Protection Profile for Privilege-Directed Content

Authoriszor Ltd
Drumhill Works
Clayton Lane
Clayton
Bradford
BD14 6RF

Telephone No: 01274 880666

Ref: Auth_CC/PP/DES/01

Issue 1.3

Date: 22nd December 2000

Contents list

References

[CC1] Common Criteria version 2.1, Part 1

[CC2] Common Criteria version 2.1, Part 2

[CC3] Common Criteria version 2.1, Part 3

Abbreviations and Acronyms

EAL	Evaluation Assurance Level
PP	Protection Profile
SFP	Security Function Policy
ST	Security Target
TOE	Target of Evaluation
TSC	TSF Scope of Control
TSF	TOE Security Functions
TSP	TOE Security Policy

1. Protection Profile Introduction

1.1 Protection Profile Identification

1.1.1 Title: Protection Profile for Privilege-Directed Content.

1.1.2 Assurance Level: EAL4.

1.1.3 Registration: LLF/T205

1.2 Protection Profile Overview

1.2.1 This protection profile consists of the standard protection profile sections defined in [CC1].

1.2.2 This protection profile specifies security features and an intended environment of a product designed to protect a website by offering to a web visitor only content consistent with authorisations granted to that visitor, and to protect such a website from subversion. Hereafter within this protection profile, any instance of such a product is referred to as the TOE. The security functionality of such a product is referred to within this Protection Profile as the TSF.

2. TOE Description

2.1 TOE Security Environment

2.1.1 Introduction

2.1.1.1 The TOE is constructed of two parts: a client and a server.

2.1.1.2 The client and server are installed on separate platforms, and interact via the Internet (or over an intranet, extranet or similar network). The interaction is such that the content managed by the server appears to be a normal website accessible over the network, whether or not the client is installed or authenticated.

2.1.1.3 The term “intermediate network” is used hereafter in this protection profile to describe the Internet, intranet, extranet or similar network which is used to access the TOE server.

2.1.1.4 The term “client-side” is used hereafter in this protection profile to describe those users and parts of the TOE which may be situated outside the host organisation and which access the TOE server via the intermediate network. The client-side parts of the TOE are referred to simply as the “client”.

2.1.1.5 The term “server-side” is used hereafter in this protection profile to describe those users and parts of the TOE which are situated inside the host organisation and which have privileges to manage the content or appearance of the website protected by the TOE.

2.1.1.6 The term “public” is used hereafter to describe those client-side users who do not have the TOE client installed, or whose client relates to an TOE server other than the TOE server under consideration.

2.1.1.7 The term “privileged users” is used hereafter in this protection profile to denote those client-side users who have the TOE client installed and are thereby afforded privileges over and above those afforded to “public” users. The apparent content of the website is determined by the privileges afforded to the client, and may range from the equivalent of “public” access to the equivalent of full disclosure of the contents of the website.

2.1.1.8 The term “subject” is used to describe an active entity within the TOE, which may act on behalf of a user or on behalf of the TOE itself; passive entities in the TOE are described as “objects”. This definition follows those given in paragraph 57 of [CC1] and paragraph 30 of [CC2].

2.1.1.9 The Term “Signature” is used hereafter to describe the encrypted information passed by the client to the server for authentication.

2.1.2 Client-Side

2.1.2.1 “Public” users can access the TOE server over the intermediate network, but are (in the recommended configuration) allowed to access only general and innocuous information, on what appears to be a website.

2.1.2.2 The apparent content of the website, as seen by “privileged users”, is determined by the privileges afforded to the client, and may range from the equivalent of “public” access to the equivalent of full disclosure of the contents of the website.

2.1.2.3 The server modifies the apparent content of the website according to the privileges granted to the client; the scripts which determine the website content as seen by any particular client are protected from interference by their location separate from the webroot. In addition, the integrity of the webroot is periodically checked and refreshed, thus providing protection against “file deposition” attacks.

2.1.2.4 The client identifies and authenticates itself to the server (during the processing of each request to access website content) by means of a signature derived from a client activation key supplied by the administrators (and delivered separately from the client software) and individual machine characteristics (e.g. the hard drive serial number, operating system version, system universal unique ID and motherboard Serial No.), thus providing protection against theft of the client software. This identification and authentication process is transparent to the user, whether or not it is successful. If the authentication process is unsuccessful due to corruption or for any other reason, the user will be deemed to be a “non-privileged user”. The user is required to successfully access the TOE server to gain client privileges.

2.1.3 Server-Side

2.1.3.1 The TOE server is managed within the host organisation,via administrative accounts arranged into a hierarchy of privilege groups, of which the group with the highest privilege (below the Operating System Administrators) is System Managers.

2.1.3.2 Allocation of the System Manager privilege (i.e. the addition of users to the System Managers group, or the removal of users from that group) can be performed only by authenticated Operating System Administrators of the server-side host system, while members of the System Managers group are able to allocate and define privileges for the remaining privileged groups, and are able to define the membership of those groups.

2.1.3.3 The privileges of any administrator on the server side are determined by the groups of which that user is a member, and by the privileges assigned to those groups. The term “administrators” is used hereafter in this protection profile to denote members of these privileged groups operating on the server side.

2.1.3.4 The server-side administration facilities include real-time alarms (indicating attempted intrusion or authentication failure), an audit log capable of recording attempted intrusions and authentication failures, facilities to manage users (including revoking the privileges of a client if the client is reported stolen, and the report arrives by a means not controlled by the TOE; or replacing a client key if the client machine is changed or upgraded). The administrator cannot re-instate an authorised user whose attempt to access the TOE server is unsuccessful.

2.1.3.5 The website content displayed to a user is stored remotely, and web pages are generated separately for each information request.

2.1.3.6 The webroot may contain dummy pages (in order to mislead intruders into thinking that these pages are the real source of the website content), and is refreshed periodically, providing a defence against file deposition or modification attacks.

2.2 Assumptions

2.2.1 [A.SERVER_PHYS] It is assumed that the server is maintained in a physically secure location.

2.2.2 [A.SERVER_ADMIN_NOEVIL] It is assumed that the System Managers and other server-side administrators are trusted not to breach the security policy.

2.2.3 [A.SERVER_AUDIT_CHECK] It is assumed that the server-side audit logs are frequently checked by the administrators.

2.2.4 [A.CLIENT_THEFT_REPORT] It is assumed that, if a machine carrying a TOE client is stolen, that the legitimate owner will report this fact as soon as the theft is discovered.

2.2.5 [A.CLIENT_KEY_USE_ON_ISSUE] It is assumed that, when a client activation key is issued to a user, that the user will respond via a predetermined procedure to confirm receipt of this key.

2.2.6 [A.CLIENT_KEY_FAILURE_REPORT] It is assumed that, if a client activation key fails to give the appropriate privileges to the associated user, the user will report the failure.

2.2.7 [A.ALERT] It is assumed that if the TOE generates a real-time warning, the server-side administrators will take prompt and appropriate action. Such action may consist, for instance, of setting a client's access to "public" in response to a report that that client's key has been compromised.

2.2.8 [A.BACKUP] It is assumed that the administrators will use other facilities on the server to make backups as necessary, in order that a 'clean' copy shall exist which can be restored in the case of defacement or corruption of the website, or of the loss of functionality of the server.

2.3 Threats

2.3.1 [T.SERVER_PHYS] There is a threat of physical attack against the server.

2.3.2 [T.SERVER_ADMIN_EVIL] There is a threat of subversion by server-side administrators.

2.3.3 [T.CLIENT_IMPERSONATION] There is a threat that a client-side privileged user may be impersonated by a person who is not authorised to use that user's privileges, and thereby that information may be leaked to such an impostor.

2.3.4 [T.SERVER_INTEGRITY_ATTACK] There is a threat that the server may be attacked and an unauthorised attempt made to modify files on the server.

2.3.5 [T.SERVER_DENIAL_ATTACK] There is a threat that an attempt may be made to prevent the server from processing authorised requests to access information or to perform legitimate administrator functions.

