Federal Office
for Information Security

Common Criteria Protection Profile

for

Remote-Controlled Browsers Systems (ReCoBS)

Common Criteria

BSI-PP-0040

Version 1.0 (2008-02-26)

Foreword

This Protection Profile - Remote-Controlled Browsers Systems (ReCoBS) - is issued by Bundesamt für Sicherheit in der Informationstechnik, Germany. It is based on and compatible with the ReCoBS concept previously developed at BSI [6].

The document has been prepared as a Protection Profile (PP) following the rules and formats of Common Criteria version 3.1 Revision 2 [1], [2], [3].

Correspondence and comments to this Protection Profile should be referred to:

CONTACT ADDRESS

Bundesamt für Sicherheit in der Informationstechnik
Postfach 200363
53133 Bonn, Germany

Tel +49 (0)3018 9582-0
Fax +49 (0)3018 9582-5400

Email bsi@bsi.bund.de

Change history

Version	Date	Reason	Remarks
0.9	2007-11-15	Incorporated all pending (minor) changes, sync to R2, yet again improved Figure 2; basis for evaluation	
0.91	2008-01-18	Updated according to comments of evaluation facility datenschutz nord	
0.92	2008-02-06	Further updates in response to comments of evaluation facility and certification body	
0.93	2008-02-11	Final updates in response to comments of evaluation facility	
0.94	2008-02-11	One spot in last update was missed as pointed out by evaluation facility, fixed	
1.0	2008-02-26	Certified version, identical to version 0.94	

Invariants

Name	Value	Display
Current Version	1.0	1.0
Date	2008-02-26	2008-02-26
Classification	unclassified	unclassified
Author(s)	Dr. Helge Kreutzmann	Dr. Helge Kreutzmann

Table of Content

1 PP Introduction 7
1.1 PP Reference 7
1.2 TOE Overview 7
1.2.1 Overview 7
1.2.2 Usage and major security features 8
1.2.3 TOE Type 11
1.2.4 Available non-TOE hardware/software/firmware 11
1.3 Application Note: Further Security Measures 12
1.4 Structure and Conventions 14
2 Conformance Claim 16
2.1 Conformance Claim 16
2.2 Conformance Claim Rationale 16
2.3 Conformance Statement 16
3 Security Problem Definition 17
3.1 Introduction 17
3.2 Assumptions 18
3.3 Threats 20
3.4 Organisational Security Policies 22
4 Security Objectives 23
4.1 Security Objectives for the TOE 23
4.2 Security Objectives for the Environment 24
4.3 Security Objectives Rationale 27
5 Extended Components Definition 28
5.1 Extended Components Rationale 28
6 Security Requirements 29
6.1 Security Functional Requirements for the TOE 29
6.1.1 Flow Control Policy “TOE transmission protocol” 29
6.1.2 FDP_IFC.1 Subset information flow control 29
6.1.3 FDP_IFF.1 Simple security attributes 30
6.1.4 FMT_MSA.1 Management of security attributes 31
6.1.5 FMT_MSA.3(t) Static attribute initialisation 31
6.1.6 FMT_SMF.1 Specification of Management Functions 32
6.1.7 FMT_SMR.1 Security roles 32
6.2 Security Functional Requirements Rationale 32
6.3 Security Assurance Requirements for the TOE 32

6.4 Application Note: Security Requirements for the IT Environment 33
6.4.1 TOE Host Access Policy "AC_HOST" 34
6.4.2 The Integrity Self Test 34
6.4.3 FDP_ACC.2 Complete access control 35
6.4.4 FDP_ACF.1 Security attribute based access control 35
6.4.5 FIA_SOS.1 Verification of secrets 36
6.4.6 FIA_UAU.2 User authentication before any action 36
6.4.7 FIA_UID.2 User identification before any action 36
6.4.8 FMT_MSA.3(h) Static attribute initialisation 37
6.4.9 FMT_SMR.2 Restrictions on security roles 37
6.4.10 FPT_TST.1 TSF testing 38

7 Rationales 39

7.1 Security Objectives Rationale 39

7.1.1 Protection offered by the TOE against the Threats 39
7.1.2 Protection offered by the TOE environment against the Threats 42
7.1.3 Consideration of the assumptions 45
7.2 Security Requirements Rationale 46

7.2.1 Security Functional Requirements Rationale 46
7.2.2 Dependency Rationale 51
7.2.3 Security Assurance Requirements Rationale 53

8 Glossary and Acronyms 54

9 Literature 57

1 PP Introduction

1.1 PP Reference

Title:	Common Criteria Protection Profile for Remote-Controlled Browsers Systems (ReCoBS)
Sponsor:	Bundesamt für Sicherheit in der Informationstechnik (Federal Office for Information Security), Bonn, Germany
Editor:	Dr. Helge Kreutzmann, BSI
CC Version:	3.1
Assurance Level:	EAL3+
General Status:	final
Version Number:	1.0
Registration:	BSI-PP-0040
Keywords:	WWW, Internet, Active Content, Browser, Terminal Server

1.2 TOE Overview

1.2.1 Overview

1 A **Remote-Controlled Browsers System** (**ReCoBS**) is a modular part of a security gateway to enable the almost unlimited access to content on the **World Wide Web** (**WWW**) from a **Local Computer** (**LC**) of a user inside a **Local Network** (**LAN**). At the same time it prevents both the local information of users as well as the local computer and net devices (machines) on the LAN from (negative) effects of malware contained in active content within web pages.

2 In brief, the TOE is a ReCoBS which is intended for comfortable access to WWW content on the Internet without compromising integrity, availability or confidentiality of information in the LAN:

- WWW content can be accessed without severe restrictions (e.g. filtering of active content which severely limits the usability of some WWW content) – "access"
- Access occurs from the Local Computer (LC) of each user (i.e. no dedicated devices/networks for access necessary) – "comfortable"
- Access of WWW content does not impair integrity, availability or confidentiality of information in the Local Network (LAN) – "secure"

3 Compared to other solutions for secure (in the sense of the definition above) WWW access the TOE does not require a dedicated and physically separated network or net devices but rather existing LCs and infrastructure can be reused (in combination with the TOE).

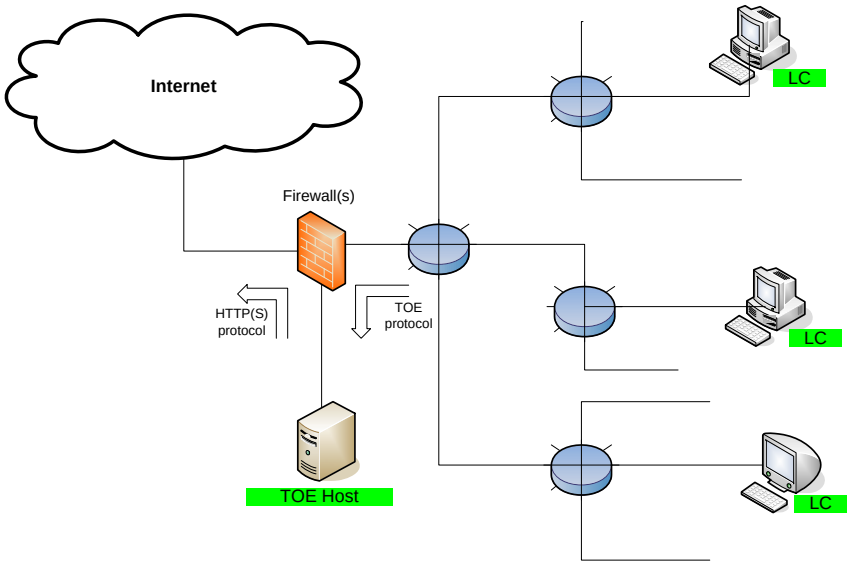

Figure 1: Schematic plot of a ReCoB system (running on systems marked in green). The TOE client is installed on the LC in the LAN, while the TOE server runs on a machine (called TOE host) in the DMZ, i.e. a machine which is separated from both the LAN as well as the Internet by firewalls.

4 *Application note: For the purpose of achieving the aims of the TOE no filtering of WWW content is required. Due to legal or performance reasons some filtering might, however, be required, cf. to Section 1.3 for possible examples.*

5 *Application note: The TOE as presented in this PP is not the only possible solution for achieving "secure" WWW access. Other options with a similar "level of security" usually require a dedicated network (wiring, LCs), i.e. are expensive and space consuming, or are based on strict filtering, i.e. limit the usefulness of WWW access, as a significant amount of WWW content will be inaccessible (e.g. those contained in active content). In brief, many scenarios for WWW access fall short on either of the three points listed in paragraph 2. For an discussion of (other) possible solutions please refer to [6].*

1.2.2 Usage and major security features

1.2.2.1 Idea and aim of the TOE

6 The **Target of Evaluation (TOE)** consists of the TOE server, the TOE protocol and the TOE client. The TOE server runs on one or more machines, called TOE host. The TOE host contains all hard- and software required to execute the TOE server and is situated in the **Demilitarised Zone (DMZ)** as part of the IT environment for the TOE (cf. Section 1.2.4). The term "ReCoBS server" denotes the combination of TOE server and TOE host and is hence not used in this PP. The TOE client runs on the LC (cf. Figure 1). Figure 2 contains a more detailed view (for clarity, all details about the network topology within the LAN are omitted in this figure). The TOE server and TOE client communicate over a certain protocol called "TOE protocol", which is part of the TOE. This protocol passes the firewall infrastructure (which is required to operate the TOE) and traverses the network between the DMZ and the LC.

7 The TOE is thus not a complete firewall but rather a modular part of a security gateway for Internet access intended for secure surfing in the WWW which has to be

integrated into a firewall infrastructure. The basic idea of a ReCoBS is a breach of information flow which transforms the HTML code (including active content) into pure audio-visual[1] information[2]. The increase of security is mainly based on this breach. By separating the execution and the display environment the entire HTTP stream (i.e. HTML code, graphics, PDF files, etc.), including the problematic active content (like ActiveX controls, Java applets, JavaScript programs), does not reach the LCs, only the comparatively harmless representation of this content as pure audio-visual data is transmitted onto the LCs.

8 To achieve this, the users run the TOE client on their LCs in the LAN, which connects to the TOE server (cf. Figure 1) executed on a dedicated TOE host in the DMZ. Each user is able to remotely control one (or more) browsers on the TOE host ("execution environment") from his LC ("display environment") using the TOE protocol. The TOE protocol consists of key presses and mouse events (client to server), audio-visual data (server to client) and optionally limited clipboard exchange. As the TOE server and the browsers run on a TOE host, all code embedded in the HTTP stream, including malware in active content, is executed there as well. Furthermore, access to WWW content is granted only from a TOE host. Hence possible side effects (both intended and unintended) are limited to the TOE host. Since the TOE host fulfils dedicated security requirements the risk of a (temporarily accepted) compromise is greatly reduced. Such a system might be implemented by a specially tailored terminal server, but due to untrusted code running on the TOE host a standard terminal server cannot be used unaltered.

9 This **Protection Profile** (**PP**) defines the security requirements for a ReCoB system. These requirements are specified on a level which enables both manufacturers to develop a wide range of possible implementations and at the same time defines the security requirements precise enough to pass an evaluation according to the **Common Criteria** (**CC**). It is based on (and compatible with) the ReCoBS concept [6] developed by BSI.

1.2.2.2 Intended environment

10 Typical environments for the TOE are companies, (public) authorities or sections thereof where unlimited access to WWW content is required. The TOE is intended to be part of an overall security infrastructure, like firewalls, e-mail scanners etc., which protects against threats from untrustworthy networks and data. The TOE should not be used if – according to a risk analysis - a physically dedicated network with dedicated LCs solely for WWW access is required (e.g. because of highly sensitive or classified data in the LAN).

[1] Some implementations might choose to omit the transmission of audio data.

[2] To allow Copy and Paste a TOE might allow a controlled transfer of pure textual data as well which can be enabled or disabled depending on the requirements of the organisation using the TOE.

