). They are all drawn from Part 3 of the Common Criteria. The assurance components are listed in Table 5.3.

**Table 5.3 – EAL2 Assurance Requirements**

| Assurance Component | Name |
|---|---|
| ACM_CAP.2 | Configuration items |
| ADO_DEL.1 | Delivery procedures |
| ADO_IGS.1 | Installation, generation, and start-up procedures |
| ADV_FSP.1 | Informal functional specification |
| ADV_HLD.1 | Descriptive high-level design |
| ADV_RCR.1 | Informal correspondence demonstration |
| AGD_ADM.1 | Administrator guidance |
| AGD_USR.1 | User guidance |
| ATE_COV.1 | Evidence of coverage |
| ATE_FUN.1 | Functional testing |
| ATE_IND.2 | Independent testing – sample |
| AVA_SOF.1 | Strength of TOE security function evaluation |
| AVA_VLA.1 | Developer vulnerability analysis |

## 5.4 Functional Security Requirements for the IT Environment

This chapter defines the TOE security functional requirements for the IT environment. A list of the requirements is provided in Table 5.4. The full text of the security functional requirements is contained below.

Note: In this chapter the wording “TSF” has been changed to “IT environment” according to the Final Interpretation 058.

**Table 5.4 – TOE Functional Security Requirements for the environment**

| # | Functional Requirement | Title | Dependencies |
|---|---|---|---|
| **Identification & Authentication** | | | |
| 1 | FCS_COP.1 | Cryptographic operation | FCS_CKM.1<br>FCS_CKM.4 |

| | | | |
|---|---|---|---|
| | | | FMT_MSA.2 |
| **Information Flow Control** | | | |
| 2 | FMT_MSA.1 (1) | Management of security attributes (1)– UNAUTHENTICATED SFP | FDP_IFC.1 FMT_SMR.1 |
| 3 | FMT_MSA.1 (2) | Management of security attributes (2) – UNAUTHENTICATED_APPL SFP | FDP_IFC.1 FMT_SMR.1 |
| 4 | FMT_MSA.1 (3) | Management of security attributes (3) – AUTHENTICATED SFP | FDP_IFC.1 FMT_SMR.1 |
| **Audit Generation** | | | |
| 5 | FPT_STM.1 | Reliable time stamps | |
| 6 | FAU_SAR.2 | Restricted audit review | FAU_SAR.1 |
| 7 | FAU_SAR.3 | Selectable audit review | FAU_SAR.1 |
| 8 | FAU_STG.1 | Protected audit trail storage | FAU_GEN.1 |
| **Security Management** | | | |
| 9 | FMT_SMR.1 | Security roles | |

Application note:

Dependencies for FCS_COP.1 are not further resolved because these components are part of the IT environment and handled by the underlying operating system. The IT environment has to ensure that the dependencies are fulfilled. These components are listed in Table 5.5 with a corresponding explanation.

**Table 5.5 – Dependencies of FCS_COP.1 fulfilled by the IT environment**

| | |
|---|---|
| FCS_CKM.1 Cryptographic key generation<br>FCS_CKM.4 Cryptographic key destruction<br>FMT_MSA.2 Secure security attributes | The TOE has an interface to the Security Support Provider Interface (SSPI), which enables to access dynamic-link libraries containing common authentication and cryptographic data schemes. The DLLs are called Security Support Providers (SSPs). SSPs make security packages available to applications. A security package maps various SSPI functions to the security protocols specified in the package. The SSPI libraries contain functions which are used to manage and establish secure connections, like cryptographic key generation and destruction. |

All other dependencies are fulfilled by the TOE or the IT environment.

## 5.4.1 Class FCS – Cryptographic support

**FCS_COP.1 Cryptographic operation**

FCS_COP.1.1 The IT environment shall perform [*assignment: encryption, decryption*] in accordance with a specific cryptographic algorithm [*assignment: see Table 5.6*] and cryptographic key sizes [*assignment: see Table 5.6*] that meet the following: [*assignment: SSL protocol*]

**Table 5.6 – Cipher types available in cryptographic API**

| **Cipher type**[10] | **minimum Key length used for symmetric encryption** |
|---|---|
| SSL_RSA_EXPORT_WITH_RC4_40_MD5 | 40 Bit RC4 |
| SSL_RSA_WITH_RC4_128_MD5 | 128 Bit RC4 |
| SSL_RSA_WITH_RC4_128_SHA | 128 Bit RC4 |
| SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5 | 40 Bit RC2 |
| SSL_RSA_WITH_DES_CBC_SHA | 56 Bit DES |
| SSL_RSA_WITH_3DES_EDE_CBC_SHA | 168 Bit 3DES |
| SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA | 168 Bit 3DES |
| SSL_RSA_EXPORT1024_WITH_DES_CBC_SHA | 56 Bit DES |
| SSL_RSA_EXPORT1024_WITH_RC4_56_SHA | 56 Bit RC4 |

Application Note:

RSA key length is set in the certificate used for the connection.

Since 1.1.2001 export regulations due to strong encryption do not longer exist, so higher encryption grades might be possible.

## 5.4.2 Class FPT – Protection of the TSF

**FPT_STM.1 Reliable time stamps**

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

[10] Reference (Knowledge Base Article): http://support.microsoft.com/default.aspx?scid=kb;en-us;245030

## 5.4.3 Class FAU – Security audit

**FAU_SAR.2 Restricted audit review**

FAU_SAR.2.1 The IT environment 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 IT environment shall provide the ability to perform [*selection: searches, sorting, ordering*] of audit data based on:

[*assignment:*

*a) user identity;*

*b) presumed subject address;*

*c) date;*

d) *time*].

**FAU_STG.1 Protected audit trail storage**

FAU_STG.1.1 The IT environment shall protect the storage audit records from unauthorized deletion.

FAU_STG.1.2 The IT environment shall be able to [*selection: prevent*] modifications to the audit records

## 5.4.4 Class FMT – Security Management

**FMT_MSA.1 Management of security attributes (1) – UNAUTHENTICATED SFP**

FMT_MSA.1.1 (1) The IT environment shall enforce the [*assignment: UNAUTHENTICATED SFP*] to restrict the ability to [*assignment: add a rule, delete a rule, modify attributes in a rule,*] the security attributes [*assignment: listed in section FDP_IFF1.1(1)*] to [*assignment: the authorized administrator*].

**FMT_MSA.1 Management of security attributes (2) – UNAUTHENTICATED_APPL SFP**

FMT_MSA.1.1 (2) The IT environment shall enforce the [*assignment: UNAUTHENTICATED_APPL SFP*] to restrict the ability to [*assignment: add a rule, delete a rule, modify attributes in a rule,*] the security

attributes [*assignment: listed in section FDP_IFF1.1(2)*] to [*assignment: the authorized administrator*].

**FMT_MSA.1 Management of security attributes (3) – AUTHENTICATED SFP**

| | |
|---|---|
| FMT_MSA.1.1 (3) | The IT environment shall enforce the [*assignment: AUTHENTICATED SFP*] to restrict the ability to [*assignment: add a rule, delete a rule, modify attributes in a rule,*] the security attributes [*assignment: listed in section FDP_IFF1.1 (3)*] to [*assignment: the authorized administrator*]. |

**FMT_SMR.1 Security roles**

| | |
|---|---|
| FMT_SMR.1.1 | The IT environment shall maintain the role [*assignment: authorized administrator*]. |
| FMT_SMR.1.2 | The IT environment shall be able to associate users with the role. |

## 5.5 Security Requirements for the Non-IT Environment

No security requirements for the Non-IT environment are defined.

# 6 TOE Summary Specification

The TOE summary specification in the following specifies the security functionality in form of security functions as well as the assurance measures of the TOE.

## 6.1 TOE Security Functions

The TOE consists of three security functions (SF) which will be described in more detail in the following chapters. These security functions are:

SF1: Identification and Authentication

SF2: Information Flow Control

SF3: Audit Generation

### 6.1.1 SF1 – Identification and Authentication

The TOE can be configured that only particular users are allowed to access the internet respectively the intranet through the TOE using Basic authentication.

Basic authentication is the standard method of authentication for Hypertext Transfer Protocol (HTTP) transmissions for incoming and outgoing requests. Basic authentication sends and receives user information as text characters. No encryption is used with Basic authentication. The following describes the authentication procedure: The TOE asks the client for user authentication. It gets the user name and password in clear text (base-64 encoded) and uses the data to get an impersonation token using the underlying operating system API “LogonUser”. This token is used to pass the rules.

Basic authentication for web requests can be secured using an SSL channel, so user identification and authentication credentials are encrypted during transmission.

User identification and authentication is configured in the TOE properties dialog separately from the information flow control rules. Because incoming and outgoing Web requests are independent of each other, it is also possible to configure the authentication methods differently for each. Both configurations – for incoming and outgoing web requests – allow selecting Basic authentication for a specific port. When a client requests HTTP content, the TOE checks the rules to determine if a specific rule allows anonymous users access (either because it applies to all users, or it applies to a client address set that includes the IP address of the client). If so, then the request will be allowed. Otherwise, if no rule has been configured to allow anonymous users access, the TOE will require that the client authenticate itself, to determine if a rule applies to the specific, authenticated user. In other words, when a client requests HTTP content, authentication information is not passed to the TOE, unless the TOE requires it.