2.3.6 [T.CLIENT_UNATTENDED] There is a threat that a client may be left unattended and may therefore allow impersonation of the client's legitimate user by some other person with physical access to the client machine.

2.3.7 [T.NET_INTERCEPT] There is a threat that information sent over the network may be intercepted by some intruder "sniffing" on the network, and that the information may therefore be leaked to a third party.

2.3.8 [T.REPLAY] There is a threat that information sent over the network may be intercepted and replayed, in order to repeat a transaction which shall legitimately occur once only.

2.3.9 [T.NET_EVIL_COMMAND] There is a threat that an intruder on the intermediate network may introduce a command for which he has no appropriate authority.

2.3.10 [T.KEY_DELIVERY_INTERCEPT] There is a threat that the client key may be intercepted during delivery to the legitimate user.

2.3.11 [T.SERVER_INSECURE] There is a threat that insecurity of the server platform could undermine the security features of the server product, to the extent of allowing an attacker to bypass those security features.

2.4 Organisational Security Policies

2.4.1 [P.INSTALL] The installation process for the TOE server performs an automatic examination of the configuration of the operating system on which TOE server is being installed; warnings are generated of any insecurities found. A policy shall be implemented whereby appropriate action is taken upon such warnings.

2.4.2 [P.AUDIT] A policy shall be implemented to ensure that the audit logs are examined regularly and frequently, and appropriate action taken over any irregularities discovered.

2.4.3 [P.PRIVILEGE] A record shall be kept of all persons to whom privilege is granted, and of the extent of such privileges. This applies both to server-side administrators and to client-side privileged users.

2.4.4 [P.PUBLIC] The information made available to client-side "public" users (those without a relevant TOE client) shall be restricted to generalisations and material suitable for general release.

2.4.5 [P.COMPROMISED_CLIENT] The information made available to client-side privileged users whose clients may have been compromised (e.g. cases of theft or suspected impersonation) shall be restricted to that for "public" users.

2.4.6 [P.ALERT] There shall be a prompt and appropriate response by the administrators in the event of a real-time warning being generated by the TOE.

2.4.7 [P.SERVER_LEAST_PRIVILEGE] The privilege accorded to each server-side administrator shall be the least privilege, consistent with the job function of that administrator.

2.4.8 [P.PRIVILEGE_REVOCATION] When a privileged user or server-side administrator no longer requires a particular privilege, that privilege shall be withdrawn without delay from the user or administrator concerned.

2.4.9 [P.CLIENT_NOT_UNATTENDED] A policy shall be put in place that privileged client-side users shall be required to leave their machines in a secure state (i.e. Switched off, or with a password-protected screen-saver) when unattended.

2.4.10 [P.CLIENT_REVALIDATION] A procedure shall exist whereby a user can request replacement of a client key, and whereby the administrators can comply with this request.

2.4.11 [P.KEY_DISTRIBUTION] A procedure shall exist whereby the administrators can send a client key to a user via a method not controlled by the TOE, and whereby the administrators can verify that the correct user has obtained the key.

3. Security Objectives

3.1 Security Objectives for the TOE

3.1.1 [O.SERVICE] Each client-side privileged user shall be able to access information appropriate to the privilege which he holds, and “public” users shall be able to see a “public” view of the website.

3.1.2 [O.NO_LEAK] Client-side users (whether privileged or “public”) shall not be able to access information in excess of that for their privilege.

3.1.3 [O.DETECT] The TOE shall be able to detect compromise of the TOE client where such compromise results in a change in the detailed system configuration[1] (e.g. in the case of theft of a client key or client activation key).

3.1.4 [O.TRANSPARENT] The TOE shall not signal to a compromised client that it has detected the compromise.

3.1.5 [O.NO_NET_INTRUDER_COMMANDS] The TOE shall be able to detect attempts by intruders on the intermediate network to introduce commands for which they have no appropriate authority (including attempts to record and replay legitimate commands and attempts to masquerade as legitimate users).

3.1.6 [O.SERVER_SECURE] The server platform shall be configured in a secure manner, so as not to afford to an attacker the opportunity to bypass the security features offered by the TOE product, and the security features of the TOE product shall be used to enhance the security of the server platform.

3.1.7 [O.ADMIN] The server-side administrators shall be provided with the facilities required to perform their function.

3.2 Security Objectives for the Environment

3.2.1 [O.NET] The TOE shall be able to operate over the Internet, over intranets and extranets, and over any similar network structure.

3.2.2 [O.SERVER_PHYS] The server shall be protected against physical attack and against physical access by unauthorised persons.

3.2.3 [O.SERVER_ADMIN_NOEVIL] The server administrators shall be worthy of trust that their actions will not breach this security policy.

3.2.4 [O.CLIENT_NOT_UNATTENDED] The client should be under the control of the legitimate user, and not left unattended while it is connected over the network to the server following a legitimate connection.

3.2.5 [O.KEY_DELIVERY_INTERCEPT_DETECT] Procedures shall be able to detect the occurrence, if a key is intercepted during attempted delivery to a legitimate user.

1 [1] The details of exactly which characteristics of the system configuration are encoded into the client key are configurable by the administrators.

4. IT Security Requirements

4.1 TOE Security Requirements

4.1.1 TOE Security Functional Requirements

4.1.1.1 In response to a requirement for webroot integrity (O.SERVER_SECURE), the webroot directory is "swept" periodically, and any unauthorised deletions, modifications and additions are reversed at the next "sweep" (i.e. within a few seconds). This (which could be viewed as a slightly unorthodox variant of FPT_SEP.2, although only FPT_SEP.1 is claimed) protects the website directly against e-vandalism and indirectly against file deposition attacks (e.g. Distributed Denial of Service attacks), the latter because there is only a limited time available to an attacker to use a deposited file before it is made unavailable at the next "sweep". Also, attacks directed at the web server or associated mail server are inhibited by the fact that there is no web server or mail server visible to an attacker. Similarly, the website data and scripts are non-local to the server and hence shielded from attack.

4.1.1.2 The TOE server has the ability to identify TOE clients, both positively and negatively. This could be viewed as FIA_UID.1 in an unorthodox interpretation (in which "public" data could be viewed prior to user identification - which in the case of "public" users would never occur, while privileged users would be identified automatically and transparently); there is also functionality to identify individual TOE clients. All privileged users have a client activation key (distributed separately from the client software - which can be freely distributed). Although it is possible for the key to be stolen (and a procedure P.KEY_DISTRIBUTION shall be in place ensure that the key is correctly delivered, and that the genuine user acknowledges receipt of the key, which is the assumption A.CLIENT_KEY_USE_ON_ISSUE), the TOE checks the machine identity at the time of first use (by using 9 machine parameters, e.g. the hard disk drive serial number, operating systems version,system universal unique ID, motherboard serial No) and stores the client platform profile on both the client and the server. On an attempt to access the website content, the TOE checks the machine identity again and checks the received signature against the expected result. Any discrepancy in the signature causes the client key to reported as stolen and causes the server to return only the "public" version of information requested. Because of the level of detail stored, modifications to the client platform will require a new client key to be installed.

4.1.1.3 The client identification key serves as a client activation key. Although the client software can be made freely available, the key is also required in order to receive non-"public" information.

4.1.1.4 Authentication and verification actions are performed for each access request and for each object on the page.

4.1.1.5 Among the server-side administrators, there is a hierarchy of managers, the highest of which is the System Manager (appointed by an authenticated operating system administrator on the platform hosting the server), who can allocate rights to other types of manager. The next highest is a Site Manager (appointed by a System

Manager) who can allocate rights on a particular site (i.e. within his scope). No manager, at any level, is able to allocate privileges beyond those which he is able to use.

4.1.1.6 There are several audit logs, which are individually switchable between recording “all events”, “no events” (or, where implemented, “critical events”), all such options being configurable by an appropriately privileged manager.

4.1.1.7 The following functional components from [CC2] are included in this Protection Profile.

a. FAU_ARP.1 Security alarms

Hierarchical to: No other components.