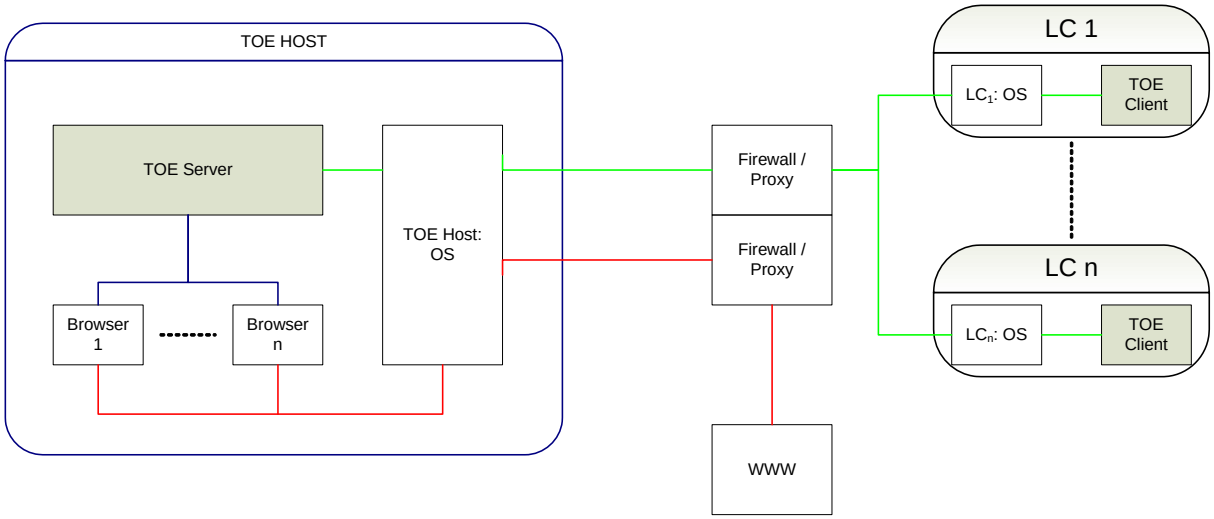

Figure 2: Schematic information flow of data from the WWW to the LC. TOE parts are denoted in grey, the environment in white. The individual browsers communicate with the WWW using HTTP(S) (denoted in red), while the TOE server communicates with the TOE client using the TOE protocol (denoted in green). The information breach (denoted in blue) occurs on the TOE server. Only components relevant for the TOE are displayed, e.g. implementations will contain further devices (e.g. routers, switches).

1.2.2.3 Basic description of the TOE functionality

11 As depicted in Figure 2 the browsers run on the TOE host. Embedded active content – along with all other content - can be used without limitations on the TOE host. The representation of this content is then transmitted as pure (audio-)visual data via the TOE protocol to the TOE client, where the graphical (and audio) representation of the content is displayed. Additionally the TOE may offer the possibility for the user to copy a textual representation of the content from the TOE server to a clipboard on the LC. In converse, the user controls the browser remotely from his LC using the TOE client. This control is achieved by transmitting key presses and mouse events from the TOE client via the TOE protocol to the TOE server. Additionally the TOE may offer the possibility for the user to paste textual content from a clipboard on the LC to the TOE server. Thus execution and display/control of (active) content are separated.

12 Integrity, availability and confidentiality of data in the LAN is ensured by the TOE in conjunction with the firewall infrastructure, as the breach in information processing by usage of the TOE protocol prevents any code (including malware) from the WWW from reaching net devices in the LAN[3], and any data from the LAN reaching the browser (unless explicitly entered by the user). The separation of information between the LAN and the TOE server includes identity information as well: to avoid that malware obtains identity information on the TOE host (e.g. usernames and passwords) and an attacker subsequently uses this identity information to open a direct connection inside the LAN (e.g. via remote login) the TOE host operates an identification and

[3] The TOE protects against content from the WWW obtained by direct access, typically using a browser. As stated above, further measures are required for controlling other paths of data transport, e.g. e-mail, into the LAN.

authentication system independent from any **identification and authentication (I&A)** in the LAN (e.g. on the LC) and ensures that no (trivial) mapping of user attributes (like e-mail addresses) used on the TOE server to those used inside the LAN is possible. Additional organisational measures have to be employed to avoid credentials (e.g. passwords) used on net devices from reuse on the TOE host.

13 To allow for Copy and Paste the TOE may offer the user the additional possibility to transfer pure text from WWW pages into the clipboard on the LC and individual textual contents of the clipboard on the LC after individual confirmation by the user as pure text to the TOE server. Both directions – if available - can be separately enabled by an administrator of the TOE and default to off.

14 In the concept described in this PP the effects of malware are thus limited to the TOE host. To reduce the likelihood of possible security breaches on the TOE host, several measures are employed there (cf. Section 1.2.4 for details).

1.2.3 TOE Type

15 Firewall component for secure WWW access

1.2.4 Available non-TOE hardware/software/firmware

16 As detailed in Section 1.2.2.1 the TOE consists of three parts. First the TOE server, which is executed on the TOE host. The TOE host consists of all the hard- and software required to run the TOE server, is situated in the DMZ and forms part of the IT environment for the TOE. Secondly the TOE client, which is executed on the LC. The LC includes all the hard- and software required to run the TOE client (but may include more hard-/software). The LC is located inside the LAN and is part of the IT environment as well. Both the TOE server as well as the TOE client are applications on their respective machines (TOE host respective LC). Due to performance reasons the implementation may choose to operate multiple instances of TOE servers on multiple TOE hosts (i.e. a clustered solution).

17 The DMZ is separated by firewalls from the LAN. The TOE server and client communicate via the TOE protocol which passes the firewalls. Thus both firewalls and the LAN (with the additional connection to/from the DMZ) form also a part of the IT environment of the TOE.

18 Besides offering an independent I&A (cf. paragraph 12) several measures to support the TOE are implemented on the TOE host. First, the TOE host is limited to the task of running the TOE server and those services necessary to achieve the security objectives of this PP (e.g. I&A), i.e. only necessary services and programmes are available on the TOE host[4]. Secondly, the TOE host is configured restrictive to limit the effect of malware, i.e. programmes may only access ressources (e.g. network ports, configuration files) necessary to allow access to WWW content as defined in Section

[4] Typically the developer provides a statement which demonstrates why every service/program is necessary. For services required by this PP (e.g. I&A) a reference to the relevant requirement in this PP usually suffices.

1.2.1[5], deploys measures to ensure that unaltered binaries (e.g. unmodified browser binaries) are executed after log-in and ensures the termination of **all** programmes started by the user during log off of this user. Finally, the entire TOE host (including the TOE server) is reinitialised at fixed time intervals, including integrity checks, to ensure that new attack vectors of malware are at least detected at these times. So, in conclusion, malware has limited access to resources on the TOE host, is terminated at log off and, if vulnerabilities not known at the time of the TOE's certification are exploited, is at least terminated and detected during the regular reset / check cycles.

19 *Application note: The TOE is designed around a special limited kind of terminal server. As discussed in detail below, however, many popular terminal servers are not suitable as base for the TOE as these products offer a "rich" protocol between client and server which contains several channels for malware on the TOE to access or manipulate data on the LC. If an implementation chooses to deploy an existing terminal server product within the TOE then it has to be verified that **only** the functionality described in this PP is present.*

1.3 Application Note: Further Security Measures

20 *The aim of the TOE is to protect information within the LAN against unauthorised access from active content from the Internet. Depending on the implementation scenario further security requirements might be necessary to achieve other related security objectives. These security requirements are not a part of this PP and are given only as examples for users and authors of STs; if implemented by a developer they should remain optional (i.e. depending on the policy of the organisation using the TOE it should be possible to permanently disable any additional functionality provided for these requirements):*

a) Availability of TOE data: To protect any browsing-relevant user data[6] stored on the TOE (e.g. bookmarks) against loss additional functionality should be implemented. Typically a backup and restore mechanism could be provided, with an additional guide for an administrator to properly administer this mechanism. It has to be taken into account that both hardware failures as well as manipulation by malware can be the cause for data loss. Please note that such a mechanism must not open up any channel from the TOE host into the LAN, i.e. it has to be implemented independently of any backup mechanism existing in the LAN.

b) Availability of the TOE server: To ensure the availability the TOE server could either run in parallel on several TOE hosts - such that load is distributed evenly over all TOE hosts and failure of a single TOE host does not cause failure of the entire TOE server cluster - or additional functionality on the TOE host could limit the resource requirement of each session such that no user can completely block the machine. The latter could include functionality to limit memory usage, CPU usage, bandwidth usage and so on. Please note, however, that some usage pattern

[5] This limitation is sometimes denoted as „hardened".

[6] *The author of an ST has to properly define what kind of user data occurs in his TOE.*

might include high resource requirements (e.g. handling of large objects will cause large memory requirements).

c) *Data minimisation on TOE host: To further avoid disclosure of data used inside the LAN an additional policy for the organisation where the TOE is deployed and for the TOE users should be set up. This policy should oblige all users to only transmit those data to the TOE server which are absolutely necessary for accessing the WWW content. For example using web e-mail or translation services with data from the LAN should be prohibited. This policy should be accompanied by appropriate training of the users.*

d) *Prevention of WWW misuse: To prevent users from abusing their WWW access (e.g. by accessing illegal or inappropriate content[7]) a filtering WWW proxy with mandatory login (to prevent pseudo anonymity) should be implemented. This filtering should not occur on the TOE host, however, since otherwise malware could interfere with the filtering. If the legal/organisational requirements are less strict, this aim can also be achieved by an organisational policy regarding WWW usage. Additionally or alternatively the proxy could log all WWW access to prevent the TOE from being used as "anonymizer" (i.e. to assign each WWW access (especially those not wanted) to an individual user).It should be noted that logging and filtering of WWW access might interfere with privacy requirements in certain jurisdictions.*

e) *Confidentiality (external): To prevent eavesdroppers controlling machines outside the DMZ/LAN from obtaining sensitive information an additional policy for the organisation where the TOE is deployed and for the TOE users should be set up. This policy should oblige all users to carefully monitor what kind of information they transmit to WWW sites, e.g. what kind of WWW sites they visit and what kind of information they search for. This security measure should be supported by the mandatory use of anonymisation services [7].*

f) *Confidentiality (internal): To prevent personnel with physical access to LCs or other net devices (including cabling) from eavesdropping activities during the time of a session[8] ("sniffing on the wire" or "mirroring[9] of sessions input/output") either the devices and cabling can be physically protected or all content of the traffic on the network in the LAN and into the DMZ can be encrypted. In the latter case, however, the firewall and the proxies still need to be able to monitor and modify unencrypted content and hence both the firewall and the TOE host (including their respective cabling) should be physically protected in any case.*

g) *Monitoring of the TOE host: To detect exploit (attempts) and/or to monitor usage (e.g. to enforce policies of the organisation where the TOE is deployed) the TOE*

---

[7] *The definition of "illegal" and "inappropriate" depends heavily on the jurisdiction and the organisation operating the TOE.*

[8] *Similar measures would be necessary if integrity instead of (or additionally to) confidentiality needs to be preserved (either because of manipulations or because of errors during transmission).*

[9] *Some protocols include mirroring of sessions input/output, i.e. directing both input and output on the client side to several machines in parallel (e.g. for user support). To achieve this security objective such functionality must not be present in the TOE protocol.*

as well as the TOE host should offer logging facilities. It is strongly recommended that guidance is provided as well, how to set up the logging (e.g. possible legal requirements) and how and when to interpret the logged data. Depending on the jurisdiction the logging might have to comply to certain legal requirements, e.g. privacy laws.

h) Download and Printing: To fully utilise the WWW content, additional post processing of the data obtained from the WWW might be necessary. This mainly includes the possibility to print and to save content (i.e. to download). It is important for manufacturers of ReCoB systems to ensure that these functionalities do not violate the security objectives of the TOE, e.g. that it is not possible to open up separate channels from the TOE server to the LC for downloads or printing or to implement some kind of „shared storage" (area of concurrent access for both LC and TOE host) between the TOE server and the TOE client[10]. Also the manufacturer has to provide guidance how to estimate and reduce the risk that downloaded content contradicts the security objectives of this PP.

21 *The TOE is a part of the firewall infrastructure of an organisation. To prevent malware from using other means to reach net devices on the LAN (e.g. by using e-mail, USB sticks, mobile devices from outside the LAN) further security infrastructure (e.g. virus scanners) is necessary.*

1.4 Structure and Conventions

22 A PP is a structured description of a certain IT security problem along with a general but precise description of a **Target of Evaluation** (**TOE**) to counteract or mitigate theses security issues. To fulfil the requirements of the **Common Criteria** (**CC**) the PP needs to follow a certain structure. The most important parts of a PP are the PP Introduction (Chapter 1), the Security Problem Definition (Chapter 3), the Security Objectives (Chapter 4), the Security Requirements (Chapter 6), and the Rationales (Chapter 7).

23 Chapter 1 (PP Introduction) contains general information about the TOE, e.g. intended usage and environment, requirements for non-TOE hard- and software and description of assets. This Chapter is a prerequisite for understanding the security requirements and is intended for consumers to estimate the usefulness of the TOE for their needs. It is important to note, however, that a PP does not describe a certain implementation of

[10] *To implement printing and downloading, the TOE host could be connected to a dedicated printer or printer server. Another solution could be to send the print job or the downloaded data respectively by e-mail to the external mail server of the organisation using the TOE. After ordinary processing on the mail server (e.g. virus scanning) the data could then be transferred to a server inside the LAN (like any other e-mail) where a mapping function would determine the recipient (either the user or the printer of a user) and forward it to the recipient. Alternatively downloads could be stored in a dedicated area in the DMZ and a dedicated service, completely independent of the TOE server and TOE host, could allow access by a special file transfer protocol (only initiated from within the LAN) to this data after appropriate security clearance (e.g. manual by the administrator, by policy, after automated checks) has happened. Further solutions are possible as well, as long as they don't contradict the security objectives of this PP.*

a product, but rather a class of products with certain common features. Thus the term TOE in this PP refers to any member of this class.