The TOE can be forced to ask unauthenticated users for identification. A Web publishing rule or access policy rule can apply to client address sets or to users. If a rule applies to users then for outgoing requests, the TOE will check outgoing Web request properties to

determine how the user should be authenticated. For incoming requests, the TOE will check incoming Web request properties. When the user is forced to identify himself and he does not provide the information asked for the request will be denied.

Because no encryption is used for username and password transmission the security function claims for SOF-basic.

### 6.1.2 SF2 – Information Flow Control

The TOE controls the flow of incoming and outgoing IP packets and controls information flow on protocol level. This control has to be active before any information can be transmitted through the TOE. So information flow control is subdivided into IP packet filters, Protocol rules, Site and content rules, Server- and Web publishing, and Application filters.

**Packet filters**

Packet filters control the flow of IP packets to and from the TOE. All packets on the external interface are dropped unless they are explicitly allowed.

For information flow control the TOE offers IP packet filters, protocol rules and site and content rules.

**IP packet filters** allow or block packets destined for specific computers on the internal network. For IP packet filters the TOE allows

- the selection a protocol filter from a predefined list of protocols:

  DNS lookup,

  ICMP (all outbound/ping response/ping query/source quench/timeout/unreachable),

  PPTP (call/receive),

  SMTP,

  POP3,

  Identd,

  HTTP (port 80/port 443),

  NetBIOS (WINS client only/all),

  and the IP address to which the IP packet filter is applied (default IP addresses for each external interface on the TOE, a specific external IP address or a specific computer)

- defining a custom filter which contains

  the protocol type (TCP, UDP and ICMP),

  the direction of connection establishment (inbound/outbound/both),

  local port IP and port number (can be “all ports”, “fixed port” or “dynamic”, port between 1025 and 5000),

  remote port IP and port number (can be “all ports” or “fixed ports”),

  and the IP address to which the IP packet filter is applied (default IP addresses for each external interface on the TOE, a specific external IP address or a specific computer)

**Protocol rules** define which protocols can be used for communication between the local network and the Internet. For Protocol rules the TOE allows

- the action taken with requests (allow or deny),
- the protocols the rule applies to (“all IP traffic”, “selected protocols” (from a predefined list) or “all IP traffic except selected”),
- a schedule when the protocol rule is active or inactive,
- and the requests the rule applies to (any request, a specific client set or a user or groups from the domain namespace).

The predefined list maps a protocol name to a specific port number, port type and direction.

**Site and content rules** define what content on which Internet sites can be accessed by clients behind the TOE. For Site and content rules the TOE allows

- the type of action the rule performs (allow or deny),
- deny rules can optionally redirect a HTTP request to another site,
- the destination the rule applies to (all destinations, all internal destinations, all external destinations, specific destination sets, all destinations except those from a selected list),
- a schedule when the site and content rule is active or inactive,
- the possibility to specify content type (MIME),
- and the requests the rule applies to (any request, a specific client set or a user or groups from the domain namespace).

**Server publishing**

Server publishing rules map incoming requests to the appropriate servers behind the TOE. These rules will grant access dynamically, as specified, from Internet users to the specific publishing server.

Server publishing define

- the IP address of internal server,
- the external IP address on the TOE,

- mapped server protocol,
- used port,
- restrictions (all or defined client sets).

**Web publishing**

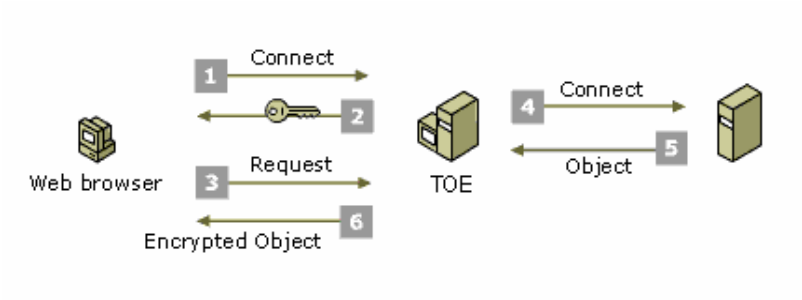
The TOE uses Web publishing rules to relieve the concerns associated with publishing Web content to the Internet without compromising internal network security. Web publishing rules determine how the TOE should intercept incoming requests for Hypertext Transfer Protocol (HTTP) objects on an internal Web server and how the TOE should respond on behalf of the Web server. Requests are forwarded downstream to an internal Web server, located behind the TOE.

It allows to define:

- IP address of internal server,
- external IP address on ISA Server,
- whether the request should be discarded or redirected,
- destination of URL redirection,
- destination ports when bridging requests as HTTP, HTTPS (SSL) or FTP,
- type of bridging for HTTP requests (HTTP, SSL, FTP),
- type of bridging for SSL requests (HTTP, SSL, FTP),
- SSL requirements (secured SSL channel required for published site and (optional) with 128-bit encryption),
- optional a required certificate to authenticate the Web server,
- restrictions (rule applies to any requests, specified client addresses, selected users and groups from the domain namespace).

Figure 6.1 illustrates an SSL bridging scenario. In this scenario, the client requests objects from the TOE, which forwards the request to the published Web server. In the figure, the Web browser connects to the TOE. The TOE returns a server-side certificate, authenticating itself to the client. When the client and the TOE complete the SSL negotiation, the client sends an encrypted HTTP request to the TOE. The TOE decrypts the request and checks if the requested object is in its cache. If the object is in the cache, the TOE returns the object to the client. If the object is not in the cache, then the TOE encrypts the request and sends the request to the Web server. The Web server returns a server-side certificate to the TOE. When the TOE and the Web server complete the SSL negotiation, the TOE sends the encrypted HTTP request to the Web server. The Web server decrypts the request and returns it to the TOE.

**Figure 6.1 – SSL bridging scenario**




**Application filters**

The Exchange RPC filter provided with the TOE enables publishing of Exchange RPC servers, making them accessible to external clients.

The Exchange RPC filter:

- filters UUIDs (Universal Unique Identifier[11]),
- enforces encryption (this functionality is not part of the evaluation),
- validates the RPC packet syntax (max size and validation of the RPC bind request).

The RPC filter provided with ISA Server enables publishing of RPC servers, making them accessible to external clients.

The RPC filter adds the “Exchange RPC (Server)” protocol definition which can be used in protocol rules. The RPC filter can be configured to filter specific UUIDs using the RPC Wizard within the TOE. It permits the administrator to select the services from a list of interfaces available on the server that the wizard presents, or define them manually. These service definitions can be used in server publishing rules so that external clients can access them.

In an Exchange Server/Outlook client scenario for example, the RPC application filter works as follows:

1. The Outlook client issues request over Port 135 (TCP) through the TOE to the Exchange Server, to find the service port number associated with the Exchange RPC UUID.
2. The Exchange Server sends a response back, through the TOE, to the Outlook client, with a port number on which the client can communicate. The connection to Port 135/tcp is then closed.
3. The TOE uses the RPC application filter to capture this information, and maintains it in a table.

[11] A UUID is an identifier that is unique across both space and time, with respect to the space of all UUIDs. A UUID can be used for multiple purposes, from tagging objects with an extremely short lifetime, to reliably identifying very persistent objects across a network.

4. The TOE allocates a new port on the TOE itself, and changes the response that it sends to the Outlook client, to reflect this change. This information is also maintained in the table.
5. The Outlook client issues a request–seemingly to the Exchange Server, but actually to the new port on the TOE. The TOE then sends the packet to the Exchange Server. Only communication over this port is allowed.

**Some additional notes about controlling incoming and outgoing web requests**

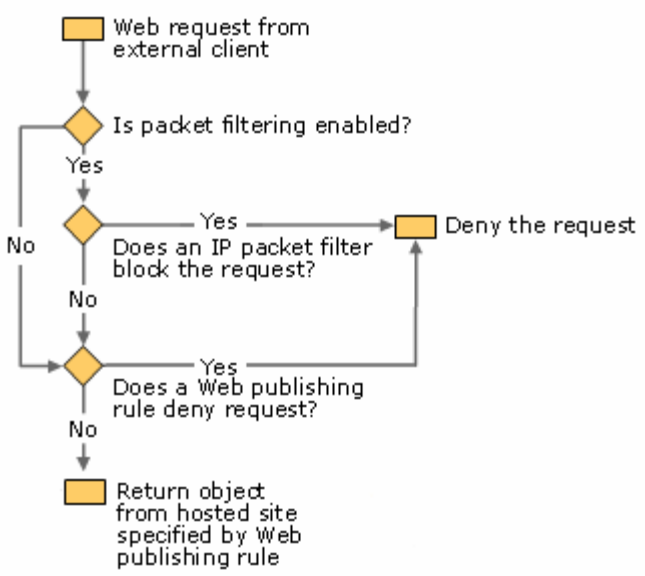
The following figures demonstrate how ISA Server respectively the TOE processes a request from an external or internal client.