FAU_ARP.1.1 The TSF shall take [assignment: *list of the least disruptive actions*] upon detection of a potential security violation.

[Assignment:] The actions to be taken on detection of a potential security violation on the client side are (a) to alert the administrators on the server side in real time, (b) to record (a selection, configurable in advance by server-side administrators, of) relevant information about the incident in the server-side audit log, and (c) to display only “public” information to the corresponding client-side user.

Dependencies: FAU_SAA.1 Potential violation analysis

b. FAU_GEN.1 Audit data generation

Hierarchical to: No other components.

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

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

b) All auditable events for the [selection: *minimum, basic, detailed, not*

specified] level of audit; and

[Selection:] A product satisfying this protection profile is able to generate audit events for the *basic* level of audit.

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

[Assignment:] The following events are specifically defined as auditable: (a) mismatch between a client key and its corresponding machine details; (b) a server-side withdrawal of a client’s key (which would be the result of administrator action in response to a report, delivered via a route not subject to TOE control, that a client key had been stolen or was otherwise untrustworthy).

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

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [assignment: *other audit relevant information*]

[Assignment:] No other audit record content is required.

Dependencies: FPT_STM.1 Reliable time stamps

c. FAU_SAA.1 Potential violation analysis

Hierarchical to: No other components.

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

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

a) Accumulation or combination of [assignment: *subset of defined auditable events*] known to indicate a potential security violation;

[Assignment:] No cumulative or combined rules are defined, that may indicate a violation of the TOE.

b) [assignment: *any other rules*].

[Assignment:] A mismatch between the identity of a client key, the machine data profile stored on the key's installation, and the machine data profile generated during a particular access attempt, is taken as evidence of a violation of the TSP.

Dependencies: FAU_GEN.1 Audit data generation

d. FAU_SAR.1 Audit review

This component will provide authorised users [with] the capability to obtain and interpret the [audit] information. In case of human users this information needs to be in a human understandable presentation. In case of external IT entities the information needs to be unambiguously represented in an electronic fashion.

Hierarchical to: No other components.

FAU_SAR.1.1 The TSF shall provide [assignment: *authorised users*] with the capability to read [assignment: *list of audit information*] from the audit records.

[Assignment:] The users authorised to read audit information are server-side administrators who are members of the relevant server-side management groups.

[Assignment:] No assignment made - the list of audit information is to be determined in the Security Target for each individual system.

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

Dependencies: FAU_GEN.1 Audit data generation

e. FAU_SAR.2 Restricted audit review

Hierarchical to: No other components.

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

Dependencies: FAU_SAR.1 Audit review

f. FAU_SAR.3 Selectable audit review

Hierarchical to: No other components.

FAU_SAR.3.1 The TSF shall provide the ability to perform [selection: *searches, sorting, ordering*] of audit data based on [assignment: *criteria with logical relations*].

[Selection:] No selection made - the selection of audit capabilities is to be defined in the Security Target for each individual product or system; however, the choice of *searches* is expected to be included in the selection.

[Assignment:] No assignment made - the details of the capabilities of the searches/sorting/ordering (whichever is provided according to the previous selection) of audit data are to be decided for each individual system.

Dependencies: FAU_SAR.1 Audit review

g. FAU_SEL.1 Selective audit

Hierarchical to: No other components.

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

a) [selection: *object identity, user identity, subject identity, host identity, event type*]

[Selection:] No selection made - the selection of criteria for inclusion in the list of events actually audited is a matter for individual systems; however it is noted that user identity will be available only in the case of authenticated users (in particular, not in the case of "public" users)

b) [assignment: *list of additional attributes that audit selectivity is based upon*].

[Assignment:] No assignment made - the selection of additional criteria for inclusion in the list of events actually audited is a matter for individual systems.

Dependencies: FAU_GEN.1 Audit data generation

FMT_MTD.1 Management of TSF data

h. FAU_STG.1 Protected audit trail storage

Hierarchical to: No other components.

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

FAU_STG.1.2 The TSF shall be able to [selection: *prevent, detect*] modifications to the audit records.

[Selection:] The TSF shall be able to *prevent* modifications to the audit records.

Dependencies: FAU_GEN.1 Audit data generation

i. FDP_ACC.2 Complete access control

Hierarchical to: FDP_ACC.1

FDP_ACC.2.1 The TSF shall enforce the [assignment: *access control SFP*] on [assignment: *list of subjects and objects*] and all operations among subjects and objects covered by the SFP.

[Assignment:] No assignment made, as the access control SFP is to be decided by individual systems. However, the access control SFP is to be constrained by the following rules: (a) unidentified or unauthenticated user ("public users") are to be allowed only the lowest ("public") level of access; (b) the content of websites is to be writeable only by server-side administrators.

[Assignment:] The access control SFP will apply to all subjects (including all users in all categories) and objects within the TSP, as specified in FDP_ACC.2.2 below.

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

Dependencies: FDP_ACF.1 Security attribute based access control

j. FDP_ACF.1 Security attribute based access control

Hierarchical to: No other components.

FDP_ACF.1.1 The TSF shall enforce the [assignment: *access control SFP*] to objects based on [assignment: *security attributes, named groups of security attributes*].

[Assignment:] No assignment made: the choice of SFP is the domain of individual systems.

[Assignment:] No assignment made: the choice of SFP is the domain of individual systems. However, the attributes shall include the authenticated identity, or lack thereof, of individual users, and the privileges allocated to individual authenticated users (which includes both privileged client-side users and server-side administrators).

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*].

[Assignment:] No assignment made - the precise rules are the domain of individual systems. However, in general there is a hierarchy of privileges, where server-side administrators have greater access rights (e.g. the ability to change the website content) than privileged client-side users, who in turn have greater access rights (in terms of viewing website content) than "public" client-side users.

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*].

[Assignment:] No assignment made - the precise rules are the domain of individual systems.

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*].

[Assignment:] No assignment made - the precise rules are the domain of individual systems. However, the rules shall include the denial of write access to website content to all client-side users (whether privileged or not).

Dependencies: FDP_ACC.1 Subset access control

FMT_MSA.3 Static attribute initialisation

k. FDP_IFC.2 Complete information flow control

Hierarchical to: FDP_IFC.1

FDP_IFC.2.1 The TSF shall enforce the [assignment: *information flow control SFP*] on [assignment: *list of subjects* and *information*] and all operations that cause that information to flow to and from subjects covered by the SFP.

[Assignment:] No assignment made - the precise SFP is a matter for individual systems.

[Assignment:] As required by FDP_IFC.2.2 below, the information flow SFP is to be applied to all subjects and objects in the TSC.

FDP_IFC.2.2 The TSF shall ensure that all operations that cause any information in the TSC to flow to and from any subject in the TSC are covered by an information flow control SFP.

Dependencies: FDP_IFF.1 Simple security attributes

l. FDP_IFF.1 Simple security attributes

Hierarchical to: No other components.

FDP_IFF.1.1 The TSF shall enforce the [assignment: *information flow control SFP*] based on the following types of subject and information security attributes: [assignment: *the minimum number and type of security attributes*].

[Assignment:] No assignment made - the precise SFP is a matter for individual systems.

[Assignment:] No assignment made - the precise number and types of security attributes are matters for individual systems. However, one of the attributes will be the authenticated identity of the client-side user, allowing for the possibility that a client-side user may not have an authenticated identity (i.e. a “public” user will not be authenticated).

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

[Assignment:] No assignment made - the precise SFP is a matter for individual systems.

FDP_IFF.1.3 The TSF shall enforce the [assignment: *additional information flow control SFP rules*].

[Assignment:] No assignment made - the precise SFP is a matter for individual systems.

FDP_IFF.1.4 The TSF shall provide the following [assignment: *list of additional SFP capabilities*].

[Assignment:] No assignment made - the precise SFP is a matter for individual systems.

FDP_IFF.1.5 The TSF shall explicitly authorise an information flow based on the following rules: [assignment: *rules, based on security attributes, that explicitly authorise information flows*].