24 Chapter 3 (Security Problem Definition) details both the assumptions for the user and the environment (which can be understood as requirements for the deployment of the TOE) as well as the threats and **organisational security policies** (**OSP**) which list the threats the TOE counteracts or mitigates and the policies it enforces.

25 Chapter 4 (Security Objectives) details independent of a certain implementation how the TOE counteracts the threats and fulfils the OSPs. Further the security objective for each assumption regarding TOE usage is described.

26 Chapter 5 (Extended Components Definition) does not contain any definition of an additional functional requirement as all functional requirements for this PP are taken from the CC [2].

27 In Chapter 6 (Security Requirements) both the functional requirements regarding the TOE as well as the assurance requirements are defined.

28 Chapter 7 (Rationales) establishes that the PP is a complete and consistent set of IT security requirements and that a TOE which implements the requirements of this PP fulfils the security objectives completely.

29 Finally Chapter 8 contains a list of Glossary and Acronyms and Chapter 9 lists all references cited within the PP.

30 To provide a clear structure within this PP, certain formatting is associated with certain content. Texts written in *italics* are application notes which are not relevant for evaluation of this PP but rather aid the reader and ST authors to better understand possible options or provide additional examples. Emphasised text is written in **bold** or underlined, e.g. abbreviations when introduced. In Chapter 6 certain text blocks from the CC are used. Some words within this texts are printed in SMALL CAPS to denote that they have been altered (refined, operations completed) compared to the original CC text (cf. page 29 for further details).

2 Conformance Claim

2.1 Conformance Claim

31 This protection profile claims conformance to

- Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; Version 3.1 Revision 1, September 2006
- Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Requirements; Version 3.1 Revision 2, September 2007
- Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements; Version 3.1 Revision 2, September 2007

32 as follows:

- Part 2 conformant
- Part 3 conformant
- Package conformant to EAL3 augmented with ALC_CMS.4 and ALC_FLR.3.

33 This PP does not claim conformance to any other PP.

2.2 Conformance Claim Rationale

34 This PP claims conformance to CC part 2 and 3. As no SFRs or SARs were added this PP is conformant to CC part 2 and 3. Further this PP claims conformance to the package EAL3 augmented with ALC_CMS.4 and ALC_FLR.3. As EAL3, ALC_CMS.4 and ALC_FLR.3 do not contain any uncompleted operations and both ALC_CMS.4 and ALC_FLR.3 do not contain any dependencies this conformance is satisfied (cf. also Section 6.2).

35 Since the PP does not claim conformance to any PP no rationale is required.

2.3 Conformance Statement

36 Security targets or other PPs wishing to claim conformance to this PP can do so as strict PP-conformance. Demonstrable PP-conformance is not allowed for this PP.

3 Security Problem Definition

3.1 Introduction

Assets

37 The assets can be distinguished into primary and secondary assets. The main aim of this TOE is to protect the primary assets against manipulation and eavesdropping, as well as to avoid **Denial of Service** (**DOS**) attacks on them. The primary assets are:

- Data stored on machines in the LAN
- Data or information stored or transported in proximity to machines or devices in the LAN, e.g. printed information within the range of a camera connected to a machine in the LAN, spoken words within the range of a microphone connected to a net device in the LAN

38 Secondary assets are themselves of no value, but the possession or control of these assets enables or eases access to primary assets. Therefore these assets need to be protected as well.

- Credentials (i.e. authentication attributes like passwords) used on the LAN and TOE
- Security attributes (e.g. access permissions) on the TOE

39 Application note: The manufacturer of the TOE host should ensure in general that no information is required on the TOE host which contains details of the structure of attributes assigned to persons or net devices inside the LAN, e.g. user names should be sufficiently different on the TOE than those used in the LAN, e-mail addresses used inside the LAN might not be obtained from the TOE (host) and so on. Even if an attacker obtains such information on the TOE host he should not be able to transform this information to the form used in the LAN (e.g. map user names on the TOE to user names on LCs or to e-mail addresses of users). Manufacturers should further include guidance for an administrators to alert him on this problem and to avoid leakage of such information on the TOE host.

Subjects

40 **Administrator:**
A person who administers the TOE host and who is able to access the TOE on a dedicated service interface to

1. add/remove users on the TOE
2. change security attributes of the **TOE Security Functions** (**TSF**)

41 Application note: Some TOEs might split the role of the administrator into several dedicated roles. Also several persons might fulfil the role of the administrator. In these cases all statements of the PP are valid for all persons acting in any of theses roles.

42 Application note: The possible security attributes depend on the implementation; possible examples are the (de-)activation of the Copy and Paste channels, default browser settings or resource limits for users.

43 **User:**
A person at an LC authorised by an administrator to access the WWW via the TOE. This implies that the user is able to use the TOE client to connect to the TOE server ("initiate a session") and use a WWW browser to connect to WWW sites. He is further able to configure and use plugins required for display of special content and to interact with active content.

44 **Attacker:**
A person who is neither a user nor an administrator and has no physical access to any net device in the LAN or DMZ, i.e. he can only attempt to access the TOE or net device on the LAN or in the DMZ from outside the LAN and DMZ (typically from a machine on the internet).

45 The attacker is able to manipulate certain WWW sites in any way desired by him and to place any programme, including malware, on these sites, e.g. as active content in WWW sites. Those programmes may be taken from public sources (e.g. sites on the WWW) and might include source code or may be developed specifically by the attacker for this attack. He is also able to spoof other electronic communication media like e-mail when in contact with users or administrators. The attacker is not able to physically access any net device in the LAN or the DMZ. The aim of the attack could be eavesdropping of (sensitive) information, manipulation of data (including configuration) on net devices inside the LAN or preventing access to data and services in the LAN. The attacker might be motivated financially or ideationally.

46 *Application note: The TOE protects against attackers from the Internet without physical access to any machine in the DMZ/LAN. If the attacker might have physical access to machines in the LAN or DMZ then additional security objectives will need to be fulfilled. Please refer to Section 1.3 for examples of additional security measures.*

3.2 Assumptions

47 **A.Firewall**[11]:
The TOE client runs on LCs inside the LAN. The TOE host is located in the DMZ and the TOE server runs thereon. Both the connection between the TOE host (situated in the DMZ) and any net device in the LAN as well as the connection between the TOE host and the Internet are separated by a firewall (cf. Figure 1). This firewall operates both on incoming as well as on outgoing traffic and includes network proxies, which operate on the transport (e.g. Internet Protocol) and additionally on the application layer for HTTP(S). The following rules are enforced by the firewall:

1. Connections from net devices in the LAN (including the LCs) to the TOE host can only use a port dedicated for the operation of the TOE server.
2. Only TOE clients running on net devices in the LAN can open connections to the TOE host.
3. It is impossible for the TOE host to initiate connections to net devices inside the LAN.

[11] The term "firewall" is used here rather generically in the sense of a security gateway, cf. the definition in Sec. 8.

4. It is impossible for net devices inside the LAN to connect to content on the WWW directly, i.e. bypassing the TOE.
5. Only the TOE protocol as defined in paragraph 79 and 82 can be used in connections between the TOE host and a TOE client in the LAN.

48 *Application note: The detailed setup of the firewall infrastructure depends on the individual needs of the organisation operating the LAN. Additional measures as detailed in Section 1.3, e.g. a web proxy with filtering between the TOE host and the Internet, might be implemented or encryption could be used between the firewall and the LC[12]. Furthermore it is strongly suggested to use an application level proxy also for the TOE protocol, if available.*

49 *Application note: The preceding assumption does not preclude the possibility for a dedicated (local) service port on the TOE host for the administrator. If such service port is included in the TOE host, it has to be ensured that it is sufficiently secured, e.g. identification/authentication is sufficiently strong, and that A.Firewall still remains valid, e.g. no access from a net device in the LAN onto this service port is possible.*

50 A.LC:
The TOE clients - running on LCs inside the LAN - will not be manipulated by users or software on the LC.

51 *Application note: If the LC were untrustworthy different security requirements would occur. Please cf. to Chapter 1.3 and especially paragraph 21 for a discussion.*

52 **A.Admin**:
The administrator is trustworthy, proficient and does not use this role to access WWW content.

53 **A.Authentication**:
The users do not reuse any identification and authentication attribute (credential) on the TOE which has been used on any net device within the LAN.

54 *Application note: The assumption cannot be stated as a requirement for the TOE or the IT environment, as the authentication routines of the TOE are completely separated from any authentication routine in the LAN.*

55 *Application note: Authors of STs and developers are strongly urged to implement an authentication scheme which does not rely on explicitly entered identification (e.g. login names) and authentication attributes (credentials like passwords, PINs) but rather uses some kind of pre-shared secret (distributed in advance by the administrator) which, upon login on the LC, is automatically available for authentication on the TOE without user intervention. Typically the login procedure on the LC would decrypt the authentication attribute and provide it to the TOE client. The TOE client would send this attribute as if it were entered by the user to the TOE host where it is verified. Such a mechanism would strongly support A.Authentication by reducing the possibility for users to reuse a credential from a net device in the LAN on the TOE host.*

[12] *On the firewall itself, however, the communication between the TOE host and TOE client must be unencrypted to allow the proxy between the TOE host and TOE client to operate.*

56 *Application note: Reuse of identification or authentication attributes includes (simple) mappings from internal used attributes to attributes used on the TOE (host), e.g. replacing all lower case letters in a login name by upper case letters or switching the order of the first and last name.*

57 Since the TOE server is intended to support simultaneous sessions of multiple users it offers an exposed target for sniffing data transmitted to and from the internet. Further any additional software could be used to attempt to bypass security functionality on the TOE host. To reduce the number of possible vulnerabilities in the IT environment, i.e. the number of flaws in the IT environment on the TOE host which could be exploited, only programmes required for the TOE host and TOE server for proper operation are assumed to be installed:

58 **A.Minimal:**
Only programmes required for the operation of the TOE are available on the TOE host (e.g. TOE server, web browser, web browser extensions).

59 *Application note: Should the author of an ST decide to drop this assumption the manufacturer of this TOE has to prove that in his IT environment on the TOE host any additional software (i.e. software not necessary for the proper operation of the TOE) cannot be manipulated, i.e. cannot be used to intercept, manipulate or eavesdrop any TOE related programme, neither direct nor indirect (e.g. by manipulating the data the TOE operates on or by manipulating security attributes on the TOE host).*

60 *Application note: Depending on the specific TOE host used, also the programmes available (e.g. the web browser(s), plugins, ...) should be kept current, at least in respect to security updates provided by the manufacturer or trusted sources. It is suggested that the manufacturer of the TOE provides guidance for the administrator on how he can ensure this update process. Due to different delivery procedures for security/functionality updates of different manufacturers and due to different user expectations (e.g. stable and tried vs. bleeding edge software) no requirements regarding this update procedure are made in this PP.*

3.3 Threats

61 The TOE protects against threats originating from attackers as defined in Section 3.1.

62 For the purpose of the PP it is not relevant how malware is transported on the TOE host. In the following list of threats the malware is therefore assumed to be already on the TOE host. For this PP it is also irrelevant whether the malware runs completely autonomously on the TOE host or whether it is controlled remotely by the attacker.

63 *Application note: The threats described in this PP focus on attacks via WWW content (i.e. malware embedded in web pages). It should be noted, however, that malware might be included in non-WWW-content as well (e.g. in programmes downloaded via FTP), and that other media might be used to transport malware (e.g. e-mail, CD-ROMs). Thus the TOE plays an important part in securing the LAN but by far must not be the only measure implemented to obtain comprehensive protection.*

64 The following threats are completely counteracted by the TOE, provided the environment described in Section 3.2 is present:

65 **T.Malware**:
A user downloads and opens/executes data (e.g. programmes) from WWW sites (either explicitly or embedded in active content) onto an LC which impairs integrity, availability or confidentiality of data on net devices within the LAN.

66 *Application note: A typical example would be a virus, which would transmit sensitive information (e.g. passwords) from the LC to sites outside the LAN.*

67 **T.Eavesdrop**:
Malware uses available hardware (e.g. web cameras, microphones) to eavesdrop the physical workplace of the user, i.e. the physical environment of the LC.

68 The following threats are a consequence of the design of the TOE. The TOE and TOE host provide additional security functionality to counteract or minimise these threats.

69 **T.Credentials**:
An attacker deploys the authentication credentials obtained ("sniffed") on the TOE server to log onto a net device in the LAN directly.