**Incoming Web Request:** For an incoming Web Request, ISA Server checks IP packet filters and publishing rules to determine if the request is allowed and which internal server should service the request. The rules are processed in the following order:

1. IP packet filters
2. Web publishing rules

Figure 6.2 illustrates the processing flow for an incoming Web request.

**Figure 6.2 – Incoming Web Request**

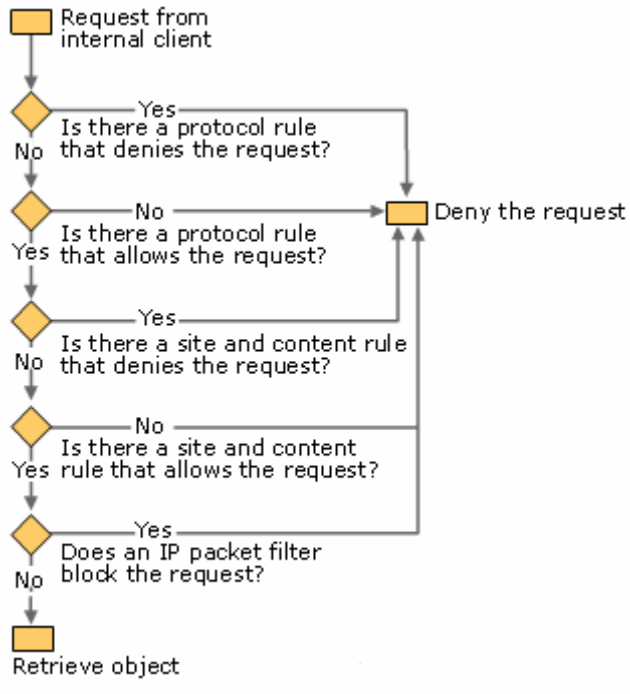



**Outgoing Web Request:** For an outgoing Web Request, ISA Server checks site and content rules and protocol rules to determine if access is allowed. A request is allowed only if both a protocol rule and a site and content rule each allow the request and if there is no rule that explicitly denies the request. The rules are processed in the following order:

1. Protocol rules
2. Site and content rules
3. IP packet filters

Figure 6.3 illustrates the processing flow for an outgoing Web request.

**Figure 6.3 – Outgoing Web Request**




The TOE ensures that information contained in packets from previous sessions is no longer accessible once the session has been completed. The storage and processing of data packets through the TOE ensures that no residual information is transferred to future sessions through the firewall.

This security function has no probabilistic or permutational mechanism and therefore no SoF claim is necessary.

### 6.1.3 SF3 – Audit Generation

The TOE stores logging information in different log files:

- Packet filter log file

  By default the Packet Filter log file contains records of packets that were dropped in the packet filter level. It is possible to turn on logging for packets that were permitted to traverse the firewall. Blocking IP packet filters can be configured selectively to create or not to create a log file entry when a packet has been blocked.

- Firewall service log file

  The Firewall log file contains 2 rows per connection: the first is upon connection establishment. The seconds is when the connection terminates. In case there is no permission to establish the connection only the first row is used (with status “access denied”).

- Web proxy service log file

  The Web Proxy log file stores a line per HTTP request that it gets. Each request is always logged.

- Windows application event log file

  The Windows application event log file stores important system events and failures.

and detects the occurrence of the following selected events:

- filter rules permitted (packet filter log file),
- filter rules denied (packet filter log file),
- failed authentication of users (firewall service log file),
- passed requests though the TOE (firewall service log file),
- passed requests of users that have been previously authenticated through the TOE (firewall service log file),
- received HTTP requests (web proxy log file),
- log failure (windows event log file),
- service started, stopped or not responding (windows event log file).

Options for log file generation can be:

- daily,
- weekly,
- monthly or
- yearly.

Table 6.1 lists fields that are stored in the packet filter log file. Table 6.2 contains fields that are either stored in firewall service log file or in web proxy service log file. The Windows event log file contains information about: event type, date and time, source of the event, event category and user information (if applicable).

**Table 6.1 – TOE audit information (packet filters)**

| Descriptive name | Description |
|---|---|
| Date | Date the packet was received. |
| Time | The time the packet was received (service info fields) |
| Source IP | The Internet Protocol (IP) address of the source (remote) computer. The source computer is the computer from which the data packets originated. |
| Destination IP | The IP address of the destination (local) computer. The destination computer is usually the ISA Server computer. |
| Protocol | The particular transport level protocol that is used during the connection, such as Transmission Control Protocol (TCP), User Datagram Protocol (UDP), or Internet Control Message Protocol (ICMP). |
| Source port | For TCP and UDP protocols, the remote port used to create a connection. For |

| | |
|---|---|
| (or protocol type, if ICMP) | ICMP protocol, the type used when creating the connection. |
| Destination port<br>(or protocol code, if ICMP) | For TCP and UDP protocols, the local port used to create a connection. For ICMP protocol, the code used when creating the connection. |
| TCP flags (tcp-flags) | For a TCP data packet, represents the TCP flag value in the IP header. The possible values are FIN, SYN, RST, PSH, ACK, and URG |
| Interface | Indicates whether the packet was accepted (1) or dropped (0). By default, only dropped packets are logged. |
| Interface IP address | Interface on which the packet was received; usually only one interface. |
| Header | The entire IP header of the data packet that generated the alert event. The IP header is logged in hexadecimal format. |
| Payload | A listing of a portion of the data packet (after the IP header). The IP packet is logged in hexadecimal format. |

**Table 6.2 – TOE audit information (firewall and proxy service)**

| Descriptive name | Description |
|---|---|
| Client IP | The Internet Protocol (IP) address of the requesting client. |
| Client user name | Account of the user making the request. If ISA Server Access Control is not being used, ISA Server uses anonymous. |
| Client agent | The client application type sent by the client in the Hypertext Transfer Protocol (HTTP) header. When ISA Server is actively caching, the client agent is ISA Server.<br>For Firewall service, this field includes information about the client's operating system. |
| Authentication status | Indicates whether or not client has been authenticated with ISA Server. Possible values are Y and N. |
| Date | The date that the logged event occurred. |
| Time | The time that the logged event occurred. In W3C format, this is in Greenwich mean time. |
| Service name | The name of the service that is logged.<br>w3proxy indicates outgoing Web requests to the Web Proxy service.<br>fwsrv indicates Firewall service.<br>w3reverseproxy indicates incoming Web requests to the Web Proxy service. |
| Proxy name | The name of the computer running ISA Server. This is the computer name that is assigned in Windows 2000. |
| Referring server name | If ISA Server is used upstream in a chained configuration, this indicates the server name of the downstream server that sent the request. |
| Destination name | The domain name for the remote computer that provides service to the current connection. For the Web Proxy service, a hyphen (-) in this field may indicate that an object was retrieved from the Web Proxy server cache and not from the destination. |
| Destination IP | The network IP address for the remote computer that provides service to the current connection. For the Web Proxy service, a hyphen (-) in this field may indicate that an object was sourced from the Web Proxy server cache and not from the destination. One exception is negative caching. In that case, this field indicates a destination IP address for which a negative-cached object was returned. |
| Destination port | The reserved port number on the remote computer that provides service to the |

| | |
|---|---|
| | current connection. This is used by the client application initiating the request. |
| Processing time | This indicates the total time, in milliseconds, that is needed by ISA Server to process the current connection. It measures elapsed server time from the time that the server first received the request to the time when final processing occurred on the server — when results were returned to the client and the connection was closed.<br>For cache requests that were processed through the Web Proxy service, processing time measures the elapsed server time needed to fully process a client request and return an object from the server cache to the client. |
| Bytes sent | The number of bytes sent from the internal client to the external server during the current connection. A hyphen (-), a zero (0), or a negative number in this field indicates that this information was not provided by the remote computer or that no bytes were sent to the remote computer. |
| Bytes received | The number of bytes sent from the external computer and received by the client during the current connection. A hyphen (-), a zero (0), or a negative number in this field indicates that this information was not provided by the remote computer or that no bytes were received from the external computer. |
| Protocol name | Specifies the application protocol used for the connection. Common values are HTTP, File Transfer Protocol (FTP), Gopher, and Secure Hypertext Transfer Protocol (HTTPS).<br>For Firewall service, the port number is also logged. |
| Transport | Specifies the transport protocol used for the connection. Common values are Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). |
| Operation | Specifies the application method used. For Web Proxy, common values are GET, PUT, POST, and HEAD.<br>For Firewall service, common values are CONNECT, BIND, SEND, RECEIVE, GHBN (GetHostByName), and GHBA (GetHostByAddress). |
| Object name | For the Web Proxy service, this field shows the contents of the URL request. This field applies only to the Web Proxy service log. |
| Object MIME | The Multipurpose Internet Mail Extensions (MIME) type for the current object. This field may also contain a hyphen (-) to indicate that this field is not used or that a valid MIME type was not defined or supported by the remote computer. This field applies only to the Web Proxy service log. |
| Object source | Indicates the source that was used to retrieve the current object. This field applies only to the Web Proxy service log. Click here to see a table of some possible values. |
| Result code | This field can be used to indicate:<br>For values less than 100, a Windows (Win32) error code<br>For values between 100 and 1,000, an HTTP status code<br>For values between 10,000 and 11,004, a Winsock error code<br>Click here to see a table of some possible values. |
| Cache info | This number reflects the cache status of the object, which indicates why the object was or was not cached. This field applies only to the Web Proxy service log. Click here to see a table of some possible values. |
| Rule #1 | This reflects the rule that either allowed or denied access to the request, as follows:<br>o If an outgoing request is allowed, this field reflects the protocol rule that allowed the request.<br>o If an outgoing request is denied by a protocol rule, this field reflects the protocol rule.<br>o If an outgoing request is denied by a site and content rule, this field |