[Assignment:] No assignment made - the precise SFP is a matter for individual systems. However, specific authenticated users and authenticated administrators, or users and administrators in specific groups defined on the server-side, will have particular privileges which will specifically allow particular classes of information flow.

FDP_IFF.1.6 The TSF shall explicitly deny an information flow based on the following rules: [assignment: *rules, based on security attributes, that explicitly deny information flows*].

[Assignment:] No assignment made - the precise SFP is a matter for individual systems. However, the SFP shall include rules which shall deny write access to website content to all client-side users (whether privileged or not).

Dependencies: FDP_IFC.1 Subset information flow control

FMT_MSA.3 Static attribute initialisation

m. FDP_RIP.1 Subset residual information protection

Hierarchical to: No other components.

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the [selection: *allocation of the resource to, deallocation of the resource from*] the following objects: [assignment: *list of objects*].

[Selection:] no selection made; while “deallocation of the resource from” is probably more appropriate in the application of this section to the creation and deletion of website page components with each request, there is a second application of this section to the server side, where information in an object deallocated from one administrator shall be removed before the object is reallocated and made visible to another administrator; this second application may be implemented in either fashion (by clearing the information on allocation, or on deallocation).

[Assignment:] this requirement applies to the components of website content presented to a user, once the user has relinquished allocation of that (apparent) website.

Dependencies: No dependencies

n. FIA_AFL.1 Authentication failure handling

Hierarchical to: No other components.

FIA_AFL.1.1 The TSF shall detect when [assignment: *number*] unsuccessful authentication attempts occur related to [assignment: *list of authentication events*].

[Assignment:] 1 unsuccessful authentication event.

[Assignment:] This relates to acquisition of the correct machine characteristics (specifications and identification numbers, e.g. hard disk serial numbers) for privileged client-side users. No authentication is required for "public" users.

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall [assignment: *list of actions*].

[Assignment:] In the event of authentication failure for a privileged client-side user, the TSF shall (a) send an immediate alert to the administrators; and (b) allow access for that user only as a "public" user.

Dependencies: FIA_UAU.1 Timing of authentication [this dependency is irrelevant in view of the handling of authentication failures and "public" users].

o. FIA_ATD.1 User attribute definition

Hierarchical to: No other components.

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

[Assignment:] For privileged client-side users, the TSF shall maintain a key which corresponds to a set of characteristics of the individual authorised machine on which the client is installed. The machine characteristics included are a matter for individual systems, but a possible single example would be (e.g.) the hard drive serial number. In addition, for privileged client-side users, the TSF shall keep on the server side a flag denoting whether the client has been invalidated (e.g. as a result of having been reported stolen). No security attributes are required or wanted for "public" users.

Dependencies: No dependencies

p. FIA_SOS.1 Verification of Secrets [assignments from [PP] included]

Hierarchical to: No other components.

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

[assignment] The client keys are verified as correct if they are accepted successfully

Dependencies: No dependencies

q. FIA_SOS.2 TSF Generation of secrets

Hierarchical to: No other components.

FIA_SOS.2.1 The TSF shall provide a mechanism to generate secrets that meet [assignment: *a defined quality metric*].

[Assignment:] No assignment made. This refers to the generation of the client key, which is generated from individual machine characteristics. A suitable quality metric may be the probability that two different user/machine combinations could generate the same key.

FIA_SOS.2.2 The TSF shall be able to enforce the use of TSF generated secrets for [assignment: *list of TSF functions*].

[Assignment:] The client key is used - transparently to the user - for the identification and authentication of privileged client-side users.

Dependencies: No dependencies

r. FIA_UAU.1 Timing of authentication

Hierarchical to: No other components.

FIA_UAU.1.1 The TSF shall allow [assignment: *list of TSF mediated actions*] on behalf of the user to be performed before the user is authenticated.

[Assignment:] Access to the “public” view of the website is allowed without authentication for “public” client-side users only, and before an authorised user is authenticated; however in the case of “public” users, authentication will never occur

FIA_UAU.1.2 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

Dependencies: FIA_UID.1 Timing of identification [this dependency is irrelevant in view of the handling of unidentified “public” users].

s. FIA_UAU.3 Unforgeable authentication

Hierarchical to: No other components.

FIA_UAU.3.1 The TSF shall [selection: *detect, prevent*] use of authentication data that has been forged by any user of the TSF.

[Selection:] The TSF shall prevent the use of forged authentication data: both the client and the server retain copies of a client key which includes machine characteristics. Use of an incorrect key causes the authentication to fail and the user to have access only to the “public” view of the website.

FIA_UAU.3.2 The TSF shall [selection: *detect, prevent*] use of authentication data that has been copied from any other user of the TSF.

[Selection:] The TSF shall prevent the use of copied authentication data. If the key is copied to another machine, it will not match the machine characteristics of the new machine; when an attempt is made to use such a key, it reports itself stolen and is invalidated at the server. The authentication fails, and the user is given access only to the “public” view of the website.

Dependencies: No dependencies

t. FIA_UAU.7 Protected authentication feedback

Hierarchical to: No other components.

FIA_UAU.7.1 The TSF shall provide only [assignment: *list of feedback*] to the user while the authentication is in progress.

[Assignment:] The TSF provides no feedback to the user while the authentication is in progress. Even if the authentication fails, the user is transparently guided to the "public" view of the website.

Dependencies: FIA_UAU.1 Timing of authentication [however, in view of the treatment of failed authentications and "public" users, this dependency is irrelevant].

u. FIA_UID.1 Timing of identification

Hierarchical to: No other components.

FIA_UID.1.1 The TSF shall allow [assignment: *list of TSF-mediated actions*] on behalf of the user to be performed before the user is identified.

[Assignment:] The TSF allows access to the "public" view of the website to users without identification (i.e. before identification, but the identification will never occur).

FIA_UID.1.2 The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

Dependencies: No dependencies

v. FIA_USB.1 User-subject binding

Hierarchical to: No other components.

FIA_USB.1.1 The TSF shall associate the appropriate user security attributes with subjects acting on behalf of that user. [This includes unidentified "public" users].

Dependencies: FIA_ATD.1 User attribute definition

w. FMT_MSA.1 Management of security attributes

Hierarchical to: No other components.

FMT_MSA.1.1 The TSF shall enforce the [assignment: *access control SFP, information flow control SFP*] to restrict the ability to [selection: *change_default, query, modify, delete,* [assignment: *other operations*]] the security attributes [assignment: *list of security attributes*] to [assignment: *the authorised identified roles*].

[Assignment:] The TSF shall enforce the access control SFP...

[Selection and Assignment:] ... to restrict the ability to change_default, modify, query, delete or create ...

[Assignment:] ... the server-side administrator privileges and client-side access privileges ...

[Assignment:] ... the server-side administrators with membership of the appropriate administrator privilege groups.

Dependencies: [FDP_ACC.1 Subset access control or FDP_IFC.1 Subset information flow control]

FMT_SMR.1 Security roles

x. FMT_MSA.3 Static attribute initialisation

Hierarchical to: No other components.

FMT_MSA.3.1 The TSF shall enforce the [assignment: *access control SFP, information flow control SFP*] to provide [selection: *restrictive, permissive, other property*] default values for security attributes that are used to enforce the *SFP*.

[Assignment:] access control SFP

[Selection:] restrictive

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

[Assignment:] server-side administrators with membership of the appropriate administrator privilege groups.

Dependencies: FMT_MSA.1 Management of security attributes

FMT_SMR.1 Security roles

y. FMT_MTD.1 Management of TSF data

Hierarchical to: No other components.

FMT_MTD.1.1 The TSF shall restrict the ability to [selection: *change_default, query, modify, delete, clear*, [assignment: *other operations*]] the [assignment: *list of TSF data*] to [assignment: *the authorised identified roles*].

[Selection and assignment:] query, modify, delete, create

[Assignment:] website content

[Assignment:] (query, modify, delete, create:) server-side administrators with membership of the appropriate administrator privilege groups; (query:) privileged and “public” client-side users as appropriate for their privileges (or lack thereof).