70 *Application note: An attacker could use credentials obtained on the TOE server, e.g. by exploiting a newly found vulnerability, to use a remote login facility for the LAN to connect to a net devicein the LAN and hence completely bypass the TOE.*

71 **T.Hostcontrol:**
Malware running on the TOE host manipulates TSF data of either the TOE or the TOE host.

72 **T.Hostcrossing**:
Malware running on the TOE host in the session of user A obtains or manipulates data belonging to a session of user B (where user B is an arbitrary user of the TOE different from A), denies B access to her data or runs malware (possibly including a copy of itself) within the session of B.

73 *Application note: The data of user B contains all information used for browsing the WWW, e.g. settings and bookmarks, but also the information transmitted from and to the internet. The former is contained entirely in T.Hostcrossing, while the information exchanged with the WWW could also be manipulated by man-in-the-middle attacks for users not running the malware. These kinds of attacks are possible from outside the LAN as well (and also without the TOE present), e.g. on machines from the Internet provider, but are much harder to implement there since no untrusted code is run on these machines at the provider (i.e. outside the DMZ).*

74 **T.Spread**:
Malware running on the TOE host connects to an arbitrary net device in the LAN and transports another malware (or a copy of itself) on this net device, deploys this connection to obtain (sensitive) information from a net device in the LAN, manipulates information stored or processed on this net device or reduces the availability of net devices.

75 **T.Clientspread**:
Malware running on the TOE host uses the TOE protocol to deploy this connection to obtain (sensitive) information from this LC, to manipulate information stored or

processed on this LC, to reduce the availability of this LC or to transport some malware (e.g. a copy of itself) on the LC where the TOE client runs on.

76 *Application note: The idea is: The smaller the scope of the client-server protocol's functionality is the better it can be evaluated and the smaller is its probability for being exploited in an unintentional way. Hence this threat is* ***not*** *countered by popular client-server protocols which offer a "rich" suite of functionality, e.g. a powerful communication channel which can transmit arbitrary information in both directions. Thus a typical off-the-shelf terminal server solution does not provide protection against this threat.*

3.4 Organisational Security Policies

77 Since the entire motivation for IT security functionality is based on countermeasures against explicitly listed threats no OSPs are defined.

4 Security Objectives

78 The security objectives are an implementation independent description how the TOE counteracts the aforementioned threats. Also for each assumption regarding TOE usage the associated security objectives are described.

4.1 Security Objectives for the TOE

79 **O.ServerToClient**:
The TOE server only transmits audio-visual data (i.e. graphical and audible representation of web pages) and those information necessary to present this data (like window size, placement requirements). The TOE server may additionally be able to transmit plain text marked on the TOE host by the user to the TOE client; if present this functionality has default to off and may only be activable by an administrator. The TOE client can only receive and present this kind of information; if text reception is activated it can only be sent into a clipboard on the LC.

80 *Application note: Transmission of audio data is not a strict requirement for an implementation. It is up to the manufacturer to either include, optionally include (e.g. configurable by an administrator) or omit the possibility to transmit audio data. Considering the purpose of the TOE, however, to enable as unlimited access to WWW content as possible, at least an optional inclusion of audio data should be considered.*

81 *Application note: Providing the ability to transmit plain text to the LC (Copy and Paste) greatly enhances the usability of the TOE. On the other hand it also increases the risk for transfer of unwanted data (including malware) and hence some implementations might not implement this transfer. If this functionality is present it must default to off and should only be activated after a risk analysis of the organisation deploying the TOE has been performed.*

82 **O.ClientToServer**:
The TOE client can only transmit

- key presses (i.e. keyboard input) and mouse events explicitly directed towards the TOE client by the user, and
- the contents of a clearly designated configuration file at start-up of the TOE client

to the TOE server. The TOE server may additionally offer to transmit the plain text contents of a clipboard on the LC after explicit individual confirmation by the user for each transmission to the TOE server; if present this functionality has default to off and may only be activable by an administrator. The TOE client is unable to access any other data on the LC and transmit it to the TOE server.

83 *Application note: If authors of STs and developers opt, as strongly recommended, for an authentication scheme which does not rely on explicitly entered authentication attributes (credentials like passwords, pins) but rather uses some kind of pre-shared secret (distributed in advance by an administrator) (cf. paragraph 55) then typically the login procedure on the LC would decrypt the authentication attribute (credential stored in the configuration file of the TOE client). Upon start-up the TOE client would*

read this attribute from this file and send it - as if it were entered by the user - to the TOE host where it is verified.

4.2 Security Objectives for the Environment

84 **OE.Firewall**:
The TOE client runs on LCs inside the LAN. The TOE host is located in the DMZ and the TOE server runs thereon. Both the connection between the TOE host (situated in the DMZ) and any net device in the LAN as well as the connection between the TOE host and the Internet are separated by a firewall (cf. Figure 1). This firewall operates both on incoming as well as on outgoing traffic and includes network proxies, which operate on the transport (e.g. Internet Protocol) and additionally on the application layer for HTTP(S). The following rules are enforced by the firewall:

1. Connections from net devices in the LAN (including the LCs) to the TOE host can only use a port dedicated for the operation of the TOE server.
2. Only TOE clients running on net devices in the LAN can open connections to the TOE host.
3. It is impossible for the TOE host to initiate connections to net devices inside the LAN.
4. It is impossible for net devices inside the LAN to connect to content on the WWW directly, i.e. bypassing the TOE.
5. Only the TOE protocol as defined in paragraph 79 and 82 can be used in connections between the TOE host and a TOE client in the LAN.

85 This prevents the bypass of the TOE and its functionality, e.g. malware running on the TOE host cannot connect to arbitrary net devices in the LAN or evade using the TOE protocol between the TOE host and TOE client.

86 **OE.LC**
The TOE clients - running on LCs inside the LAN - will not be manipulated by users or software on the LC thus operates as specified in this PP.

87 **OE.Admin**:
The administrator is trustworthy, proficient and does not use this role to access WWW content, since the TOE and its environment cannot defend themselves against misconfiguration or usage in administrator mode.

88 **OE.Credentials**:
The TOE host operates an independent I&A system and offers the possibility to define rules for this system (e.g. specification of properties of the authentication attributes (credentials) used). Users do not use the same authentication attributes (credentials) for login on the TOE host as for any other net device in the LAN. This objective helps preventing transmission of passwords and login names used in the LAN into the potentially dangerous DMZ and attackers cannot use credentials obtained in the DMZ to connect directly to any net device within the LAN.

89 **OE.Selfprotection**:
The IT environment (here: on the TOE host) prevents malware running on the TOE

host during ordinary operation from manipulating TSF data of the TOE or the TOE host.

90 *Application note: This and the following objectives for the environment might seem partially redundant. This is on purpose, as past experience has shown that systems running untrusted code experience new attack vectors (e.g. new types of vulnerabilities are found) and hence the "security" of such a system should not rely on a single objective.*

91 **OE.Manipulation**:
The IT environment (here: on the TOE host) ensures that no programme running on the TOE host within a session of an user A (including, but not limited to, the browser, plugins/extensions and any active content) is able to access or manipulate data of an user B different from A, prevent B from accessing her data or runs malware within the session of B.

92 **OE.Minimal**:
The IT environment (here: on the TOE host) ensures that only programmes required for the operation of the TOE are available on the TOE host (e.g. TOE server, web browser, web browser extensions).

93 **OE.Session**:
The IT environment (here: on the TOE host) ensures that:

1. At the beginning of the time of each session (i.e. after login on the TOE server), all programmes accessible to the user on the TOE host are in a known state, i.e. executed on their own without any further user input[13], they only run code and access data as set up by an administrator. Especially it must be ensured that no application accesses any (active) content unknowingly by the user during initialisation (e.g. the address of the start page of the browser(s) cannot be altered by any program on the TOE host).
2. At the end of the time of each session (i.e. when logging off the TOE server), all programmes running within this session (i.e. all programmes part of this session) are terminated, including programmes intended for later execution (if any). It must be ensured that no programme is able to delay or continue execution beyond the end of the time of the session.

This objective ensures that malware is at most only active during the lifetime of a session and requires explicit user input to be loaded on the TOE (i.e. cannot be active at the start of the time of the session since the initial address for the WWW browser is fixed as well).

94 *Application note: Please note that the term session (defined in Sec. 8) only relates to users, not administrators, or programmes on the TOE host running without relation to a specific user (e.g. server programmes like the TOE server itself). Hence programmes related to administrator activity or without specific relation to a user are not affected by OE.Session.*

[13] *Application note: This could occur by entering a command name without parameters or by (double) clicking on an icon to start a program.*

95 **OE.Reset**:
The IT environment offers the facility to set up global time intervals where the entire TOE host including the TOE server is reinitialised to a known state, thereby terminating all sessions running at this point of time. The reinitialisation occurs in the following distinctive steps:

1. An integrity check on the entire static data of the mass storage on the TOE host is performed from outside the operating system normally running on the TOE host. Mass storage refers to the medium which stores the running operating system and all data related to the operating system and the TOE. Static data in this objective refers to all programs and (configuration) data which should not be altered during ordinary operation. During the integrity check it is important that the state of the mass storage can not be altered.
   Application note: This can be achieved by booting the TOE host from a read only medium and performing an integrity check on the on the data of the TOE. Another possibility is to save a snapshot of the mass storage and perform the integrity check on this snapshot from an independent system.
2. The result of this integrity check is transferred to a device/account outside the TOE host[14] available to an administrator of the TOE.
3. The static data on the TOE host is reset to a known good state. It is important that only known dynamic files (for example user accounts) differ from the known good state (reference state). If the reset uses any file from the previously running TOE host, it has to be assured that these files are either identically to the files of the reference state (static data) or a detailed content analysis has to be performed so that no malware can be hidden within these files. The previously running operating system of the TOE host must not have access to the mass storage with the known good state.
   Application note: This can be for example achieved by booting the TOE host from a read only media (done in step 2), erase the mass storage of the TOE host and restore a backup image from a read only media to the mass storage. After this, the dynamic data can be created from a database. Another example is to create a new mass storage for the TOE host and reset this to the known state from a different host (for example a LUN (logic unit number) on SAN (storage attached network) devices).
   Application note: It is strongly recommended that after step 2 an additional step is included, where the dynamic data is verified for file format integrity, e.g. a bookmark file is parsed for markup errors.
4. The TOE host is booted from the mass storage that was reset to the known good state in step 3.

The manufacturer of the TOE host has to provide guidance how to handle the results of the integrity check, e.g. actions to take and if and how the manufacturer of the TOE host should be informed about failed integrity checks.

These regular reinitialisations enable an administrator to ensure that the entire TOE host and TOE server is in a known state (especially that no malware is running) at

[14] This could be a printer, a mobile phone, an e-mail account, but of course not a net device inside the LAN.

regular intervals (after which explicit user action is required to bring malware again on the TOE, cf. OE.Session).

96 *Application note: A typical reset cycle might be implemented by booting a service system from a read-only medium or by shutting down the TOE host and performing the integrity check and reset from a different machine which has access to the hard drive of the TOE host (e.g. shared access of a disk like in load balancing). A typical reset then involves copying all programmes and configuration ("static data") from a (read only) medium provided by the manufacturer and configuration/updates provided by an administrator. Resuming normal operations would simply be a reboot into the TOE host and subsequently the TOE. It is important, however, that the reset cannot be influenced or prevented from within the TOE host*

97 *Application note: Please note that neither this nor any other objective include logging. The TOE is designed to minimise the risk of running malware. Since past experience shows that malware tends to find new (classes of) vulnerabilities it is highly desirable for manufacturers to learn about these to implement appropriate functionality to counter these (new) threats (this applies especially to the security functionality of the TOE host). Hence it is highly advisable for manufacturers to implement logging and it is strongly urged to provide guidance for administrators how to analyse the logged data and report possible break ins. Due to different requirements in different jurisdictions, however, especially regarding privacy, logging cannot be mandated in detail in this PP. Regarding this issue, please refer to Section 1.3 as well.*

4.3 Security Objectives Rationale

98 Please cf. to Section 7.1.

5 Extended Components Definition

99 The PP does not contain any extended component.

5.1 Extended Components Rationale

100 As this PP does not contain any extended component no rationale is necessary.

6 Security Requirements

101 As stated in the CC, operations on the SFRs may be performed. Refinement and completion of operations are denoted by SMALL CAPS. If a refinement caused text to be replaced, the original version is printed ~~stroken out~~. For the operations the uncompleted (original) version is given as a footnote. If the operation is to be completed by an author of an ST, the operation is enclosed in square brackets. Iterated components are denoted by a suffix to the component name, e.g. FMT_MSA.3**(h)**.