| | |
|---|---|
| | reflects the protocol rule that would have allowed the request.<br>o If an incoming request was denied, this field reflects the Web publishing or server publishing rule that denied the request.<br>o If no rule specifically allowed the outgoing or incoming request, the request is denied. In this case, the field is empty. |
| Rule #2 | This reflects the second rule that either allowed or denied access to the request.<br>o If an outgoing request is allowed, this field reflects the site and content rule that allowed the request.<br>o If an outgoing request is denied by a site and content rule, this field reflects the site and content rule that denied the request.<br>o If no rule specifically allowed the outgoing or incoming request, the request is denied. In this case, the field is empty. |
| Session ID | This identifies a session's connections. For Firewall clients, each process that connects through the Firewall service initiates a session. For secure network address translation (SecureNAT) clients, a single session is opened for all the connections that originate from the same IP address. This field is not included in the Web Proxy service log. This field applies only to the Firewall service log. |
| Connection ID | This identifies entries that belong to the same socket. Outbound TCP usually has two entries for each connection: when the connection is established and when the connection is terminated. UDP usually has two entries for each remote address. This field is not included in the Web Proxy service log. This field applies only to the Firewall service log. |

This security function has no probabilistic or permutational mechanism and therefore no SoF claim is necessary.

### 6.1.4 Assignment of SFs to security functional requirements

The justification of the mapping between security functional requirements and security functions is given in this chapter 6.1.4. The results are summarized in Table 6.3.

**Table 6.3 – Assignment of security functional requirements to security functions**

| # | SFR | SF1 | SF2 | SF3 |
|---|---|---|---|---|
| 1 | FAU_GEN.1 | | | X |
| 2 | FAU_SAR.1 | | | X |
| 3 | FAU_STG.3 | | | X |
| 4 | FIA_AFL.1 | X | | |
| 5 | FIA_ATD.1 | X | | |
| 6 | FIA_UID.2 | X | | |

| | | | | |
|---|---|---|---|---|
| 7 | FIA_UAU.2 | X | | |
| 8 | FDP_IFC.1 (1) – UNAUTHENTICATED SFP | | X | |
| 9 | FDP_IFC.1 (2) – UNAUTHENTICATED_APPL SFP | | X | |
| 10 | FDP_IFC.1 (3) – AUTHENTICATED SFP | | X | |
| 11 | FDP_IFF.1 (1) – UNAUTHENTICATED SFP | | X | |
| 12 | FDP_IFF.1 (2) – UNAUTHENTICATED_APPL SFP | | X | |
| 13 | FDP_IFF.1 (3) – AUTHENTICATED SFP | | X | |
| 14 | FDP_RIP.1 | | X | |
| 15 | FMT_MSA.3 | | X | |
| 16 | FPT_RVM.1 | | X | |

**FAU_GEN.1** (Audit data generation) is mapped to SF3 and outlines what data must be included in audit records. Audit data generated by the TOE is stored in different log files as stated in SF3. When applicable, information about the identified user is stored in the log files.

This component traces back to and aids in meeting the following objectives: O.AUDREC and O.ACCOUN because the TOE generates a readable audit trail of security-related events which contains user accountability for information flows.

**FAU_SAR.1** (Audit review) is mapped to SF3 and ensures that the user can interpret the recorded information. The log files are stored in human readable form (clear text) by the TOE and can be reviewed by any suitable tool running on the underlying operating system.

This component traces back to and aids in meeting the following objective: O.AUDREC because the TOE generates a human readable (clear text) audit trail of security-related events.

**FAU_STG.3** (Action in case of possible audit data loss) is mapped to SF3 and ensures that the user is alerted in case of possible audit data loss.

This component traces back to and aids in meeting the following objective: O.AUDREC because the TOE makes sure that no records are lost (for example of not enough storage capacity).

**FIA_AFL.1** (Authentication failure handling) is mapped to SF1. This component exists to specify action after some number of unsuccessful authentication attempts. It ensures that users cannot endlessly attempt to authenticate without leaving no trace in the log files.

This component traces back to and aids in meeting the following objectives: O.IDAUTH because the TOE uniquely identifies the user and authenticates the claimed identify for all users.

**FIA_ATD.1** (User attribute definition) is mapped to SF1. This component exists to provide users with attributes to distinguish one user from another, for accountability purposes and to associate the role chosen in FMT_SMR.1 with a user.

This component traces back to and aids in meeting the following objectives: O.IDAUTH because the TOE identifies the user with his username.

**FIA_UID.2** (User identification before any action) is mapped to SF1. This component ensures that the user identify himself (when required) before any information is passed though the TOE. The Basic authentication method provides this functionality for the users.

This component traces back to and aids in meeting the following objectives: O.IDAUTH and O.ACCOUN because the user is identified with his username witch has to exist in the local user database to be authenticated successfully.

**FIA_UAU.2** (User authentication before any action) is mapped to SF1 and ensures that users are identified when necessary. When authentication is required it must occur before any data is passed though the TOE. Basic authentication method provides this functionality for the users. Note, that firewall administrators are not authenticated by the TOE itself. This is done by the environment (underlying operating system).

This component traces back to and aids in meeting the following objectives: O.IDAUTH and O.ACCOUN because the user is identified with his username witch has to exist in the local user database to be authenticated successfully.

Application Note:

This Security Target consists of multiple information flow control Security Function Policies (SFPs). The CC allows multiple policies to exist, each having a unique name. This is accomplished by iterating FDP_IFC.1 for each of the three named information flow control policies. Following SFPs exist:

- UNAUTHENTICATED SFP and UNAUTHENTICATED_APPL SFP

  The subjects under control of this policy are external IT entities on an internal or external network sending information through the TOE to other external IT entities.

- AUTHENTICATED SFP

  The subjects under control of this policy are human users on an internal or external network who must be authenticated at the TOE before using the services in FIA_UAU.2. The information flowing between subjects in both policies is traffic with attributes, defined in FDP_IFF.1.1, including source and destination addresses. The rules that define each information flow control SFP are found in FDP_IFF.1.2. Component FDP_IFF.1 is iterated third times to correspond to each of the three iterations of FDP_IFC.1.

**FDP_IFC.1 (1)** (Subset information flow control (1)) is mapped to SF2 and identifies the entities involved in the UNAUTHENTICATED information flow control SFP (i.e., users sending information to other users and vice versa). It refers to the IP packet filters and Server publishing mentioned in SF2.

This component traces back to and aids in meeting the following objective: O.MEDIAT because the TOE mediates the flow of all information from users on a connected network to users on another connected network.

**FDP_IFC.1 (2)** (Subset information flow control (2)) is mapped to SF2 and identifies the entities involved in the UNAUTHENTICATED_APPL information flow control SFP (i.e., users sending information on application level to other users and vice versa). It refers to the Protocol rules, Site and content rules, Web publishing and Application filters that are used unauthenticated mentioned in SF2.

This component traces back to and aids in meeting the following objective: O.MEDIAT because the TOE mediates the flow of all information from users on a connected network to users on another connected network.

**FDP_IFC.1 (3)** (Subset information flow control (3)) is mapped to SF2 and identifies the entities involved in the AUTHENTICATED information flow control SFP. Users who want to use one of these services must be authenticated at the TOE. It refers to the HTTP and HTTPS protocols used in Protocol rules, Site and content rules and Web publishing that are used authenticated as mentioned in SF2.

This component traces back to and aids in meeting the following objective: O.MEDIAT because the TOE mediates the flow of all information from users on a connected network to users on another connected network.

**FDP_IFF.1 (1)** (Simple security attributes (1)) is mapped to SF2 and identifies the attributes of the users sending and receiving the information in the UNAUTHENTICATED SFP, as well as the attributes for the information itself. Then the policy is defined by saying under what conditions information is permitted to flow.