Dependencies: FMT_SMR.1 Security roles

z. FMT_REV.1 Revocation

Hierarchical to: No other components.

FMT_REV.1.1 The TSF shall restrict the ability to revoke security attributes associated with the [selection: *users, subjects, objects, other additional resources*] within the TSC to [assignment: *the authorised identified roles*].

[Selection:] (privileged) users [there being no security attribute to revoke in the case of “public” users].

[Assignment:] server-side administrators with membership of the appropriate administrator privilege groups.

FMT_REV.1.2 The TSF shall enforce the rules [assignment: *specification of revocation rules*].

[Assignment:] A privileged client-side user will have his key revoked if (a) the key is reported stolen, either by the user or by being used on a machine which does not match the stored machine characteristics, or (b) by administrator action if the user is no longer authorised to have the associated privilege.

Dependencies: FMT_SMR.1 Security roles

a.a FMT_SMR.1 Security roles

Hierarchical to: No other components.

FMT_SMR.1.1 The TSF shall maintain the roles [assignment: *the authorised identified roles*].

[Assignment:] (a) server-side administrators (of various kinds as defined by membership of privilege groups defined by the System Manager, who is himself a server-side administrator, or by members of such privilege groups with the privilege to define such privilege groups); (b) privileged users (of various kinds as defined by access control groups defined by the server-side administrators); (c) client-side "public" users (for whom identification and authentication are unnecessary, but who can gain only minimum access to the website).

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

Dependencies: FIA_UID.1 Timing of identification

ab. FPT_RVM.1 Non-bypassability of the TSP

Hierarchical to: No other components.

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

Dependencies: No dependencies

ac. FPT_SEP.1 TSF domain separation

Hierarchical to: No other components.

FPT_SEP.1.1 The TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

FPT_SEP.1.2 The TSF shall enforce separation between the security domains of subjects in the TSC.

Dependencies: No dependencies

ad. FPT_STM.1 Reliable time stamps

Hierarchical to: No other components.

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

Dependencies: No dependencies

4.1.2 TOE Security Assurance Requirements

4.1.2.1 The TOE claims conformance to evaluation level EAL4. The assurance components appropriate to this assurance level are listed below (the requirements for these assurance components being found in [CC3]):

a. [ACM_AUT.1] Configuration Management - Partial CM Automation

b. [ACM_CAP.4] Configuration Management - Generation Support and Acceptance Procedures

c. [ACM_SCP.2] Configuration Management - Problem Tracking CM Coverage

d. [ADO_DEL.2] Delivery and Operation - Detection of Modification

e. [ADO_IGS.1] Delivery and Operation - Installation, Generation and Start-up Procedures

f. [ADV_FSP.2] Development - Fully Defined External Interfaces

g. [ADV_HLD.2] Development - Security Enforcing High-Level Design

h. [ADV_IMP.1] Development - Subset of the Implementation of the TSF

i. [ADV_LLD.1] Development - Descriptive Low-Level Design

j. [ADV_RCR.1] Development - Informal Correspondence Demonstration

k. [ADV_SPM.1] Development - Informal TOE Security Policy Model

l. [AGD_ADM.1] Guidance Documentation - Administrator Guidance

m. [AGD_USR.1] Guidance Documentation - User Guidance

Note that the operation of the TOE is transparent to the unprivileged user, hence no “public” user guide is needed (or wanted). However, for privileged client-side users, documentation is required regarding the installation and use of the client identification/activation key.

n. [ALC_DVS.1] Life-Cycle Support - Identification of Security Measures

o. [ALC_LCD.1] Life-Cycle Support - Developer-Defined Life-Cycle Model

p. [ALC_TAT.1] Life-Cycle Support - Well-Defined Development Tools

q. [ATE_COV.2] Tests - Analysis of Coverage

r. [ATE_DPT.1] Tests - Testing: High-Level Design

s. [ATE_FUN.1] Tests - Functional Testing

t. [ATE_IND.2] Tests - Independent Testing - Sample

u. [AVA_MSU.2] Vulnerability Assessments - Validation of Analysis

v. [AVA_SOF.1] Vulnerability Assessments - Strength of TOE Security

Function Evaluation

w. [AVA_VLA.2] Vulnerability Assessments - Independent Vulnerability Analysis

4.2 Security Requirements for the IT Environment

4.2.1 The intended IT environment is divided into three parts: the client-side machine, the server-side environment, and the network connecting the two (which may be the Internet).

4.2.2 The client-side machine is trusted if and only if there is a client which corresponds to the server under consideration, and has not been detected as compromised; such a client is trusted to receive a particular subset of the available information. Users may also access the website from the client side via untrusted machines (machines with no relevant client key installed, or where the client has been revoked), but will see only the “public” view of the website.

4.2.3 The network (whether the Internet, intranet, extranet or any similar network) is totally untrusted.

4.2.4 The server-side environment must allow connections to the untrusted (Internet or similar) network only via those ports and by those methods which are required for the operation of the server. In addition, the server platform must be configured in a secure manner, so as not to afford to an attacker an opportunity to bypass the security features of the server product.

4.3 Security Requirements for the non-IT Environment

4.3.1 The client-side platform must be physically protected against theft, and the client activation key must be protected against compromise during delivery and during entry into the client-side platform.

4.3.2 The network (whether the Internet, intranet, extranet or any similar network) is totally untrusted.

4.3.3 The server-side environment must be physically secure, and must conform to industry standards for a secure computing environment. The administrators must be trusted not to subvert the security policy. The administrators must configure the privileges of client-side privileged users according to their privilege and need-to-know. The administrators must regularly and frequently check the audit logs produced by the TOE server, and must respond appropriately and promptly to real-time warnings generated by the TOE server.

4.4 Strength of Function

4.4.1 There is a requirement for the Security Target to specify a Strength of Function for those mechanisms which depend on encryption (or on related topics such as random number generation). In this particular case, the required mechanisms are the following:

a. the authentication mechanism;

b. the encryption mechanism which protects the communications between the client and server.

4.4.2 The Strength of Function (SOF) rating claimed is SOF-medium where probabilistic or permutation mechanisms are used, i.e. non-cryptographic mechanisms. There are no TOE security objectives that require the provision of defense against attackers who possess a high attack potential. No SOF is claimed for cryptographic algorithms that may be implemented as these are outside the scope of the PP.

4.4.3 Although the TOE mechanisms in this PP are all cryptographic in nature, in order for this PP to apply to TOE's which have non-cryptographic mechanism and meet the security objectives, SOF-medium needs to be claimed.

5. Application Notes

5.1 Application of the Functionality Components Included in the TSF.

5.1.1 Introduction

5.1.1.1 This section describes in more detail the applicability of the functionality components listed in chapter 4, to TOEs satisfying this Protection Profile.

5.1.1.2 Given the physical partitioning of the TOE into a client side and a server side, separated by an untrusted network (which may be the Internet), and the different populations of users on the two sides (users on the client side and administrators on the server side), it is convenient to consider the client side and the server side separately.

5.1.1.3 [CC3] lists, for each of the functionality components, a set of accountability data, which shall be included in the set of accountability data for which there exists a capability for the TOE to perform audit collection. These requirements for accountability and audit functionality are also collected in this section.

5.1.1.4 There is a single function implemented by a probabilistic or permutational algorithm; this is the creation of the signature from the various machine characteristics.

5.2 Functionality Components Applicable to the Client Side

5.2.1 Client-Side Functionality

5.2.1.1 **FAU_ARP.1 Security Alarms and FIA_AFL.1 Authentication Failure Handling.** The TSF shall detect any failure of the authentication of the signature (for instance, if the client key has been withdrawn or reported stolen, or if the signature does not match the machine parameters), if a client key is installed. In the case of such an authentication failure, the TSF shall (a) alert the administrators on the server side in real time, (b) record in the server-side audit log a pre-configured set of relevant audit information, and (c) ensure that the corresponding client is able to view only the “public” view of the website.