6.1 Security Functional Requirements for the TOE

6.1.1 Flow Control Policy "TOE transmission protocol"

Subjects: TOE server and TOE client

Objects: Any information exchanged between TOE server and TOE client

Security attributes: CopyPasteIn – Boolean value, default false
CopyPasteOut – Boolean value, default false

Management functions: SetCopyPasteIn – Allows to set security attribute "CopyPasteIn" on the TOE client
SetCopyPasteOut - Allows to set security attribute "CopyPasteOut" on the TOE client

102 *Application note: If a TOE - depending on the design decisions of the TOE manufacturer - does not provide a means to configure Copy and Paste (either or both transmission directions) it must default Copy and Paste (in either or both directions) to off.*

Flow Control Policy:

1. The TOE server shall only send audio-visual data (graphics, sounds and data required for display (e.g. window size, placement information)) to the TOE client.
2. If CopyPasteIn is "true" then the TOE server shall also send plain text marked on the TOE host by the user to the TOE client which stores it in the clipboard on the LC.
3. The TOE client shall only send input events (i.e. key presses, mouse events) which have been explicitly directed towards the TOE client and (only during client startup) the contents of a clearly identified file to the TOE server.
4. If CopyPasteOut is "true" then the TOE client shall also send the textual contents of the clipboard on the LC to the TOE server after the user explicitly allowed this single transmission.

6.1.2 FDP_IFC.1 Subset information flow control

Hierarchical to: No other components.

Dependencies: FDP_IFF.1 Simple security attributes

FDP_IFC.1.1 The TSF shall enforce the TOE TRANSMISSION PROTOCOL[15] on ALL INFORMATION EXCHANGED BETWEEN TOE CLIENT AND TOE SERVER[16].

6.1.3 FDP_IFF.1 Simple security attributes

Hierarchical to: No other components.

Dependencies: FDP_IFC.1 Subset information flow control
FMT_MSA.3 Static attribute initialisation

FDP_IFF.1.1 The TSF shall enforce THE TOE TRANSMISSION PROTOCOL[17] based on the following types of subject and information security attributes: ALL INFORMATION EXCHANGED BETWEEN TOE CLIENT AND TOE SERVER AND THE SECURITY ATTRIBUTES COPYPASTEIN AND COPYPASTEOUT[18].

FDP_IFF.1.2 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: THE INFORMATION EXCHANGE BETWEEN THE TOE CLIENT AND TOE SERVER OCCURS ACCORDING TO THE TOE TRANSMISSION PROTOCOL RULE 1 AND 3 INDEPENDENT OF ANY SECURITY ATTRIBUTE[19].

FDP_IFF.1.3 The TSF shall enforce ~~the~~ NO ADDITIONAL INFORMATION FLOW CONTROL SFP RULES[20].

FDP_IFF.1.4 The TSF shall explicitly authorise an information flow based on the following rules: [SELECTION: NO ADDITIONAL RULES, THE INFORMATION EXCHANGE BETWEEN TOE CLIENT AND TOE SERVER CONFORMS TO THE TOE TRANSMISSION PROTOCOL RULE 2 (DEPENDING ON THE SETTING OF COPYPASTEIN), THE INFORMATION EXCHANGE BETWEEN TOE CLIENT AND TOE SERVER CONFORMS TO THE TOE TRANSMISSION PROTOCOL RULE 4 (DEPENDING ON THE SETTING OF COPYPASTEOUT AND THE DECISION OF THE USER)][21].

FDP_IFF.1.5 The TSF shall explicitly deny an information flow based on the following rules: ANY INFORMATION EXCHANGE BETWEEN TOE CLIENT AND TOE SERVER NOT CONTAINED IN THE TOE TRANSMISSION PROTOCOL, INDEPENDENT OF ANY SECURITY ATTRIBUTE, TRANSMISSION OF PLAIN TEXT FROM THE TOE SERVER TO THE TOE CLIENT IF COPYPASTEIN IS FALSE AND

---

[15] [assignment: information flow control SFP]

[16] [assignment: list of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP]

[17] [assignment: information flow control SFP]

[18] [assignment: list of subjects and information controlled under the indicated SFP, and for each, the security attributes]

[19] [assignment: for each operation, the security attribute-based relationship that must hold between subject and information security attributes]

[20] [assignment: additional information flow control SFP rules]

[21] [assignment: rules, based on security attributes, that explicitly authorise information flows]

TRANSMISSION OF PLAIN TEXT FROM THE TOE CLIENT TO THE TOE SERVER IF EITHER COPYPASTEOUT IS FALSE OR THE USER REJECTED THE TRANSMISSION[22].

6.1.4 FMT_MSA.1 Management of security attributes

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MSA.1.1 The TSF shall enforce the TOE TRANSMISSION PROTOCOL[23] to restrict the ability to CHANGE FROM RESTRICTIVE TO PERMISSIVE[24] the security attributes COPYPASTEIN (IF SETCOPYPASTEIN IS PART OF THE TOE, CF. FMT_SMF.1), COPYPASTEOUT (IF SETCOPYPASTEOUT IS PART OF THE TOE, CF. FMT_SMF.1)[25] to ADMINISTRATORS[26].

103 *Application note: Essentially the ST author has 4 options: either no Copy and Paste (then this SFR is a no-op), only Copy from TOE server to TOE client (then this SFR contains only CopyPasteIn), only controlled Paste from TOE client to TOE server (then this SFR contains only CopyPasteOut) or both Copy and controlled Paste (then this SFR contains both CopyPasteIn and CopyPasteOut). Note that the choice among these four options is made in FMT_SMF.1 below, i.e. ST authors do not have any uncompleted operation in FMT_MSA.1. Depending on FMT_SMF.1 the selection in FDP_IFF.1.4 should be made as well. Please further note that FMT_MSA.3 contains all possible security attributes for the TOE transmission protocol, even if they are not manageable according to FMT_SMF.1 (in this case they remain restrictive, i.e. false).*

6.1.5 FMT_MSA.3(t) Static attribute initialisation

Hierarchical to: No other components.

Dependencies: FMT_MSA.1 Management of security attributes
FMT_SMR.1 Security roles

FMT_MSA.3.1(t) The TSF shall enforce the TOE TRANSMISSION PROTOCOL[27] to provide RESTRICTIVE[28] default values for security attributes that are used to enforce the SFP.

---

[22][assignment: rules, based on security attributes, that explicitly deny information flows]

[23][assignment: access control SFP(s), information flow control SFP(s)]

[24][selection: change_default, query, modify, delete, [assignment: other operations]]

[25][assignment: list of security attributes]

[26][assignment: the authorised identified roles]

[27][assignment: access control SFP, information flow control SFP]

[28][selection, choose one of: restrictive, permissive, [assignment: other property]]

FMT_MSA.3.2(t) The TSF shall allow the ADMINISTRATOR[29] to specify alternative initial values to override the default values when an object or information is created.

6.1.6 FMT_SMF.1 Specification of Management Functions

Hierarchical to: No other components.

Dependencies: No dependencies.

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions: [SELECTION: SETCOPYPASTEIN, SETCOPYPASTEOUT, NONE][30].

6.1.7 FMT_SMR.1 Security roles

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification

FMT_SMR.1.1 The TSF shall maintain the roles ADMINISTRATOR AND USER[31].

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

6.2 Security Functional Requirements Rationale

104 Please cf. to Section 7.2.

6.3 Security Assurance Requirements for the TOE

105 The assurance requirements for the evaluation of the TOE, its development and operating environment are to be chosen as the predefined assurance package EAL3 augmented by the following components:

- ALC_CMS.4 (Problem tracking CM coverage), and
- ALC_FLR.3 (Systematic flaw remediation).

106 The resulting assurance package is represented below (the components augmented are printed in bold):

Assurance component, cf. CC part 3 [3]	Short description
ADV_ARC.1	Security architecture description
ADV_FSP.3	Functional specification with complete summary
ADV_TDS.2	Architectural design

[29][assignment: the authorised identified roles]

[30][assignment: list of management functions to be provided by the TSF]

[31][assignment: the authorised identified roles]

AGD_OPE.1	Operational user guidance
AGD_PRE.1	Preparative procedures
ALC_CMC.3	Authorisation controls
ALC_CMS.4	**Problem tracking CM coverage**
ALC_DEL.1	Delivery procedures
ALC_DVS.1	Identification of security measures
ALC_FLR.3	**Systematic flaw remediation**
ALC_LCD.1	Developer defined life-cycle model
ASE_CCL.1	Conformance claims
ASE_ECD.1	Extended components definition
ASE_INT.1	ST introduction
ASE_OBJ.2	Security objectives
ASE_REQ.2	Derived security requirements
ASE_SPD.1	Security problem definition
ASE_TSS.1	TOE summary specification
ATE_COV.2	Analysis of coverage
ATE_DPT.1	Testing: basic design
ATE_FUN.1	Functional testing
ATE_IND.2	Independent testing - sample
AVA_VAN.2	Vulnerability analysis

Table 1: Evaluation assurance package

107 EAL3 was chosen as it provides a good balance between assurance and economical feasibility of evaluations of TOEs in order to provide incentives for manufactures to evaluate their ReCoBS system against this PP.

108 EAL3 is augmented by ALC_CMS.4 as the TOE host is designed to run unknown, untrusted and malicious software (malware) and hence confidence must be established that the developer properly tracks and considers security flaws and their resolution during the development of the TOE in order to avoid known classes of vulnerabilities to affect the TOE even if the exact exploit might yet be unknown during the evaluation.

109 Further EAL3 is augmented by ALC_FLR.3 as during operations new vulnerabilities will be discovered by the organisations deploying the TOE. In order to deploy corrective actions, the developer has to provide the user with guidance how to report the security vulnerability and has to be able to accept, track and properly act on those reports. ALC_FLR.3 ensures that these guidance and procedures are available for the TOE.

6.4 Application Note: Security Requirements for the IT Environment

110 To aid developers of TOEs conformant to this PP possible SFRs for the environment are listed in this Section. If such developers choose to integrate (parts of) the IT

environment into their TOE they probably will have to include (some of the) following SFRs in their security target.

6.4.1 TOE Host Access Policy "AC_HOST"

Subject:

Ordinary User:	*Representation of a person allowed to use the TOE host*
System User:	*Representation of a service running on the TOE host*

Objects:

Data:	*Information stored or processed on the TOE host*
System configuration data:	*Information stored or processed on the TOE host required or designated for the (proper) operation of the TOE host or TOE server as a whole*[32]
Programme:	*Special type of data which can be executed on the TOE host*

Access Policy:

1. *The TOE host has to limit access of ordinary users or programmes acting on their behalf to the data necessary to perform this task.*
2. *The TOE host has to limit access of ordinary users or programmes acting on their behalf to the time of their session.*
3. *The TOE host has to limit reading of system configuration data by ordinary users to the necessary minimum to operate the TOE host and has to limit write access to system configuration data to system users.*
4. *The TOE host has to ensure that ordinary users or programmes acting on their behalf can only access data owned by other users if said other users explicitly allowed this access.*
5. *The TOE host has to ensure that ordinary users or programmes acting on their behalf cannot block other ordinary users from accessing data which said other users are allowed to access.*
6. *The TOE host has to ensure that configuration for startup behaviour (default values) cannot be altered by ordinary users.*

111 *Application note: This access policy is only the minimum required to reach all objectives for the TOE and its environment. To complete all operations of all SFRs, the author of an ST, if applicable, will need to amend this policy, e.g. to include the administrative user(s) for the TOE host.*

6.4.2 The Integrity Self Test

The integrity self test consists of the following distinctive steps:

[32] *This should be understood in contrast to user individual configuration, e.g. display resolution, which might be (partially) stored on the LC and transmitted to the TOE server during login.*

1. *An integrity check on the entire static data of the mass storage on the TOE host is performed from outside the operating system normally running on the TOE host. Mass storage refers to the medium which stores the running operating system and all data related to the operating system and the TOE. Static data in this objective refers to all programs and (configuration) data which should not be altered during ordinary operation. During the integrity check it is important that the state of the mass storage can not be altered.*
2. *The result of this integrity check is transferred to an device/account outside the TOE host[33] available to an administrator of the TOE.*
3. *The static data on the TOE host is reset to a known good state. It is important that only known dynamic files (for example user accounts) differ from the known good state (reference state). If the reset uses any file from the previously running TOE host, it has to be assured that these files are either identically to the files of the reference state (static data) or a detailed content analysis has to be performed so that no malware can be hidden within these files. The previously running operating system of the TOE host must not have access to the mass storage with the known good state.*
4. *The TOE host is booted from the mass storage that was reset to the known good state in step 3.*

It is accompanied by guidance on how to handle the results and if and how the manufacturer should be informed of possible failed integrity checks.