This component traces back to and aids in meeting the following objective: O.MEDIAT because the TOE mediates the flow of all information from users on a connected network to users on another connected network.

**FDP_IFF.1 (2)** (Simple security attributes (2)) is mapped to SF2 and identifies the attributes of the users sending and receiving the information in the UNAUTHENTICATED_APPL SFP for data transferred on application level, as well as the attributes for the information itself. Then the policy is defined by saying under what conditions information is permitted to flow.

This component traces back to and aids in meeting the following objective: O.MEDIAT because the TOE mediates the flow of all information from users on a connected network to users on another connected network.

**FDP_IFF.1 (3)** (Simple security attributes (3)) is mapped to SF2 and identifies the attributes of the users sending and receiving the information in the AUTHENTICATED SFP, as well as the attributes for the information itself. Then the policy is defined by saying under what conditions information (data sent on application level) is permitted to flow.

This component traces back to and aids in meeting the following objective: O.MEDIAT because the TOE mediates the flow of all information from users on a connected network to users on another connected network.

**FDP_RIP.1** (Subset residual information protection) is mapped to SF2 and ensures that neither information that had flown through the TOE nor any TOE internal data are used when padding is used by the TOE for information flows. Future sessions will not contain residual information of previous sessions in padding data.

This component traces back to and aids in meeting the following objective: O.MEDIAT because the TOE mediates the flow of all information from users on a connected network to users on another connected network.

**FMT_MSA.3** (Static attribute initialization) is mapped to SF2. This component ensures that there is a default deny policy for the information flow control security rules. The TOE ensures that by default all traffic through the TOE is denied.

This component traces back to and aids in meeting the following objectives: O.MEDIAT, O.SECSTA because the TOE mediates the flow of all information from users on a connected network to users on another connected network and ensures that the TOE must not compromises its resources or those of any connected network.

**FPT_RVM.1** (Non-bypassability of the TSP) is mapped to SF2 and ensures that on initial start-up of the TOE or recovery from an interruption the security function is invoked before any information is transmitted via the TOE.

This component traces back to and aids in meeting the following objective: O.SECSTA because it ensures that the TOE must not compromises its resources or those of any connected network on initial start-up or recovery from an interruption.

## 6.2 Assurance Measures

In Table 6.4 the TOE specific assurance measures are listed. These measures, mainly consisting of providing appropriate documentation, are fulfilling the requirements from table 5.2:

**Table 6.4 – Assurance requirements and assurance measures**

| Assurance requirements according to EAL2 | Assurance measures of the developer |
|---|---|
| **Configuration management**<br>ACM_CAP.2 (Configuration Items) | Application of a QM System including configuration control |
| **Delivery and operation**<br>ADO_DEL.1 (Delivery procedures)<br>ADO_IGS.1 (Installation, generation and start-up procedures) | Documentation of the TOE's protection mechanisms with regard to delivery, installation and start-up |
| **Development**<br>ADV_FSP.1 (Informal functional specification)<br>ADV_HLD.1 (Descriptive high-level design)<br>ADV_RCR.1 (Informal corresponding demonstration) | Definition of CC requirements with regard to development procedures and documentation, high-level design, functional specification and corresponding demonstration. |
| **Guidance documents**<br>AGD_ADM.1 (Administrator guidance)<br>AGD_USR.1 (User guidance) | Creating and delivery of administrator and user guidance |
| **Tests**<br>ATE_COV.1 (Evidence of Coverage)<br>ATE_FUN.1 (Functional Testing)<br>ATE_IND.2 (Independent Testing – sample) | Independent testing of a subset of the TSF, whether the TOE behaves as specific in the design documentation and in accordance with the TOE security functional environment. |
| **Vulnerability assessment**<br>AVA_SOF.1 (Strength of TOE security function evaluation)<br>AVA_VLA.1 (Developer vulnerability analysis) | Analyzing the security-relevant mechanisms with regard to SOF Basic (SOF document) and vulnerability analysis of obvious TOE vulnerabilities (VLA document). |

# 7 PP Claims

This security target does not claim for compliance with any existing protection profile.

Some aspects are leant on the “Application-level Firewall Protection Profile for Low-Risk Environments, Version 1.d, U.S. Government, July 20, 1999” [PP].

# 8 Rationale

This chapter provides the evidence used in the ST evaluation. This evidence supports the claims that the ST is a complete and cohesive set of requirements, that a conformant TOE would provide an effective set of IT security countermeasures within the security environment, and that the TOE summary specification addresses the requirements.

## 8.1 Security Objectives Rationale

Table 8.1 maps assumptions and threats to objectives, demonstrating that all assumptions and threats are mapped to at least one objective. Table 8.2 maps objectives to threats and assumptions, demonstrating that all objectives are mapped to at least one threat or assumption. A discussion of the rationale for threat mappings is provided below.

**Table 8.1 – Mapping the TOE Security Environment to Objectives**

| # | **Assumption / Threat / Policy** | **Security Objective** |
|---|---|---|
| 1 | A.PHYSEC | OE.PHYSEC |
| 2 | A.GENPUR | OE.GENPUR |
| 3 | A.NOEVIL | OE.NOEVIL |
| 4 | A.SINGEN | OE.SINGEN |
| 5 | A.DIRECT | OE.DIRECT |
| 6 | A.SECINST | OE.SECINST |
| 7 | A.OS | OE.OS |
| 8 | T.NOAUTH | O.IDAUTH, O.SECSTA |
| 9 | T.ASPOOF | O.MEDIAT |
| 10 | T.MEDIAT | O.MEDIAT |
| 11 | T.OLDINF | O.MEDIAT |
| 12 | T.AUDFUL | O.AUDREC |
| 13 | P.AUDACC | O.AUDREC, O.ACCOUN |

**Table 8.2 – Tracing of Security Objectives to Threats, Policies and Assumptions**

| # | Security Objective | Threat / Assumption / Policy |
|---|---|---|
| 1 | OE.PHYSEC | A.PHYSEC |
| 2 | OE.GENPUR | A.GENPUR |
| 3 | OE.NOEVIL | A.NOEVIL |
| 4 | OE.SINGEN | A.SINGEN |
| 5 | OE.DIRECT | A.DIRECT |
| 6 | OE.SECINST | A.SECINST |
| 7 | OE.OS | A.OS |
| 8 | O.IDAUTH | T.NOAUTH |
| 9 | O.MEDIAT | T.ASPOOF, T.MEDIAT, T.OLDINF |
| 10 | O.SECSTA | T.NOAUTH |
| 11 | O.AUDREC | P.AUDACC, T.AUDFUL |
| 12 | O.ACCOUN | P.AUDACC |

Note:

The security objectives for the environment are a restatement of the assumptions for the environment.

**T.NOAUTH:** An unauthorized user may attempt to bypass the security of the TOE so as to access and use security functions and/or non-security functions provided by the TOE. T.NOAUTH is countered by O.IDAUTH, O.SECSTA because the security objective ensures that the user has to authenticate before access is granted to TOE functions and the TOE ensures that its resources or those of the connected network are not compromised.

**T.ASPOOF:** An attacker may carry out spoofing in information flows mediated by the TOE between clients and servers located on internal and external networks governed by the TOE, by using a spoofed source address.

T.ASPOOF is countered by O.MEDIAT because the security objective ensures that the TOE mediates the flow of all information from users on the connected network to users on another connected network.

**T.MEDIAT:** An attacker may send impermissible information through the TOE, which results in the exploitation of resources on the internal network.

T.MEDIAT is countered by O.MEDIAT because the security objective ensures that the TOE mediates the flow of all information from users on the connected network to users on another connected network.

**T.OLDINF:** Because of a flaw in the TOE functioning, an attacker may gather residual information from a previous information flow or internal TOE data by monitoring the padding data of the information flows from the TOE. Padding data ensures that data packets contain the required number of bits and bytes and could contain residual information from previous connections.

T.OLDINF is countered by O.MEDIAT because the security objective ensures that the TOE mediates the flow of all information from users on the connected network to users on another connected network and ensures that information from a previous information flow is not available.

**T.AUDFUL**: An attacker may cause audit records to be lost or prevent future records from being recorded by taking actions to exhaust audit storage capacity, thus masking an attackers actions.

T.AUDFUL is countered by O.AUDREC because the security objective ensures that the TOE records a reliable readable audit trail and that no records are left because of less storage capacity.

**P.AUDACC:** Persons must be accountable for the actions that they conduct. Therefore audit records must contain sufficient information to prevent an attacker to escape detection.

P.AUDACC is countered by O.AUDREC, O.ACCOUN because the security objective ensures that a person is identified to make the person accountable for the action and that this action is logged in the audit trail.

**O.IDAUTH**: This security objective is necessary to counter the threat T.NOAUTH. It requires that users be uniquely identified before accessing the TOE and sending information through the TOE.

**O.MEDIAT**: This security objective is necessary to counter the threats: T.ASPOOF, T.MEDIAT and T.OLDINF which have to do with getting impermissible information to flow through the TOE. This security objective requires that all information that passes through the networks is mediated by the TOE and that no residual information is transmitted.