5.2.1.2 **FDP_ACC.2 Complete Access Control, FDP_ACF.1 Security Attribute Based Access Control, FDP_IFF.2 Simple Security Attributes, and FIA_ATD.1 User Attribute Definition.** The access control SFP extends into the client side as a set of access control groups and/or security levels, each carrying a privilege extending the view of the website visible to the corresponding user. The client key identifies the user and thereby allows the setting of security attributes which give membership of the appropriate access control groups; the client key also includes machine characteristics which enable it to report itself stolen if it is moved from one machine to another. In the absence of the client key, or where the client key has been withdrawn, reported stolen, or failed authentication, no access control group privileges are given to the user, and the effect is that the user sees only the "public" view of the website. Client-side users are not given any write access to the website structure (but they may write to forms, e-mail and similar parallel structures).

5.2.1.3 **FDP_IFC.2 Complete Information Flow Control.** The TSF reauthenticates the client key (or lack thereof) on each occasion when a new web image component is sent to the user. The result is that all information sent to the user is mediated according to the access privileges appropriate to the user; in the case of "public" users, this means that the only information available to them is that in the "public" view of the website.

5.2.1.4 **FDP_RIP.1 Subset Residual Information Protection.** The website content seen by the user as a result of each request for information is the resultant content of the various website components sent to that user, and each of these components is generated freshly at the time of the information request. No server-side copy is retained of the website as seen by the user, and any attempt to access the webroot directory will discover that the webroot is either empty or contains dummy files.

5.2.1.5 **FIA_SOS.1 TSF Verification of Secrets, and FIA_SOS.2 TSF Generation of Secrets.** When an authorised user accesses the TOE client for the first time and enters the client activation key, the TSF generates signature information from the user's authentication data and machine characteristics, copies of which are stored both client-side and (when the user first accesses the corresponding TOE server) server-side. This signature is generated cryptographically using any of several algorithms whose quality is well-known. The client keys are verified as correct when successfully accepted.

5.2.1.6 **FIA_UAU.1 Timing of Authentication, FIA_UID.1 Timing of Identification, FIA_UAU.3 Unforgeable Authentication, and FIA_UAU.7 Protected Authentication Feedback.** The identification and authentication of the client key occurs transparently to the client-side user; there is therefore no feedback of identification and authentication results to the user, and no opportunity for the user to forge authentication credentials at the time of identification and authentication, and no opportunity for a user to perform any activity requiring identification and authentication before identification and authentication are performed. However, "public" client-side users are not identified or authenticated at all, but have read access only to the "public" view of the website; thus in the case of "public" users, all such read accesses occur before identification and authentication (in the sense that such identification and authentication will never actually occur). Forgery of a client key, or copying of a client key from one machine to another, is detected cryptographically, and such detection results in the client key reporting itself stolen, and the corresponding users (both the genuine user and the user trying to use the forgery) are allowed access only to the "public" view of the website.

5.2.1.7 **FIA_USB.1 User-Subject Binding.** This is applicable to client-side users in that the access allowed to a user to the website content is dependent on the authentication of subjects acting on the user's behalf, and there is therefore a necessity to bind the user and the corresponding subjects so that their access is administered in a consistent manner.

5.2.1.8 **FMT_MSA.1 Management of Security Attributes, FMT_MTD.1 Management of TSF Data and FMT_REV.1 Revocation.** These components apply to client-side users in that client-side users are not allowed to change security attributes of any users (including revoking such attributes), whether on the client or server side; similarly client-side users are not allowed to change the server-side website structures (but are allowed to write to forms, e-mails and similar parallel structures). However, client-side users are allowed to view website content as allowed by their privileges, with only the minimum privilege being allowed to "public" users (for whom there is no security attribute to revoke).

5.2.1.9 **FMT_SMR.1 Security Roles.** This component applies to client-side users, who are assigned either a "public" role (with no privileges other than to see the "public" view of the website) or a "privileged user" role (of which there may be many types, each with the ability to see a view of the website content which is dependent on the level and type of privilege given).

5.2.1.10 **FPT_RVM.1 Non-bypassability of the TSP.** This component is applicable to client-side users in that the TSP enforcement functions are invoked (automatically, and transparently to the user) before each request for website content is actioned.

5.2.1.11 **FPT_SEP.1 TSF domain separation.** This component is applicable to client-side users in that the verification of the authentication credentials, and the selection of the website content to be presented to the user, are performed entirely on the server side. In addition, the webroot is periodically refreshed from a remote source not visible to the client-side user, thus being protected against modification from the client side.

5.2.2 Accountability and Audit Requirements for Client-Side Functionality

5.2.2.1 The following client-side actions shall be auditable (with the usual caveats that the auditing of these actions need not necessarily be turned on in actual use, and that auditing of the actions of "public" users is limited by their lack of identification information), with the accountability log always kept on the server side:

a. **FDP_ACF.1 Security Attribute Based Access Control, and FDP_IFF.2 Simple Security Attributes.** All decisions on requests for information flow (including requests which result in the "public" view being displayed).

b. **FAU_ARP.1 Security Alarms, FIA_AFL.1 Authentication Failure Handling, and FIA_UAU.3 Unforgeable Authentication.** (a) the occurrence of a failure of the authentication of a client key, and (b) the actions taken (real-time alarms and revocation of the key).

c. **FIA_SOS.1 TSF Verification of Secrets, and FIA_SOS.2 TSF Generation of Secrets, FIA_UAU.1 Timing of Authentication, and FIA_UID.1 Timing of Identification.** Acceptance or rejection of a client key.

d. **FIA_USB.1 User-Subject Binding.** Successful or unsuccessful binding of the security attributes of a subject to the user on whose behalf it acts.

e. **FMT_SMR.1 Security Roles.** Changes in the privileges of individual users.

5.3 Functionality Components Applicable to the Server Side

5.3.1 Server-Side Functionality

5.3.1.1 **FAU_ARP.1 Security Alarms, and FPT_STM.1 Reliable Time Stamps.** In the event of a security violation (e.g. an attempt by a client-side user to modify the webroot, or a report of a stolen client key, delivered by the client key itself as a result of a mismatch between the machine characteristics and the characteristics stored in the machine), the TSF shall (a) alert the administrators on the server side in real time, (b) record (a selection, configurable in advance by server-side administrators of) relevant information about the incident in the server-side audit log (a process which requires the server to maintain a reliable and consistent time-stamping mechanism), and (c) restrict the corresponding client-side user to the "public" view of the website content.

5.3.1.2 **FAU_GEN.1 Audit Data Generation.** The TSF is able to generate an audit record of the following auditable events: (a) Start-up and shutdown of the audit functions; (b) all auditable events for the *basic* level of audit, as specified in [CC2] (these auditable events are enumerated in sections 5.2.2 and 5.3.2 of this document); (c) mismatch between a client key and its corresponding machine details, and (d) server-side withdrawal of a client key (which would be the response (a) to the termination of the client's legitimate reason to view the website, or (b) to a report, delivered by a route not subject to TOE control, that a client key had been stolen or was otherwise untrustworthy). The information in the audit record shall be at least the following: date and time of the event; type of event; subject identity (which could be "unidentified client-side user"; and the outcome (success or failure) of the event.

5.3.1.3 **FAU_SAA.1 Potential Violation Analysis.** A mismatch between the identity of a client key, the machine data profile stored at the time of the key's installation, and the machine data profile generated during a particular access attempt, is taken as evidence of a violation of the TSP; such a mismatch indicates either that the key has been stolen, or that the machine has been modified since the key's installation (in which case the client-side user shall contact the server-side administrators to have the client key replaced).

5.3.1.4 **FAU_SAR.1 Audit Review; FAU_SAR.2 Restricted Audit Review.** Members of particular server-side management groups are given a privilege which allows them to obtain and interpret the audit records. No client-side user is allowed *read* access to the audit records, and no server-side administrator is allowed *read* access to the audit records without being a member of such a privileged group.