6.4.3 FDP_ACC.2 Complete access control

Hierarchical to: *FDP_ACC.1 Subset access control*

Dependencies: *FDP_ACF.1 Security attribute based access control*

FDP_ACC.2.1 *The TSF shall enforce the AC_HOST[34] on ALL SUBJECTS AND OBJECTS DEFINED IN AC_HOST[35] and all operations among subjects and objects covered by the SFP.*

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

6.4.4 FDP_ACF.1 Security attribute based access control

Hierarchical to: *No other components.*

Dependencies: *FDP_ACC.1 Subset access control*
FMT_MSA.3 Static attribute initialisation

---

[33] *This could be a printer, a mobile phone, an e-mail account but of course no device inside the LAN.*

[34] *[assignment: access control SFP]*

[35] *[assignment: list of subjects and objects]*

FDP_ACF.1.1 *The TSF shall enforce the ACCESS POLICY AC_HOST[36] to objects based on the following: THE SUBJECTS, OBJECTS AND OPERATIONS AND THEIR SECURITY RELEVANT OPERATIONS AS SPECIFIED IN AC_HOST[37].*

FDP_ACF.1.2 *The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: ALL ACCESS RULES DEFINED IN AC_HOST[38].*

FDP_ACF.1.3 *The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: NONE[39].*

FDP_ACF.1.4 *The TSF shall explicitly deny access of subjects to objects based on the RULES SPECIFIED IN AC_HOST[40].*

6.4.5 FIA_SOS.1 Verification of secrets

Hierarchical to: *No other components.*

Dependencies: *No dependencies.*

FIA_SOS.1.1 *The TSF shall provide a mechanism to verify that secrets meet [assignment: a defined quality metric].*

112 *Application note: The quality metric required for the TOE depends on the authentication mechanism used by the TOE and is hence implementation dependent. Since the implemented TOE might counteract against additional threats (cf. Section 1.3) or pre-shared secrets are used, the authors of STs might choose stricter requirements for this metric.*

6.4.6 FIA_UAU.2 User authentication before any action

Hierarchical to: *FIA_UAU.1 Timing of authentication*

Dependencies: *FIA_UID.1 Timing of identification*

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.*

6.4.7 FIA_UID.2 User identification before any action

Hierarchical to: *FIA_UID.1 Timing of identification*

Dependencies: *No dependencies*

---

[36] *[assignment: access control SFP]*

[37] *[assignment: list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes]*

[38] *[assignment: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects]*

[39] *[assignment: rules, based on security attributes, that explicitly authorise access of subjects to objects]*

[40] *[assignment: rules, based on security attributes, that explicitly deny access of subjects to objects]*

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

6.4.8 FMT_MSA.3(h) Static attribute initialisation

Hierarchical to: *No other components.*

Dependencies: *FMT_MSA.1 Management of security attributes*
FMT_SMR.1 Security roles

FMT_MSA.3.1(h) *The TSF shall enforce the ACCESS CONTROL AC_HOST[41] to provide RESTRICTIVE[42] default values for security attributes that are used to enforce the SFP.*

FMT_MSA.3.2(h) *The TSF shall allow the [assignment: the authorised ADMINISTRATIVE roles][43] to specify alternative initial values to override the default values when an object or information is created.*

113 *Application note: The authors of STs, if applicable, have to specify here which roles are responsible in the IT environment on the TOE host to manage the TSF data. Typically this will be an administrator or root user, but more elaborate schemes might have additional administrative roles.*

6.4.9 FMT_SMR.2 Restrictions on security roles

Hierarchical to: *FMT_SMR.1 Security roles*

Dependencies: *FIA_UID.1 Timing of identification*

FMT_SMR.2.1 *The TSF shall maintain the roles: ORDINARY USER, SYSTEM USER [ASSIGNMENT: ADMINISTRATIVE ROLE(S)][44].*

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

FMT_SMR.2.3 *The TSF shall ensure that the ~~conditions~~ CONDITION THAT ORDINARY USERS CANNOT SWITCH TO ANY OTHER ROLE, [ASSIGNEMENT: FURTHER RESTRICTIONS][45], IS[46] ~~are~~ satisfied.*

114 *Application note: The authors of STs, if applicable, have to specify here which administrative roles are available on the TOE host. These role(s) should be added to AC_HOST in Section 6.4.1. Typical roles are an administrator or root user, but more elaborate schemes might have additional administrative roles. In this case the open assignments in FMT_SMR.2 should be used to define those roles and to add the restriction(s) for each administrative role as well.*

---

[41] *[assignment: access control SFP, information flow control SFP]*

[42] *[selection, choose one of: restrictive, permissive,[assignment: other property]]*

[43] *[assignment: the authorised identified roles]*

[44] *[assignment: authorised identified roles]*

[45] *If each user has several roles depending on context this statement has to be adapted.*

[46] *[assignment: conditions for the different roles]*

6.4.10 FPT_TST.1 TSF testing

Hierarchical to: *No other components.*

Dependencies: *No dependencies*

FPT_TST.1.1 *The TSF shall run ~~a suite of~~ THE INTEGRITY self test~~s~~ PERIODICALLY DURING NORMAL OPERATION[47] to demonstrate the correct operation of THE TSF[48].*

FPT_TST.1.2 *The TSF shall provide authorised users with the capability to verify the integrity of TSF DATA[49].*

FPT_TST.1.3 *The TSF shall provide authorised users with the capability to verify the integrity of stored TSF executable code.*

[47] *[selection: during initial start-up, periodically during normal operation, at the request of the authorised user, at the conditions[assignment: conditions under which self test should occur]]*

[48] *[selection: [assignment: parts of TSF], the TSF]*

[49] *[selection: [assignment: parts of TSF], TSF data]*

7 Rationales

115 The Rationales demonstrate that this Protection Profile is a complete and consistent set of security requirements. Furthermore it is shown that a TOE conformant to this PP is able to counteract or minimise all threats listed above by deploying effective counter measures.

7.1 Security Objectives Rationale

The following table provides an overview for security objectives coverage.

	O.ServerToClient	O.ClientToServer	OE.Firewall	OE.LC	OE.Admin	OE.Credentials	OE.Minimal	OE.Selfprotection	OE.Manipulation	OE.Session	OE.Reset
A.Firewall			✳								
A.LC				✳							
A.Admin					✳						
A.Authentication						✳					
A.Minimal							✳				
T.Malware	✳		✳	✳							
T.Eavesdrop	✳	✳	✳	✳							
T.Credentials						✳					
T.Hostcontrol								✳		✳	✳
T.Hostcrossing									✳	✳	✳
T.Spread			✳	✳						✳	✳
T.Clientspread	✳	✳	✳	✳						✳	

Table 2: Security Objective Rationale – each threat and each assumption is addressed by at least one objective and each objective addresses at least one threat or assumption.

7.1.1 Protection offered by the TOE against the Threats

116 **T.Malware**: A user downloads and opens/executes data (e.g. programmes) from WWW sites (either explicitly or embedded in active content) onto an LC which impairs integrity, availability or confidentiality of data on net devices within the LAN.

O.ServerToClient addresses this threat directly by strictly limiting the types of data transmitted from the WWW onto the LC to audio-visual data and placement information thus preventing arbitrary data including executable code to reach the LC via this channel (cf. also Footnote 50 for an additional discussion). Due to **OE.Firewall**, direct transmission of arbitrary data from the WWW onto the LC (i.e. by using other channels and bypassing the TOE) is not possible either.

If the TOE offers Copy and Paste, a malware could in principle copy itself in a textual representation in the clipboard on the LC. Since the LC is not manipulated (**OE.LC**) all programmes on the LC are in the pre-configured state, especially the pasting has to be initiated by the user. Thus the attacker has to use another channel (e.g. social engineering) to convince the user to paste the textual representation of the malware into an appropriate interpreter (where an administrator of the LC has full control which interpreters are present at all). This is typically more difficult than sending the user a malware (e.g. contained in an document) and convince the user to open this document (and thus execute the malware). As enabling this limited Copy and Paste is, however, an additional risk, manufactures might omit this functionality and if it is present, the channel defaults to "off" and must be explicitly activated by an administrator (possibly following an individual risk analysis).

117 **T.Eavesdrop**: Malware uses available hardware (e.g. web cameras, microphones) to eavesdrop the physical workplace of the user, i.e. the physical environment of the LC.

O.ClientToServer addresses this threat directly by ensuring that only keyboard input ("key presses") and mouse events directed explicitly towards the TOE client, the contents of a clearly designated configuration file at start up (see next paragraph for rationale) and optionally individual paste actions (see below) will be transmitted to the TOE server (and hence further to WWW sites) thus preventing the contents of arbitrary files from being transferred and the transmission of arbitrary data streams (e.g. audio or video streams) onto the TOE host. Since the TOE client is not altered (manipulated), as ensured by **OE.LC**, no malware is present on the TOE which could simulate an input for the TOE client, the transport of such malware onto the client is prevented also (cf. **O.ServerToClient** and **OE.Firewall**). Furthermore the environment ensures that a direct connection from the TOE client to WWW sites, bypassing the TOE entirely, is not possible (cf. OE.Firewall) neither.

To allow some configuration to be stored on the non-manipulated LC (c.f. OE.LC and definition of attacker), e.g. credentials for the TOE host, the TOE client may send the contents of a clearly designated configuration file from the TOE client to the TOE server/host at startup. Later on (i.e. during operation) the TOE client

cannot transmit any file contents to the TOE server thus again preventing the transmission of arbitrary file contents to the TOE server. Since the LC is not manipulated by users or software on the LC the file only contains the information as set up and not "arbitrary" information from the LC.

Finally the TOE may offer the possibility to transmit the contents of the clipboard of the LC to the TOE server. Since the LC is not manipulated, the transfer has to be initiated by the LC and each paste action has to be individually confirmed by the user, malware on the TOE host is unable to access arbitrary content in the clipboard (no automatically shared clipboard is present!) but can possibly only access the textual data explicitly pasted by the user to the TOE server after explicit individual confirmation. As using the Copy and Paste channel is, however, an additional risk, manufactures might omit this functionality and if it is present, the channel defaults to "off" and must be explicitly activated by an administrator (possibly following an individual risk analysis).

118 **T.Clientspread**: Malware running on the TOE host uses the TOE protocol to deploy this connection to obtain (sensitive) information from this LC, to manipulate information stored or processed on this LC, to reduce the availability of this LC or to transport some malware (e.g. a copy of itself) on the LC where the TOE client runs on.

O.ServerToClient addresses this threat directly by strictly limiting the types of data transmitted from the WWW onto the LC to audio-visual data, formatting information and – optionally – textual information via Copy and Paste (see next paragraph) thus preventing arbitrary data including executable code to reach the LC via this channel[50]. Due to the objective **OE.Firewall** direct transmission of arbitrary data from the WWW onto the LC (i.e. by using other channels and bypassing the TOE) is not possible either.

If the TOE offers Copy and Paste, a malware could in principle use this channel to communicate with the TOE. Since the LC is not manipulated (**OE.LC**) all programmes on the LC are in the pre-configured state, especially the pasting of any textual information has to be initiated by the user. Thus the attacker has to user another channel (e.g. social engineering) to convince the user to paste the textual information into an appropriate interpreter (where an administrator of the LC has full control which interpreters are present at all) or any other place on the LC. Thus the user has full control over manipulation attempts of the malware. Similarly in the other direction malware on the TOE host can only access those

[50] In principle any type of data can be encoded in the audio-visual data stream sent from the TOE server to the TOE client. Since the TOE client can only **display** the information sent and is not manipulated (OE.LC) no decoding of other information can occur on the LC, hence preventing arbitrary data (including commands) to be **received** by the LC via the TOE protocol.

information which has been explicitly confirmed by the user to be transferred from the LC to the TOE host, i.e. no shared clipboard allows "snooping" of information processed on the LC. As using Copy and Paste is, however, an additional risk, manufactures might omit this functionality (either or both transmission directions) and if it is present, the channels default to "off" and must be explicitly activated by an administrator (possibly following an individual risk analysis).

Since the TOE client is not manipulated (cf. OE.LC) it fulfils the client part of **O.ClientToServer** and will only transmit key presses and mouse events which have been directed explicitly towards the TOE client to the TOE server (and from there to sites in the WWW) and – optionally – individual contents of the clipboard (see previous paragraph for a discussion) and the TOE client will be unable to access arbitrary files/information on LC (the configuration file is the only file available for access to the TOE client, and this file can only be read before/while logging into the TOE host, when no malware is running within (part of) the session of the user logging in (cf. **OE.Session**)), i.e. only data (including clipboard contents) consciously sent by the user is received on the TOE host and malware is unable to secretly "pull" any other information (like, e.g., from a shared clipboard as used on some terminal systems not compliant to this PP). Finally the TOE protocol has a fixed maximum throughput from the TOE server to the TOE client (roughly the number of audio-visual information transmitted per time interval) which imposes a pre-defined maximum load onto the LC which has to display this information. Thus also the availability of the LC cannot be reduced by malware running on the TOE host.

7.1.2 Protection offered by the TOE environment against the Threats

119 **T.Malware**: A user downloads and opens/executes data (e.g. programmes) from WWW sites (either explicitly or embedded in active content) onto an LC which impairs integrity, availability or confidentiality of data on net devices within the LAN.