**O.SECSTA**: This security objective ensures that no information is comprised by the TOE upon start-up or recovery and thus counters the threats: T.NOAUTH.

**O.AUDREC**: This security objective is necessary to counter the policy: P.AUDACC by requiring a readable audit trail and a means to search and sort the information contained in the audit trail and T.AUDFUL by requiring that no records are left because of not enough storage capacity.

**O.ACCOUN**: This security objective is necessary to counter the policy: P.AUDACC because it requires that users are accountable for information flows through the TOE and that authorized administrators are accountable for the use of security functions related to audit.

## 8.2 Security Requirements Rationale

In this chapter, the security objectives are mapped to the functional requirements and the rationale is provided for the selected EAL and its components and augmentation.

### 8.2.1 Security Functional Requirements Rationale

The mapping of security objectives to functional requirements (components) is provided in Table 8.3. The mapping of security objectives of the environment to functional requirements (components) is provided in Table 8.5.

**Table 8.3 – Security Objective to Functional Component Mapping**

| # | Security Objectives | Functional Component (SFR TOE) |
|---|---|---|
| 1 | O.IDAUTH | FIA_AFL.1, FIA_ATD.1, FIA_UID.2, FIA_UAU.2 |
| 2 | O.MEDIAT | FDP_IFC.1 (1), FDP_IFC.1 (2), FDP_IFC.1 (3), FDP_IFF.1 (1), FDP_IFF.1 (2), FDP_IFF.1 (3), FMT_MSA.3, FDP_RIP.1 |
| 3 | O.SECSTA | FMT_MSA.3, FPT_RVM.1 |
| 4 | O.AUDREC | FAU_GEN.1, FAU_SAR.1, FAU_STG.3 |
| 5 | O.ACCOUN | FAU_GEN.1, FIA_UID.2, FIA_UAU.2 |

**Table 8.4 – Functional Requirements to Objectives Mapping**

| # | Functional Requirements (SFR TOE) | Security Objectives |
|---|---|---|
| 1 | FAU_GEN.1 | O.AUDREC, O.ACCOUN |
| 2 | FAU_SAR.1 | O.AUDREC |

| | | |
|---|---|---|
| 3 | FAU_STG.3 | O.AUDREC |
| 4 | FIA_AFL.1 | O.IDAUTH |
| 5 | FIA_ATD.1 | O.IDAUTH |
| 6 | FIA_UID.2 | O.IDAUTH, O.ACCOUN |
| 7 | FIA_UAU.2 | O.IDAUTH, O.ACCOUN |
| 8 | FDP_IFC.1 (1) | O.MEDIAT |
| 9 | FDP_IFC.1 (2) | O.MEDIAT |
| 10 | FDP_IFC.1 (3) | O.MEDIAT |
| 11 | FDP_IFF.1 (1) | O.MEDIAT |
| 12 | FDP_IFF.1 (2) | O.MEDIAT |
| 13 | FDP_IFF.1 (3) | O.MEDIAT |
| 14 | FMT_MSA.3 | O.MEDIAT, O.SECSTA |
| 15 | FDP_RIP.1 | O.MEDIAT |
| 16 | FPT_RVM.1 | O.SECSTA |

A discussion of the rationale for the mapping is provided for each security objective below.

**O.IDAUTH:** The TOE must uniquely identify and authenticate the claimed identity of all users, before granting a user access to TOE functions or, for certain specified services, to a connected network.

O.IDAUTH is mapped to FIA_AFL.1, FIA_ATD.1, FIA_UID.2, FIA_UAU.2.

- FIA_AFL.1 Authentication failure handling

  This component exists to specify action after some number of unsuccessful authentication attempts. It ensures that users cannot endlessly attempt to authenticate without leaving no trace in the log files.

- FIA_ATD.1 User attribute definition

  This component exists to provide users with attributes to distinguish one user from another, for accountability purposes and to associate the role chosen in FMT_SMR.1 with a user.

- FIA_UID.2 User identification before any action

  This component ensures that the user identify himself (when required) before any information is passed though the TOE. The Basic authentication method provides this functionality for the users.

- FIA_UAU.2 User authentication before any action

  This component ensures that users are identified when necessary. When authentication is required it must occur before any data is passed though the TOE. Basic authentication method provides this functionality for the users. Note, that

firewall administrators are not authenticated by the TOE itself. This is done by the environment (underlying operating system).

**O.MEDIAT:** The TOE must mediate the flow of all information from users on a connected network to users on another connected network, and must ensure that residual information from a previous information flow is not transmitted in any way.

O.MEDIAT is mapped to FDP_IFC.1 (1), FDP_IFC.1 (2), FDP_IFC.1 (3), FDP_IFF.1 (1), FDP_IFF.1 (2), FDP_IFF.1 (3), FMT_MSA.3, FDP_RIP.1.

- FDP_IFC.1 Subset information flow control (1)

  This component identifies the entities involved in the UNAUTHENTICATED information flow control SFP (i.e., users sending information to other users and vice versa).

- FDP_IFC.1 Subset information flow control (2)

  This component identifies the entities involved in the UNAUTHENTICATED_APPL information flow control SFP (i.e., users sending information on application level to other users and vice versa).

- FDP_IFC.1 Subset information flow control (3)

  This component identifies the entities involved in the AUTHENTICATED information flow control SFP. Users who want to use one of these services must be authenticated at the TOE.

- FDP_IFF.1 Simple security attributes (1)

  This component identifies the attributes of the users sending and receiving the information in the UNAUTHENTICATED SFP, as well as the attributes for the information itself. Then the policy is defined by saying under what conditions information is permitted to flow.

- FDP_IFF.1 Simple security attributes (2)

  This component identifies the attributes of the users sending and receiving the information in the UNAUTHENTICATED_APPL SFP for data transferred on application level, as well as the attributes for the information itself. Then the policy is defined by saying under what conditions information is permitted to flow.

- FDP_IFF.1 Simple security attributes (3)

  This component identifies the attributes of the users sending and receiving the information in the AUTHENTICATED SFP, as well as the attributes for the information itself. Then the policy is defined by saying under what conditions information (data sent on application level) is permitted to flow.

- FMT_MSA.3 Static attribute initialization

  This component ensures that there is a default deny policy for the information flow control security rules. The TOE ensures that by default all traffic through the TOE is denied.

- FDP_RIP.1 Subset residual information protection

  This component ensures that neither information that had flown through the TOE nor any TOE internal data are used when padding is used by the TOE for information flows. Future sessions will not contain residual information of previous sessions in padding data.

**O.SECSTA:** Upon initial start-up of the TOE or recovery from an interruption in TOE service, the TOE must not compromise its resources or those of any connected network.

O.SECSTA is mapped to FMT_MSA.3 and FPT_RVM.1.

- FMT_MSA.3 Static attribute initialization

  This component ensures that there is a default deny policy for the information flow control security rules. The TOE ensures that by default all traffic through the TOE is denied.

- FPT_RVM.1 Non-bypassability of the TSP

  This component ensures that upon initial start-up of the TOE or recovery from an interruption in TOE service, the TOE security function are invoked before any information can be transmitted through the TOE.

**O.AUDREC:** The TOE must provide a means to record a readable audit trail of security-related events, with accurate dates and times, and a means to search and sort the audit trail based on relevant attributes. The TOE must provide that the audit trail is readable and no records are left because of not enough storage capacity.

O.AUDREC is mapped to FAU_GEN.1, FAU_SAR.1, and FAU_STG.3.

- FAU_GEN.1 Audit data generation

  This component outlines what data must be included in audit records. Audit data generated by the TOE is stored in different log files as stated in SF3. When applicable, information about the identified user is stored in the log files.

- FAU_SAR.1 Audit review

  This component ensures that the user can interpret the recorded information. The log files are stored in human readable form (clear text) by the TOE and can be reviewed by any suitable tool running on the underlying operating system.

- FAU_STG.3 Action in case of possible audit data loss

  This component ensures that the user is alerted in case of possible audit data loss.

**O.ACCOUN:** The TOE must provide user accountability for information flows through the TOE and for authorized administrator use of security functions related to audit.

O.ACCOUN is mapped to FAU_GEN.1, FIA_UID.2, FIA_UAU.2.

- FAU_GEN.1 Audit data generation

  This component outlines what data must be included in audit records. Audit data generated by the TOE is stored in different log files as stated in SF3. When applicable, information about the identified user is stored in the log files.

- FIA_UID.2 User identification before any action

  This component ensures that the user identify himself (when required) before any information is passed though the TOE. The Basic authentication method provides this functionality for the users.

- FIA_UAU.2 User authentication before any action

  This component ensures that users are identified when necessary. When authentication is required it must occur before any data is passed though the TOE. Basic authentication method provides this functionality for the users. Note, that firewall administrators are not authenticated by the TOE itself. This is done by the environment (underlying operating system).