5.3.1.5 **FAU_SAR.3 Selectable Audit Review.** The TSF provides a capability of performing logical operations on the audit logs; these operations are not specified in the Protection Profile but are to be defined in any Security Target which uses this Protection Profile. It is expected that the capabilities of the TSF shall include searching the audit log for the record of a particular event.

5.3.1.6 **FAU_SEL.1 Selective Audit.** The TSF offers to server-side administrators who are members of the appropriate privilege group the ability to include or exclude auditable events from the set of events actually audited, based on some function (definable by the administrators) of the event type, user identity (which may be unknown), subject identity and object identity. It is expected that the administrators will choose not to audit the full set of auditable events, in view of the large amounts of audit data thereby generated.

5.3.1.7 **FAU_STG.1 Protected Audit Trail Storage.** The TSF allows *delete* access to the audit records only to server-side administrators who are members of the appropriate privilege group; these administrators are allowed to backup the audit logs to remote storage. The TSF does not allow *modify* access to the audit records to any user or administrator. The TSF gives *append* access to the audit records to all client-side users and server-side administrators, so that their activities can be suitably recorded in the audit log.

5.3.1.8 **FDP_ACC.2 Complete Access Control; FDP_IFC.2 Complete Information Flow Control.** The access/information control SFP applies to all subjects (including all users and administrators in all categories) and objects, and to all operations among subjects and objects covered by the SFP. The access/information control SFP is to be decided by the administrators of the individual installation; however it is constrained so that (a) unidentified or unauthenticated ("public") client-side users, and those client-side users whose client keys are irrelevant to the server under consideration, or have been invalidated or reported stolen, are to be allowed access only to the "public" view of the website; (b) the structures of websites (but not parallel structures such as forms, e-mail, etc.) are to be writeable only by server-side administrators who are members of the appropriate privilege group.

5.3.1.9 **FDP_ACF.1 Security Attribute Based Access Control; FDP_IFF.1 Simple Security Attributes.** The TSF shall allow access to objects (in particular, website components) on the basis of the security attributes of particular subjects (and of the users associated with those subjects). A particularly important attribute in this context is the authenticated identity of the user associated with any given subject (or, in the case of "public" users, the lack of any such authenticated identity), since the privileges of client-side users and server-side administrators (e.g. their membership, or otherwise, of privilege groups) are determinable only once this identity has been determined. There are some particular rules to which the access control policy shall conform: (a) that server-side subjects acting on behalf of client-side users shall have only *append* access to the server-side audit log; (b) that no client-side user shall have *create*, *modify* or *delete* access to website structures (excluding parallel structures such as forms, e-mails, etc., to which *create* and *modify* access is allowed); (c) that unauthenticated "public" client-side users, and client-side users whose key is irrelevant to the server, or has been invalidated or reported stolen, shall have *read* access only to the "public" view of the website (again excluding parallel structures such as forms, e-mails, etc., to which *create* and *modify* access is allowed); (d) that authenticated client-side users shall have *read* access to a view of the website according to their membership of appropriate privilege groups; (e) that server-side administrators shall have access to website content and audit logs according to their membership of appropriate privilege groups.

5.3.1.10 **FDP_RIP.1 Subset Residual Information Protection.** This applies to the server in two distinct areas: (a) in the building of website page components from scratch when they are requested, and their deletion when no longer required, and (b) in the protection of data from one administrator-owned object from reappearance in a subsequent object owned by another administrator. The server issuing a clear cache command after every object that is sent protects the client.

5.3.1.11 **FIA_AFL.1 Authentication Failure Handling.** This requirement applies in two separate areas: (a) in the handling of client-side authentication failures, where there is a relevant client key which is then revoked and a real-time alarm raised, and (b) in the handling of session authentication by administrators on the server side (the server being assumed to be physically secure [A.SERVER_PHYS] and the administrators trusted [A.SERVER_ADMIN_NOEVIL]); the administrators are subject to suitable precautions against attempted impersonation, to be specified in individual installations.

5.3.1.12 **FIA_ATD.1 User Attribute Definition.** This requirement applies in two separate areas: (a) for privileged users on the client side, who have a client key which encodes machine characteristics as a precaution against copying, and who have privileges allocated which determine the views of the website accessible to them, and (b) for administrators on the server side, who have privileges dependent on their membership of particular privilege groups.

5.3.1.13 **FIA_SOS.1 TSF Verification of Secrets, and FIA_SOS.2 TSF Generation of Secrets, and FIA_UAU.3 Unforgeable Authentication.** This refers to the generation by the TSF of the client key which is given to privileged client-side users (a copy being sent to the client on first use, and a second copy being retained at the server). The client key encodes machine characteristics (e.g. the hard drive serial number), as a precaution against copying and theft. Use of a copied client key, or a client key which does not match the machine's characteristics, results in authentication failure (unless the server is configured by the administrators to accept such a key for the purposes of authentication), a real-time alarm at the server, the revocation of the client key, and visibility only of the "public" view of the website.

5.3.1.14 **FIA_UAU.1 Timing of Authentication, and FIA_UID.1 Timing of Identification.** Client-side identification and authentication (or the verification of the absence of a valid client key) are transparent and automatic, and are repeated for each website page component requested. Server-side identification and authentication (for administrators) take place once per session via a userid and password, and no TSF-mediated actions can be performed by an administrator before the administrator is successfully identified and authenticated.

5.3.1.15 **FIA_UAU.7 Protected Authentication Feedback.** This component applies to the response of the server to a client-side request to view a website component. Since the authentication is transparent to the user, there is no explicit feedback to the user about the result of any authentication. However, a privileged client-side user, who is aware of the nature of the product and of the nature of the protection afforded to the website, may deduce the success or failure of the authentication of a client, by the website content returned (in particular, whether there is any returned content which is not part of the "public" view of the website).

5.3.1.16 **FIA_USB.1 User-Subject Binding.** The client-side user or server-side administrator interacts with the product through subjects acting on the user's (or administrator's) behalf. It is a necessary requirement that the actions of each such subject are associated with the corresponding user (or administrator, as appropriate).

5.3.1.17 **FMT_MSA.1 Management of Security Attributes.** This component describes the ability of server-side administrators to specify (and, if necessary, to revoke) the privileges of client-side users and of server-side administrators, subject to membership (by the administrator who is changing the privileges of other users or administrators) of the necessary server-side privilege groups.

5.3.1.18 **FMT_MSA.3 Static Attribute Initialisation.** This component enforces the provision of restrictive access control rules by default, and allows server-side administrators (with membership of the appropriate privilege groups) to specify alternative default behaviour.

5.3.1.19 **FMT_MTD.1 Management of TSF Data.** This component refers to the content of the website protected by the product. The access rules are in broad terms: (a) "public" users are allowed only to view the website, and are allowed to view only the "public" view of the website (and to write to forms, e-mails and similar parallel structures within the "public" view); (b) privileged client-side users are allowed only to view the website, and are allowed to view website content (and to write to forms, e-mails and similar parallel structures) according to their privileges; (c) server-side administrators are allowed, within the scope of the privileges corresponding to their membership of administrator privilege groups, to create, read, update or delete website content, to backup the server, and to modify the membership of privilege groups (for both client-side privileged users and server-side administrators).

5.3.1.20 **FMT_REV.1 Revocation.** This component applies to the ability of server-side administrators (subject to membership of the appropriate privilege groups) to revoke the client keys owned by individual privileged client-side users, and to modify the privilege groups whose membership gives privileges to privileged client-side users and to server-side administrators. One occasion which would cause an administrator to revoke a user's client key would be if the user telephoned the administrators to report the client key (and perhaps also the machine hosting the client key) stolen.

5.3.1.21 **FMT_SMR.1 Security Roles.** This component applies to the partition of the user/administrator community into various subgroups: (a) server-side administrators (of various kinds as defined by membership of privilege groups defined by the System Manager, who is a server-side administrator, or by members of such privilege groups with the privilege to define such privilege groups); (b) privileged users (of various kinds as defined by access control groups defined by the server-side administrators); (c) client-side "public" users (for whom identification and authentication are unnecessary, but who can gain only minimum access to the website).