OE.Firewall supports the countermeasures against this threat, as it prevents direct transmission of arbitrary data from the WWW onto the LC (i.e. by using other channels and bypassing the TOE). Thus all direct data transfer from WWW sites has to pass through the TOE (and especially through the TOE protocol which is decoded / encoded on the LC as defined in this PP (cf. **OE.LC**) and thus disallows a direct connection/channel to WWW sites) and therefor effectiveness of **O.ServerToClient** is ensured.

120 **T.Eavesdrop**: Malware uses available hardware (e.g. web cameras, microphones) to eavesdrop the physical workplace of the user, i.e. the physical environment of the LC.

OE.LC ensures that the TOE client is not altered (manipulated) and no malware is present on the LC which could simulate an input for the TOE client. Further the transport of such malware onto the client is prevented also (cf. **O.ServerToClient** which only allows pure audio-visual data and – if optionally pure text data – to be transmitted to the client (cf. also Footnote 50) and **OE.Firewall** which prevents further communication channels which could bypass the TOE). Finally this objective takes care that a direct connection from the LC to WWW sites, bypassing the TOE entirely, is not possible neither. These combined measures support **O.ClientToServer** and prevent the LC and its physical environment from being eavesdropped.

121 **T.Credentials**: An attacker deploys the authentication credentials obtained ("sniffed") on the TOE server to log onto a net device in the LAN directly.

OE.Credentials addresses this threat directly since it ensures that users have a credential for the TOE which is different from any credential used in the LAN. It also ensures that the TOE host provides an independent I&A system. Therefore there is no information on the TOE host which could be used by an attacker to log onto any net device within the LAN.

122 **T.Hostcontrol:** Malware running on the TOE host manipulates TSF data of either the TOE or the TOE host.

OE.Selfprotection addresses this threat directly since it ensures that malware running on the TOE host cannot manipulate TSF data of neither the TOE nor the TOE host. This objective is supported by **OE.Reset** which ensures that the TOE returns to a known state without any malware running and with well-known TSF data in regular time intervals and **OE.Session** which ensures that all programmes (including malware) running within (/are part of) a session are terminated[51] when the user logs off the TOE server. This objective reduces the impact of possible new vulnerabilities found after the certification of the TOE which might enable malware to (partially) bypass OE.Selfprotection as malware and (possible) modification of TSF data caused by it has a maximum lifetime on the system (and requires user action to return to the system afterwards). Finally during OE.Reset those modifications will be detected.

123 **T.Hostcrossing**: Malware running on the TOE host in the session of user A obtains or manipulates data belonging to a session of user B (where user B is an arbitrary user of the TOE different from A), denies B access to her

[51] This only relates to programmes started by users, not to those associated to administrators or general tasks on the TOE host, cf. paragraph 94 and definition of "session" within this PP (in Chapter 8).

data or runs malware (possibly including a copy of itself) within the session of B.

OE.Manipulation addresses this threat directly since it ensures that no programme (including malware) running on the TOE host within a session of user A is able to access or manipulate data of an user B different from A, to prevent B from accessing her data or executes malware within the session of user B.

This objective is supported by **OE.Session** which ensures that no malware is able to run at the start of the time of a session (e.g. when starting the browser) and all programmes run as part of a session (i.e. no programme with the rights of any user A runs after that user has logged off and before she has logged on again). Thus OE.Session reduces the impact of possible new vulnerabilities found after the certification of the TOE which might enable malware to (partially) bypass OE.Manipulation. Further malware has a maximum lifetime (ensured by **OE.Reset**) on the system (and requires user action to return to the system afterwards).

124 **T.Spread**: Malware running on the TOE host connects to an arbitrary net device in the LAN and transports another malware (or a copy of itself) on this net device, deploys this connection to obtain (sensitive) information from a net device in the LAN, manipulates information stored or processed on this net device or reduces the availability of net devices.

OE.Firewall addresses this threat directly since it ensures that no programme (including malware) running on the TOE host is able to open a connection to any net device within the LAN. For countermeasures against attacks using existing connections, please see the countermeasures against T.Hostcrossing and T.Clientspread above. The reuse of existing connections to open new connections (from the TOE server to any net device) is not possible neither since the TOE client is unable to interpret such an request (it can only display audio visual data), cf. also Footnote 50 and will not be manipulated by users or software to do so (cf. **OE.LC**). This objective is supported by **OE.Session** which ensures that no malware is able to run when no user session is active (and idle users sessions, i.e. periods of time where the user is not interacting with the session (e.g. over night) are discouraged by the maximum lifetime of a session (cf. **OE.Reset**)). This objective reduces the impact of possible new communication paths through the firewall found after the certification of the TOE which might enable malware (limited) access to (information about) net devices within the LAN[52].

[52] Such a communication path might for example even be as limited as timing information gained from sending packages during periods of low network activity (“at night”) to the firewall for obtaining information about the LAN.

7.1.3 Consideration of the assumptions

125 **A.Firewall:** The TOE client runs on LCs inside the LAN. The TOE host is located in the DMZ and the TOE server runs thereon. Both the connection between the TOE host (situated in the DMZ) and any net device in the LAN as well as the connection between the TOE host and the Internet are separated by a firewall (cf. Figure 1). This firewall operates both on incoming as well as on outgoing traffic and includes network proxies, which operate on the transport (e.g. Internet Protocol) and additionally on the application layer for HTTP(S). The following rules are enforced by the firewall:

1. Connections from net devices in the LAN (including the LCs) to the TOE host can only use a port dedicated for the operation of the TOE server.
2. Only TOE clients running on net devices in the LAN can open connections to the TOE host.
3. It is impossible for the TOE host to initiate connections to net devices inside the LAN.
4. It is impossible for net devices inside the LAN to connect to content on the WWW directly, i.e. bypassing the TOE.
5. Only the TOE protocol as defined in paragraph 79 and 82 can be used in connections between the TOE host and a TOE client in the LAN.

OE.Firewall addresses this assumption directly as an requirement for the environment of the TOE.

126 **A.LC** The TOE clients - running on LCs inside the LAN - will not be manipulated by users or software on the LC.

OE.LC addresses this assumption directly as an requirement for the environment of the TOE.

127 **A.Admin**: The administrator is trustworthy, proficient and does not use this role to access WWW content.

OE.Admin addresses this assumption directly as an requirement for the administrator of the TOE. The competence of the administrator is important, since erroneous administration or usage with elevated (i.e. administrative) privileges has to be avoided.

128 **A.Authentication**: The users do not reuse any identification and authentication attribute (credential) on the TOE which has been used on any net device within the LAN.

OE.Credentials ensures the availability of an I&A system on the TOE host and addresses this assumption directly as an requirement for the users of the TOE.

129 **A.Minimal**: Only programmes required for the operation of the TOE are available on the TOE host (e.g. TOE server, web browser, web browser extensions).

OE.Minimal addresses this assumption directly as an requirement for the environment (here: the TOE host) of the TOE. Since the TOE server is intended to support simultaneous sessions of multiple users running untrustworthy software (including malware) it offers an exposed target for monitoring or manipulating data transmitted to and from the internet. To reduce the number of possible exploitable vulnerabilities in the IT environment only programmes required for the TOE host and TOE server for proper operation are assumed to be present and hence able to access or operate on the TOE server.

7.2 Security Requirements Rationale

7.2.1 Security Functional Requirements Rationale

130 The following table provides an overview for security functional requirements coverage. *For convenience for developers (cf. paragraph 110) hints for a rational for SFRs for the IT environment (for objectives for the IT environment where SFRs are sensible) are given as well.*

	O.ServerToClient	O.ClientToServer	OE.Credentials	OE.Selfprotection	OE.Manipulation	OE.Session	OE.Reset
FDP_IFC.1	⌘	⌘					
FDP_IFF.1	⌘	⌘					
FMT_MSA.1	⌘	⌘					
FMT_MSA.3(t)	⌘	⌘					
FMT_SMF.1	⌘	⌘					
FMT_SMR.1	⌘	⌘					
FDP_ACC.2					⌘	⌘	
FDP_ACF.1					⌘	⌘	
FIA_SOS.1			⌘				
FIA_UAU.2			⌘		⌘		
FIA_UID.2			⌘		⌘		
FMT_MSA.3(h)				⌘	⌘		

	O.ServerToClient	O.ClientToServer	OE.Credentials	OE.Selfprotection	OE.Manipulation	OE.Session	OE.Reset
FMT_SMR.2			✂		✂		
FPT_TST.1							✂

Table 3: Coverage of Security Objective for the TOE and TOE environment by SFR

131 **O.ServerToClient**: The TOE server only transmits audio-visual data (i.e. graphical and audible representation of web pages) and those information necessary to present this data (like window size, placement requirements). The TOE server may additionally be able to transmit plain text marked on the TOE host by the user to the TOE client; if present this functionality has default to off and may only be activable by an administrator. The TOE client can only receive and present this kind of information; if text reception is activated it can only be sent into a clipboard on the LC.

FDP_IFC.1 and **FDP_IFF.1** ensure that all communication between TOE server and TOE client obeys the TOE transmission protocol, which explicitly states that the TOE server may only transmit audio-visual data and formatting information required for display to the TOE client. If the ST author selected to include the possibility for Copying (i.e. to include SetCopyPasteIn in his ST) in FDP_IFF.1.4 and FMT_SMF.1 then the textual contents of paste actions into a clipboard on the LC are also transmitted from the TOE server to the TOE client. This additional transmission – if present – can only be activated by an administrator (**FMT_MSA.1** and **FMT_SMR.1**) and is restrictively configured (**FMT_MSA.3(t)**), i.e. textual transmission defaults to "off", allowing an administrator to perform a risk analysis before setting a new default value.

132 **O.ClientToServer**: The TOE client can only transmit

- key presses (i.e. keyboard input) and mouse events explicitly directed towards the TOE client by the user, and
- the contents of a clearly designated configuration file at start-up of the TOE client

to the TOE server. The TOE server may additionally offer to transmit the plain text contents of a clipboard on the LC after explicit individual confirmation by the user for each transmission to the TOE server; if present this functionality has default to off and may only be activable by an administrator. The TOE client is

unable to access any other data on the LC and transmit it to the TOE server.

FDP_IFC.1 and **FDP_IFF.1** ensure that all communication between TOE server and TOE client obeys the TOE transmission protocol, which explicitly states that the TOE client may only transmit key presses and mouse events which have been explicitly directed towards the TOE client, and (during start-up of the TOE client) the contents of a clearly marked file to the TOE server. If the ST author selected to include the possibility for controlled Pasting (i.e. to include SetCopyPasteOut in his ST) in FDP_IFF.1.4 and **FMT_SMF.1** then the textual contents of copy actions from the LC to the TOE server after individual confirmation by the user are also transmitted from the TOE client to the TOE server. This additional transmission – if present - can only be activated by an administrator (**FMT_MSA.1** and **FMT_SMR.1**) and is restrictively configured (**FMT_MSA.3(t)**), i.e. textual transmission defaults to "off", allowing an administrator to perform a risk analysis before setting a new default value.

133 ***OE.Credentials**: The TOE host operates an independent I&A system and offers the possibility to define rules for this system (e.g. specification of properties of the authentication attributes (credentials) used). Users do not use the same authentication attributes (credentials) for login on the TOE host as for any other net device in the* LAN.

***FIA_UID.2** requires the user to identify himself before any operation with the TOE, **FIA_UAU.2** requires the user to authenticate before any operation with the TOE. These two requirements implement an independent[53] I&A system.*

***FIA_SOS.1** ensures that the TOE validates that the authentication attributes (credentials) meet certain minimum criteria.*

***FMT_SMR.2** defines the roles to be present in the TOE and ensures that users are linked to these roles .*

134 ***OE.Selfprotection**: The IT environment (here: on the TOE host) prevents malware running on the TOE host during ordinary operation from manipulating TSF data of the TOE or the TOE host.*

***FMT_MSA.3(h)** enforces the access control policy AC_HOST to ensure that the default values for security attribute on the host are restrictive. It further defines administrative roles which are allowed to specify alternative initial values for objects or information created.*

[53] *Independent from any I&A in the LAN, as assured by **FDP_IFC.1** and **FDP_IFF.1** (i.e. the TOE transmission protocol).*

135 ***OE.Manipulation****: The IT environment (here: on the TOE host) ensures that no programme running on the TOE host within a session of an user A (including, but not limited to, the browser, plugins/extensions and any active content) is able to access or manipulate data of an user B different from A, prevent B from accessing her data or runs* malware within the session of B.