**Table 8.5 – Security Objective to Functional Component of the environment Mapping**

| # | **Objective (Environment)** | **Functional Component** |
|---|---|---|
| 1 | OE.OS | FPT_STM.1, FAU_SAR.2, FAU_SAR.3, FAU_STG.1, FMT_SMR.1, FMT_MSA.1 (1), FMT_MSA.1 (2), FMT_MSA.1 (3), FCS_COP.1 |

**Table 8.6 – Functional Requirements to Objectives for the environment Mapping**

| # | Functional Requirement | Objective |
|---|---|---|
| 1 | FPT_STM.1 | OE.OS |
| 2 | FAU_SAR.2 | OE.OS |
| 3 | FAU_SAR.3 | OE.OS |
| 4 | FAU_STG.1 | OE.OS |
| 5 | FMT_SMR.1 | OE.OS |
| 6 | FMT_MSA.1 (1) – UNAUTHENTICATED SFP | OE.OS |
| 7 | FMT_MSA.1 (2) – UNAUTHENTICATED_APPL SFP | OE.OS |
| 8 | FMT_MSA.1 (3) – AUTHENTICATED SFP | OE.OS |
| 9 | FCS_COP.1 | OE.OS |

A discussion of the rationale for the mapping is provided for each objective below.

**OE.OS:** The OS has to implement functions for: reliable time stamp, file protection and tools for audit review that can be used by the TOE.

O.OS is mapped to FPT_STM.1, FAU_SAR.2, FAU_SAR.3, FAU_STG.1, FMT_SMR.1, FMT_MSA.1 (1), FMT_MSA.1 (2), FMT_MSA.1 (3), FCS_COP.1.

- FPT_STM.1 Reliable time stamps

  This component ensures that the date and time on the TOE is dependable. This is important for the audit trail to trace recorded audit data.

- FAU_SAR.2 Restricted audit review

  This component ensures that audit log files can be reviewed by authorized persons only. The operating system restricts access to protected log files to authorized persons.

- FAU_SAR.3 Selectable audit review

  This component ensures that a variety of searches and sorts can be performed on the audit trail. Additional audit tools are supplied by the underlying operating system.

- FAU_STG.1 Protected audit trail storage

  This component ensures that the audit data cannot be deleted by unauthorized persons. The operating system restricts access to protected log files to authorized persons.

- FMT_SMR.1 Security roles

  Each of the CC class FMT components in this Security Target depend on this component. It requires the ST writer to choose roles. The role “authorized administrator” is defined by this component and ensures that the underlying operating system is responsible for implementing such role.

- FMT_MSA.1 Management of security attributes (1) – UNAUTHENTICATED SFP

  This component ensures the TSF enforces the UNAUTHENTICATED SFP to restrict the ability to change specified security attributes that are listed in section FDP_IFF1.1 (1).

- FMT_MSA.1 Management of security attributes (2) – UNAUTHENTICATED_APPL SFP

  This component ensures the TSF enforces the UNAUTHENTICATED_APPL SFP to restrict the ability to change specified security attributes that are listed in FDP_IFF1.1 (2).

- FMT_MSA.1 Management of security attributes – AUTHENTICATED SFP

  This component ensures the TSF enforces the AUTHENTICATED SFP to restrict the ability to change specified security attributes that are listed in section FDP_IFF1.1 (3).

- FCS_COP.1 Cryptographic operation

  This component ensures that SSL encryption can be used for

- o securing a Basic authentication and
- o establishing an SSL bridging connection.

### 8.2.2 Security Assurance Requirements Rationale

EAL 2 was selected because the TOE requires a low to moderate level of independently assured security and requires a thorough investigation of the TOE and its development without substantial re-engineering. EAL 2 provides assurance by an analysis of the security functions, using a functional and interface specification, guidance documentation, and the high-level design of the TOE to understand the security behavior. The analysis is supported by independent testing of the TOE security functions, evidence of developer testing based on the functional specification and high-level design, selective independent confirmation of the developer test results, strength of function analysis, and evidence of a developer search for obvious vulnerabilities.

### 8.2.3 Strength of Function Rationale

The TOE is expected to be designed to protect against “low” attack potential because of the chosen Basic authentication method (which can be secured by SSL). Thus, based on the CEM Annex B, Table B.2, the strength of function is SOF Basic. The strength of function claim is minimum and adequate. The strength of function only applies to non-cryptographic mechanisms. The SOF requirement applies to the identification and authentication functionality for the TOE. SF2 and SF3 do not apply to non-cryptographic, probabilistic or permutational mechanisms. Because of the selected security objectives, which do not expect medium and high potential attackers, the chosen SOF claim of SOF Basic is adequate.

### 8.2.4 Dependency Rationale

**Table 8.7 – TOE Functional Requirements Dependencies**

| # | Requirement (SFR TOE) | Dependencies |
|---|---|---|
| 1 | FAU_GEN.1 | FPT_STM.1 |
| 2 | FAU_SAR.1 | FAU_GEN.1 |
| 3 | FAU_STG.3 | FAU_STG.1 |
| 4 | FIA_AFL.1 | FIA_UAU.1 (covered by FIA_UAU.2) |
| 5 | FIA_ATD.1 | none |
| 6 | FIA_UID.2 | none |
| 7 | FIA_UAU.2 | FIA_UID.1 |
| 8 | FDP_IFC.1 (1) – UNAUTHENTICATED SFP | FDP_IFF.1 (1) |
| 9 | FDP_IFC.1 (2) – UNAUTHENTICATED_APPL SFP | FDP_IFF.1 (2) |

| | | |
|---|---|---|
| 10 | FDP_IFC.1 (3) – AUTHENTICATED SFP | FDP_IFF.1 (3) |
| 11 | FDP_IFF.1 (1) – UNAUTHENTICATED SFP | FDP_IFC.1 (1), FMT_MSA.3 |
| 12 | FDP_IFF.1 (2) – UNAUTHENTICATED_APPL SFP | FDP_IFC.1 (2), FMT_MSA.3 |
| 13 | FDP_IFF.1 (3) – AUTHENTICATED SFP | FDP_IFC.1 (3), FMT_MSA.3 |
| 14 | FDP_RIP.1 | none |
| 15 | FMT_MSA.3 | FMT_MSA.1, FMT_SMR.1 |
| 16 | FPT_RVM.1 | none |

**Table 8.8 – Functional Requirements Dependencies for the IT Environment**

| # | **Requirement (SFR Environment)** | **Dependencies** |
|---|---|---|
| 1 | FPT_STM.1 | none |
| 2 | FAU_SAR.2 | FAU_SAR.1 |
| 3 | FAU_SAR.3 | FAU_SAR.1 |
| 4 | FAU_STG.1 | FAU_GEN.1 |
| 5 | FMT_SMR.1 | none |
| 6 | FMT_MSA.1 (1) | FDP_IFC.1 (1), FMT_SMR.1 |
| 7 | FMT_MSA.1 (2) | FDP_IFC.1 (2), FMT_SMR.1 |
| 8 | FMT_MSA.1 (3) | FDP_IFC.1 (3), FMT_SMR.1 |
| 9 | FCS_COP.1 | FCS_CKM.1, FCS_CKM.4, FMT_MSA.2 |

The timestamp is provided by the underlying operating system. So FPT_STM.1 is part of the IT environment.

The TOE does not maintain the role “authorized administrator”. Access control to the TOE is granted by the underlying operating system that also maintains the role “authorized administrator”. So FMT_MSA.1 (1), FMT_MSA.1 (2), FMT_MSA.1 (3), and FMT_SMR.1 have been placed in the IT environment.

The log files are stored in human readable form (clear text) by the TOE and can be reviewed by any suitable tool running on the underlying operating system, so FAU_SAR.3 has been placed in the IT environment.

Access to the log files is restricted to authorized persons by the underlying operating system, so FAU_STG.1 and FAU_SAR.2 have been placed in the IT environment.

Cryptographic support is part of the underlying operating system that provides

- the Crypto API (CAPI) for common cryptographic operations and
- Schannel.dll for SSL related operations.

Dependencies for FCS_COP.1 are not further resolved because these components are part of the environment and handled by the underlying operating system. The IT environment has to ensure that the dependencies are fulfilled. These components are listed in Table 8.9 with a corresponding explanation.

**Table 8.9 – Dependencies of FCS_COP.1 fulfilled by the IT environment**

| | |
|---|---|
| FCS_CKM.1 Cryptographic key generation | The TOE has an interface to the Security Support Provider Interface (SSPI), which enables to access dynamic-link libraries containing common authentication and cryptographic data schemes. The DLLs are called Security Support Providers (SSPs). SSPs make security packages available to applications. A security package maps various SSPI functions to the security protocols specified in the package. The SSPI libraries contain functions which are used to manage and establish secure connections, like cryptographic key generation and destruction. |
| FCS_CKM.4 Cryptographic key destruction | |
| FMT_MSA.2 Secure security attributes | |

## 8.3 TOE Summary Specification Rationale

This chapter shows that the TOE security functions and assurance measures are suitable to meet the TOE Security Requirements.