FIA_UID.2 *requires the user to identify himself before any operation with the TOE host,* ***FIA_UAU.2)*** *requires the user to authenticate before any operation with the TOE host. These two requirements implement the prerequisites for an access control system for the TOE host.*

FMT_SMR.2 *ensures that the TOE host distinguishes between ordinary users, system users and administrative user(s), assigns these users to roles and prevents ordinary users to switch to any other role (e.g. to the administrative user role).*

FDP_ACC.2 *ensures that the access control AC_HOST is enforced for all subjects and objects of the TOE host covered in the SFP and conversely that all subjects and objects within the TSC are covered by AC_HOST.*

FDP_ACF.1 *ensures that AC_HOST is enforced, including all rules granting and refusing access.*

FMT_MSA.3(h) *defines that the access control AC_HOST provides restrictive defaults and defines the administrative user(s) who can alter these defaults.*

136 ***OE.Session:*** *The IT environment (here: on the TOE host) ensures that:*

1. *At the beginning of the time of each session (i.e. after login on the TOE server), all programmes accessible to the user on the TOE host are in a known state, i.e. executed on their own without any further user input, they only run code and access data as set up by an administrator. Especially it must be ensured that no application accesses any (active) content unknowingly by the user during initialisation (e.g. the address of the start page of the browser(s) cannot be altered by any program on the TOE host).*
2. *At the end of the time of each session (i.e. when logging off the TOE server), all programmes running within this session (i.e. all programmes part of this session) are terminated, including programmes intended for later execution (if any). It must be ensured that no programme is able to delay or continue execution beyond the end of the time of the session.*

FDP_ACC.2 *ensures that the access control AC_HOST is enforced for all subjects and objects of the TOE host covered in the SFP and conversely that all subjects and objects within the TSC are covered by AC_HOST, including the requirement that all*

programmes started directly or indirectly by a user must be part of a session, i.e. cannot be started before the user logs on and must be terminated during log out.

***FDP_ACF.1** ensures that AC_HOST is enforced, including all rules granting and refusing access, especially during log out.*

137 ***OE.Reset:*** *The IT environment offers the facility to set up global time intervals where the entire TOE host including the TOE server is reinitialised to a known state, thereby terminating all sessions running at this point of time. The reinitialisation occurs in the following distinctive steps:*

1. *An integrity check on the entire static data of the mass storage on the TOE host is performed from outside the operating system normally running on the TOE host. Mass storage refers to the medium which stores the running operating system and all data related to the operating system and the TOE. Static data in this objective refers to all programs and (configuration) data which should not be altered during ordinary operation. During the integrity check it is important that the state of the mass storage can not be altered. Application note: This can be achieved by booting the TOE host from a read only medium and performing an integrity check on the on the data of the TOE. Another possibility is to save a snapshot of the mass storage and perform the integrity check on this snapshot from an independent system.*
2. *The result of this integrity check is transferred to a device/account outside the TOE host available to an administrator of the TOE.*
3. *The static data on the TOE host is reset to a known good state. It is important that only known dynamic files (for example user accounts) differ from the known good state (reference state). If the reset uses any file from the previously running TOE host, it has to be assured that these files are either identically to the files of the reference state (static data) or a detailed content analysis has to be performed so that no malware can be hidden within these files. The previously running operating system of the TOE host must not have access to the mass storage with the known good state. Application note: This can be for example achieved by booting the TOE host from a read only media (done in step 2), erase the mass storage of the TOE host and restore a backup image from a read only media to the mass storage. After this, the dynamic data can be created from a database. Another example is to create a new mass storage for the TOE host and reset this to the known state from a different host (for example a LUN (logic unit number) on SAN (storage attached network) devices).*

Application note: It is strongly recommended that after step 2 an additional step is included, where the dynamic data is verified for file format integrity, e.g. a bookmark file is parsed for markup errors.

4. *The TOE host is booted from the mass storage that was reset to the known good state in step 3.*
5. *The manufacturer of the TOE host has to provide guidance how to handle the results of the integrity check, e.g. actions to take and if and how the manufacturer of the TOE host should be informed about failed integrity checks.*

FPT_TST.1 *ensures that periodically during normal operation a self test is executed which involves terminating all sessions, running an integrity check on the static data from outside the operating system of the TOE host, providing an administrator with the results, resetting the static data of the TOE host to a known good state and resuming normal operation of the TOE, thus directly fulfilling the objective OE.Reset.*

7.2.2 Dependency Rationale

138 The following table provides an overview over all SFRs and their dependencies. *For convenience for developers (cf. paragraph 110) the suggested SFRs for the IT environment are included as well.*

Reference	SFR	Dependencies	Comment
6.1.2	FDP_IFC.1 Subset information flow control	6.1.3 FDP_IFF.1 Simple security attributes	
6.1.3	FDP_IFF.1 Simple security attributes	6.1.2 FDP_IFC.1 Subset information flow control 6.1.5 FMT_MSA.3(t) Static attribute initialisation	
6.1.4	FMT_MSA.1 Management of security attributes	[FDP_ACC.1 Subset access control or 6.1.2 FDP_IFC.1 Subset information flow control] 6.1.7 FMT_SMR.1 Security roles 6.1.6 FMT_SMF.1 Specification of Management	For the first dependency FDP_IFC.1 was chosen.

Reference	SFR	Dependencies	Comment
		Functions	
6.1.5	FMT_MSA.3(t) Static attribute initialisation	6.1.4 FMT_MSA.1 Management of security attributes 6.1.7 FMT_SMR.1 Security roles	
6.1.6	FMT_SMF.1 Specification of Management Functions	none	
6.1.7	FMT_SMR.1 Security roles	FIA_UID.1 Timing of identificaton	Dependency **not** satisfied, see rationale below.
6.4.3	*FDP_ACC.2 Complete access control*	6.4.4 *FDP_ACF.1 Security attribute based access control*	*FDP_ACC.2 is hierarchical to FDP_ACC.1*
6.4.4	*FDP_ACF.1 Security attribute based access control*	6.4.3 *FDP_ACC.2 Complete access control* 6.4.8 *FMT_MSA.3(h) Static attribute initialisation*	
6.4.5	*FIA_SOS.1 Verification of secrets*	none	
6.4.6	*FIA_UAU.2 User authentication before any action*	*FIA_UID.1 Timing of identification*	*Fulfilled by 6.4.7 FIA_UID.2 User identification before any action*
6.4.7	*FIA_UID.2 User identification before any action*	*none*	*FIA_UID.2 is hierarchical to FIA_UID.1*
6.4.8	*FMT_MSA.3(h) Static attribute initialisation*	*FMT_MSA.1 Management of security attributes* 6.4.9 *FMT_SMR.2 Restrictions on security roles*	*First dependency **not** satisfied, see rationale below.*
6.4.9	*FMT_SMR.2*	6.4.7 *FIA_UID.2*	*FMT_SMR.2 is*

Reference	SFR	Dependencies	Comment
	Restrictions on security roles	*User identification before any action*	*hierarchical to FMT_SMR.1*
6.4.10	*FPT_TST.1 TSF testing*	*none*	

Table 4 Dependencies between the SFR for the TOE

139 As shown in Table 4 all dependencies (except for 6.1.7 - FMT_SMR.1 Security roles *and 6.4.8 - FMT_MSA.3(h) Static attribute initialisation* see below) are fulfilled either directly or by functional requirements hierarchical to the dependency.

140 **FMT_SMR.1** depends on FIA_UID.1 which is not fulfilled in this PP. Since the entire identification and authentication (I&A) is provided by the TOE host (i.e. the IT environment) it is not possible for the TOE to enforce the timing of the identification but rather it has to assume that the IT environment only allows access after identification (and authentication).

141 ***FMT_MSA.3(h)*** *depends on FMT_SMR.2 (fulfilled) and on FMT_MSA.1 (not fulfilled).* ***FMT_MSA.1*** *restricts the management of (some of) the security attributes of the TOE host to certain authorised identified roles. For the purpose of the TOE, however, the detailed management of the TOE host (including the complete list of security attributes and operations with them) is not relevant. This is consistent with the TOE host access policy (Section 6.4.1) which also is limited to those subjects, objects and operations which are relevant to the aims of the TOE itself. This unfulfilled dependency does not exclude the possibility that the TOE host implements something similar to FMT_MSA.1 for proper operation.*

7.2.3 Security Assurance Requirements Rationale

142 The rationale is included in Section 6.2.

8 Glossary and Acronyms

Term	Definition
Active content	A programme which is integrated in a web page and delivered to the browser upon accessing that web page for executing (on the TOE host). Examples are ActiveX and JavaScript.
Authentication attribute	A means to demonstrate the presence of a certain person. A typical example possible for the TOE are passwords. Synonymous to credential in this PP.
Configuration data	Variable data which is required to ensure the intended operation of the TOE and its environment, e.g. access rights and passwords.
Credential	see authentication attribute
Demilitarised Zone	(Part of) a network which is separated both from the LAN as well as from the Internet by firewalls.
Firewall	A system of hard or software based components which ensures secure linkage of IP networks by limiting the technically possible communication to the those defined in a security policy. Sometimes the term “security gateway” is used instead.
HTTP ports	A finite list of port numbers used to access content of the WWW. Such a list typically includes port 80, 443 and possibly 8080 and similar numbers.
Local Area Network	Network which has been encapsulated from the Internet by firewalls. The LC are located within the LAN. The TOE client runs on the LC, while the TOE server runs on the TOE host, which is situated in the DMZ.
Local computer	A computer in a LAN with controlled access to the Internet. A local computer is used by one or several users for completion of their tasks.
Malware	A programme (which might be an active content) which performs actions without explicit consent by the user under which environment it is launched. This term includes both remote controlled as well as autonomous programmes.
Net device	All machines connected to a network which can either or both receive and transmit data, e.g. LC, routers, switches.
Protocol data	Data generated by the TOE or TOE host intended for audit, e.g. user name, access times and URLs of requested web pages.
ReCoBS server	This term denotes the combination of the TOE host and the TOE server. It is taken from the BSI concept [6] but not used within this PP.
Session	Set of programmes started on the TOE host directly or indirectly by a user between log on and log off of this user on the TOE host.
Time of a session	Time starting at log on and finishing at log off of a user at the TOE server.

Term	Definition
Target of Evaluation (TOE)	This term, taken from the CC, denotes the IT product, IT component or IT system, which has to be evaluated for fulfilling all security requirements. Within this PP the TOE consists of the TOE server and the TOE client.
TOE client	Program running on a LC to connect to the TOE server via the TOE protocol.
TOE host	One or several machines located in the DMZ on which the TOE server runs. The TOE host is not part of the TOE itself but forms an important part of the IT environment (namely for the TOE server). The term TOE host includes all software necessary to run the TOE server (including but not limited to the operating system) and software required for WWW access (e.g. web browsers and extensions).
TOE transmission protocol	Set of commands and possible types of information which the TOE client can send to the TOE server combined with the set of commands and possible types of information which can be sent from the TOE server to the TOE client, cf. Section 6.1.1. The general term for the connection between the TOE server and TOE client is “TOE protocol”.
TOE server	Program(s) running on the TOE host to send the (audio-)visual representation of web content to the TOE client (using the TOE protocol) and transfer the user input from the TOE client (received via the TOE protocol) to the browsers running on the TOE host.
TSF data	Data for the operation of the TOE upon which the enforcement of the SFRs relies.
World Wide Web	Within the scope of this PP, the WWW denominates all resources outside the LAN and the DMZ (content, machines) accessible via HTTP or HTTPS.

Acronyms

Acronym	Term
CC	Common Criteria
BSI	Bundesamt für Sicherheit in der Informationstechnik (Federal Office for Information Security)
DMZ	Demilitarised Zone
DOS	Denial of Service
I&A	Identification and Authentication
IT	Information Technology
HTML	Hypertext Markup Language
HTTP	Hypertext Transmission Protocol
HTTPS	Hypertext Transmission Protocol Secure
LAN	Local Area Network

LC	Local Computer
OSP	Organisational Security Policy
PDF	Portable Document Format
PP	Protection Profile
ReCoBS	Remote-Controlled Browsers System
SME	Small and Medium sized Enterprise
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Function
WWW	World Wide Web

9 Literature

Common Criteria

[1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; Version 3.1 Revision 1, September 2006

[2] Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Requirements; Version 3.1 Revision 2, September 2007

[3] Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements; Version 3.1 Revision 2, September 2007

[4] Common Methodology for Information Technology Security Evaluation CEM: Evaluation Methodology, Version 3.1 Revision 2, September 2007

General

[5] Modulare Erweiterung von Sicherheitsgateways, Teil 2, Bundesanzeiger-Verlag 2007, ISBN 978-3-89817-838-1; see also http://www.bsi.de/fachthem/sinet/gefahr/aktiveinhalte/schutzmoeglichkeiten/recobs/index.html

[6] “Aktive Inhalte (Teil 2)” in <kes> 2005#6 pp.54ffl

[7] Anonymisation Services, for a list cf. e.g. to http://www.bsi.bund.de/literat/anonym/literat.htm