### 8.3.1 TOE Security Functions Rationale

Table 6.3 in chapter 6 shows that the security functions defined in the TOE Summary Specification address all of the TOE security functional requirements. All security functions are necessary because there is at least one security functional requirement mapped to each security function. The corresponding rationale and the mapping is provided for each security functional requirement within chapter 6.1.

### 8.3.2 Security Requirements are mutually supportive and internally consistent

All security functional requirements are taken from the Common Criteria part 2. The TOE - together with its environment - fulfils all the dependencies defined in the selected SFRs. This shows that the security functions work together so as to satisfy the security functional requirements.

The Table 6.3 shows that all security functional requirements are satisfied by at least one security function. The definitions of the security functional requirements and the assurance components in the preceding chapters demonstrate that mutual support and consistency are given for both groups of requirements. The fact that the SFRs and the assurance requirements support each other and that there are no inconsistencies between these groups is shown in the chapters above.

### 8.3.3 Assurance Measures Rationale

The Table 6.4 in chapter 6 shows how all assurance requirements were satisfied and that there is at least one assurance measure defined in the TOE Summary Specification to meet each of the security assurance requirements.

## 8.4 PP Claims Rationale

This security target is in no compliance with any existing protection profile.

# 9 Appendix

## 9.1 References

[CC] *Common Criteria for Information Technology Security Evaluation*, version 2.1, revision August 1999
*Part 1: Introduction and general model, CCIMB-99-031,*
*Part 2: Security functional requirements, CCIMB-99-032,*
*Part 3: Security Assurance Requirements, CCIMB-99-033*
*Incorporated with interpretations as of 2002-02-28*

[CEM] *Common Methodology for Information Technology Security Evaluation, Part 1: Introduction and general model*, version 0.6, revision 11.01.1997, *Part 2: Evaluation Methodology,* version 1.0, revision August 1999
*Incorporated with interpretations as of 2002-02-28*

[MSISA] *Microsoft Internet Security and Acceleration Server 2000 manual,* Microsoft Corp.

[PP] *Application-level Firewall Protection Profile for Low-Risk Environments*, Version 1.d, U.S. Government, July 20, 1999

## 9.2 Acronyms and Glossary

**Acronyms**

| | |
|---|---|
| API | Application Programming Interface |
| CC | Common Criteria |
| EAL | Evaluation Assurance Level |
| IT | Information Technology |
| MIME | Multipurpose Internet Mail Extensions |
| MMC | Microsoft Management Console |
| OWA | Outlook Web Access |
| PP | Protection Profile |
| SF | Security Function |
| SFP | Security Function Policy |
| SFR | Security Functional Requirement |
| SOF | Strength of Function |
| SPI | Stateful Packet Inspection |
| SSL | Secure Socket Layer |

| | |
|---|---|
| ST | Security Target |
| TOE | Target of Evaluation |
| TSC | TSF Scope of Control |
| TSF | TOE Security Functions |
| TSFI | TSF Interface |
| TSP | TOE Security Policy |

## Glossary

| | |
|---|---|
| access policy | contains “IP packet filters”, “protocol rules” and “site and content rules” |
| Active Directory | Active Directory is a so called Directory Service. It promises to support a single unified view of objects on a network and allows locating and managing resources faster and easier. |
| application filters | Application filters can access the data stream or datagrams associated with a session within the Firewall service and work with some or all application-level protocols. |
| authentication | Authentication is "A positive identification, with a degree of certainty sufficient for permitting certain rights or privileges to the person or thing positively identified." In simpler terms, it is "The act of verifying the claimed identity of an individual, station or originator" [Schou, Corey (1996). Handbook of INFOSEC Terms, Version 2.0. CD-ROM (Idaho State University & Information Systems Security Organization)] |
| base-64 | Encoding scheme for characters |
| Basic authentication | Basic authentication is the standard authentication method for Hypertext Transfer Protocol (HTTP). Though user information is encoded, no encryption is used with basic authentication. |
| broadcast network | A broadcast network (like Ethernet) has a local address for the interface and a broadcast address for the local subnet. |
| Cache mode | one of three installation modes of ISA Server |
| client (computer) set | a set of specific computers |
| credentials | An authentication method used to validate client-to-server and server-to-server communication. Credentials include a user name and a password that is used to validate requests from client computers or from other computers in an array or chain. |
| dynamic filters | Dynamic filters are automatically started by the Firewall service, Web proxy, or SOCKS proxy service. This feature allows the ISA services to automatically open and close communication ports on the external interface when transmission of packets is needed. |
| Feature Pack | A collection of feature extensions for a specific Microsoft product. |
| Firewall mode | one of three installation modes of ISA Server |
| Firewall service | Firewall service is a Windows 2000 service that supports requests from firewall and Secure network address translation (SecureNAT) clients. |
| firewall service log file | contains entries with connection establishments and terminations |

| | |
|---|---|
| HTTP filter | A Hypertext Transfer Protocol (HTTP) filter provided with ISA, that forwards HTTP requests from Firewall and secure network address translation clients to the Web Proxy service. |
| Identification | Identification, according to a current compilation of information security terms, is "the process that enables recognition of a user described to an automated data processing system. This is generally by the use of unique machine-readable names" [Schou, Corey (1996). Handbook of INFOSEC Terms, Version 2.0. CD-ROM (Idaho State University & Information Systems Security Organization)]. |
| inbound | see “incoming” |
| inbound access | Ability to send information from an external network, such as the Internet, to an internal or external network. |
| incoming (traffic) | (traffic) from the external to the internal network interface |
| Integrated mode | one of three installation modes of ISA Server (contains all features of firewall and cache mode) |
| IP packet filters | IP packet filters allow or deny traffic on the packet layer. |
| ISA Server | Microsoft Internet Security and Acceleration Server 2000 |
| Kerberos V5 | authentication protocol (http://www.ietf.org/rfc/rfc1510.txt) |
| load balancing | In a load balancing scheme, requests are forwarded to another server with more capacity, if one server starts to get unavailable because of the number of requests. |
| loopback network | A loopback network allows an application to connect on a local service. |
| MIME | MIME is also used to define content-types used in HTTP transmissions and gives the data type associated with the MIME type name. There are in some cases more than one MIME type in use for a given data type. |
| MMC | Microsoft Management Console – A configuration management tool supplied with Windows 2000 that can be extended with plugins. |
| NTLM | NTLM is an authentication scheme used by Microsoft browsers, proxies, and servers (Microsoft Internet Explorer, Internet Information Server and others). This scheme is also sometimes referred to as the NT challenge/response (NTCR) scheme. |
| outbound | see “outgoing” |
| outbound access | Ability to send information from an internal or internal network to an external network, such as the Internet. |
| outgoing (traffic) | (traffic) from the internal to the external network interface |
| packet filter log file | contains records of packets that were dropped / allowed |
| packet traffic | packet traffic is sent on layer 2 |
| padding | One or more bits appended to data in order to ensure that is contains the required number of bits and bytes. |
| port number | A number that identifies a certain Internet application with a specific connection. |
| Protocol rules | Protocol rules indicate whether a particular protocol is accessible for inbound and outbound communication. |

| | |
|---|---|
| publishing rules | publish virtually any computer on an internal network to the Internet (see Web publishing and Server publishing) |
| remote procedure call (RPC) | A message-passing facility that allows a distributed application to call services available on various computers in a network. Used during remote administration of computers. |
| Secure Sockets Layer (SSL) | A protocol that supplies secure data communication through data encryption and decryption. SSL enables communications privacy over networks. |
| Server publishing | Server publishing allows virtually any computer on an internal network to publish to the Internet. |
| Service Pack | A collection of bug fixes for a specific Microsoft product. |
| Site and content rules | Site and content rules specify which sites and content can be accessed. |
| static filters | Filters that allow packets from other administrator-selected services from the Internet. A static filter is created during configuration of ISA by using the user interface. If IP packet filtering is enabled, the static filter is always on. |
| UUID | Universal Unique Identifier - A UUID is an identifier that is unique across both space and time, with respect to the space of all UUIDs. A UUID can be used for multiple purposes, from tagging objects with an extremely short lifetime, to reliably identifying very persistent objects across a network. |
| W3C | World Wide Web Consortium (W3C) develops interoperable technologies (specifications, guidelines, software, and tools) concerning Web technology (http://www.w3c.org) |
| Web Proxy service | The Web Proxy service is a Windows 2000 service that supports requests from any Web browser. The Web Proxy service works at the application level on behalf of a client requesting an Internet object that can be retrieved using one of the protocols supported by the Web Proxy protocols: File Transfer Protocol (FTP), Hypertext Transfer Protocol (HTTP), and Gopher. The Web Proxy service also supports the Secure HTTP (HTTPS) protocol for secure sessions using Secure Sockets Layer (SSL) connections. |
| Web proxy service log file | stores one line per HTTP request |
| Web publishing | Web publishing publishes Web content to the Internet |