5.3.1.22 **FPT_RVM.1 Non-Bypassability of the TSP, and FPT_SEP.1 TSF Domain Separation.** The non-bypassability component applies to the client-side users, for whom identification and authentication is an automatic and transparent process (whether successful or not), and for whom there is therefore no opportunity to bypass the TSP, or to attack it through the network protocol. Attempts to use other means to attack the webroot from the client side (e.g. file replacement or file deposition) are countered by separation within the TSP (e.g. the periodic refreshing of any altered content in the webroot directory from a remote 'mirror' source, and the remote placement of the website content).

5.3.2 Accountability and Audit Requirements for Server-Side Functionality

5.3.2.1 The following paragraphs set out the requirements for events which shall be auditable, as required by the *basic* level of accountability and audit, as set out in [CC2]; these are in addition to the events set out as a minimum in section FAU_GEN.1 above. It is noted that only a subset of these actually need to be audited, as there is always a trade-off between obtaining adequate information in the event of an incident, and taking too much information on a routine basis, and thereby filling up the available storage space.

5.3.2.2 **FAU_ARP.1 Security Alarms.** The auditable events required are: The actions taken upon detection of a potential security violation.

5.3.2.3 **FAU_SAA.1 Potential Violation Analysis.** The auditable events required are: (a) the enabling or disabling of any of the analysis mechanisms, and (b) any automated responses performed by the tool (which in the context of this protection profile means any detection of an invalid client key).

5.3.2.4 **FAU_SAR.1 Audit Review.** The auditable events required are: Reading of information from the audit records.

5.3.2.5 **FAU_SAR.2 Restricted Audit Review.** The auditable events required are: Unsuccessful attempts to read information from the audit records.

5.3.2.6 **FAU_SAR.3 Selective Audit Review.** There are no auditable events at the *basic* level of accountability and audit, relating to the analysis of the audit records.

5.3.2.7 **FAU_SEL.1 Selective Audit.** The auditable events required are: All modifications to the audit configuration which occur while the audit functions are operating.

5.3.2.8 **FDP_ACF.1 Security Attribute Based Access Control.** The auditable events required are: All requests to perform an operation on an object covered by the SFP.

5.3.2.9 **FDP_IFF.1 Simple Security Attributes.** The auditable events required are: All decisions on requests for information flow.

5.3.2.10 **FIA_AFL.1 Authentication Failure Handling.** The auditable events required are: (a) the detection of an invalid, stolen or revoked client key; (b) the revocation of a client key.

5.3.2.11 **FIA_SOS.1 TSF Verification of Secrets, and FIA_SOS.2 TSF Generation of Secrets.** The auditable events required are: Rejection or acceptance of any tested secret (i.e. of tested client keys).

5.3.2.12 **FIA_UAU.1 Timing of Authentication and FIA_UID.1 Timing of Identification.** The auditable events required are all use of the identification and authentication mechanism, including the identity of the (server-side) user identifier or client key used. In view of the unconventional handling of unauthenticated "public" users, there is no reason to require the auditing of unauthenticated users accessing the "public" view of the website.

5.3.2.13 **FIA_USB.1 User-Subject Binding.** The auditable events required are successful and unsuccessful attempts to create a subject bound to a particular user or administrator; however there is no reason to require auditing of binding of subjects to unidentified "public" users.

5.3.2.14 **FMT_MSA.1 Management of Security Attributes.** The auditable events required are: All modifications of the values of security attributes (e.g. the addition of a user or administrator to, or removal of a user or administrator from, a privilege group).

5.3.2.15 **FMT_MSA.3 Static Attribute Initialisation.** The auditable events required are: Modifications of the default attributes of newly-created users or administrators.

5.3.2.16 **FMT_MTD.1 Management of TSF Data.** The auditable events required are: All modifications to the value of TSF data (i.e. website content and security attributes of users and administrators).

5.3.2.17 **FMT_REV.1 Revocation.** The auditable events required are: All attempts to revoke security attributes (i.e. to remove users or administrators from privilege groups, or to invalidate client keys), whether successful or not.

5.3.2.18 **FMT_SMR.1 Security Roles.** The auditable events required are: Modifications to the group of users or administrators who comprise a role.

5.3.2.19 **FPT_STM.1 Reliable Time Stamps.** The auditable events required are: Changes to the time as used by the product.

6. Rationale

6.1 Security Objectives Rationale

6.1.1 Chapter 3 of this Protection Profile defines twelve security objectives which determine the functionality presented in this Protection Profile. They are justified in turn in the following paragraphs. The mapping of these objectives to the threats, assumptions and policies is given in a table in section 6.4 of this document.

6.1.2 [O.SERVICE] *There is an objective that client-side privileged users shall be able to access information appropriate to the privilege which they hold.* The rationale for privilege-directed content is that there exists a class of privileged users, whose privileges may vary from one user to another, but whose privilege always exceeds that of the unprivileged member of the public. The rationale for having such a privilege is that a user with such a privilege shall be able to use that privilege to access the appropriate information.

6.1.3 [O.NO_LEAK] *There is an objective that client-side users (whether privileged or "public") shall not be able to access information in excess of that to which they have the appropriate privilege.* This is the converse objective to [O.SERVICE]; the rationale for having a privilege is that it distinguishes those with the privilege from those without the privilege, and the absence of any particular privilege shall therefore act as a disqualification from accessing the information corresponding to that privilege.

6.1.4 [O.DETECT] *There is an objective that the TOE shall be able to detect compromise of the client where such compromise results in a change in the detailed system configuration (e.g. in the case of theft of a client key or activation key).* This objective is present so as to prevent a user (whether privileged or "public") from masquerading as another user (and thereby circumventing the privilege mechanism) by stealing or copying a client key. Note that it is assumed that the policies and assumptions A.CLIENT_THEFT_REPORT, A.CLIENT_KEY_USE_ON_ISSUE, A.CLIENT_KEY_FAILURE_REPORT and P.KEY_DISTRIBUTION are strong enough to ensure that if a client key is stolen before first use, or if a machine carrying a client key is stolen, then the fact of the theft will be promptly reported, so that the administrators can invalidate the compromised key.

6.1.5 [O.TRANSPARENT] *There is an objective that the TOE shall not signal to a compromised client that it has detected the compromise.* The rationale for this objective is that a compromise can be effectively dealt with without alerting the potential attacker, which means that the potential attacker is not prompted to mount a more serious attack. Indeed, the objective can be extended: a protected website does not advertise itself as protected, and members of the public can access the "public" view of the website without any apparent security interference.

6.1.6 [O.NO_NET_INTRUDER_COMMANDS] *There is an objective that the TOE shall be able to detect attempts by intruders on the intermediate to introduce commands for which they have no appropriate authority (including attempts to record and replay legitimate commands and attempts to masquerade as legitimate users).* The rationale for this objective is that otherwise the TOE would be vulnerable to intruders on the untrusted intermediate network, who could masquerade as legitimate users, and could issue commands to be performed as if by the legitimate users, and could record transactions and later replay them, again as if issued by the legitimate users.

6.1.7 [O.SERVER_SECURE] *The server platform shall be configured in a secure manner, so as not to afford to an attacker the opportunity to bypass the security features offered by the server product.* This objective covers the requirement to select appropriate configuration options for the server platform, so as not to undermine the security of the website.

6.1.8 [O.ADMIN] *The server-side administrators shall be provided with the facilities required to perform their function.* This objective covers the requirement for facilities to be made available to authorised and authenticated administrators to perform user account management and user privilege assignment.

6.1.9 [O.NET] *There is an objective that the TOE shall be able to operate over the Internet, over intranets and extranets, and over any similar network structure.* This objective covers the intended use of a product satisfying this Protection Profile: to protect a website on the Internet. This objective implies that the intervening network is totally untrusted, and that there may be users, with access over that network to a protected site, whose intentions are entirely hostile and malicious.

6.1.10 [O.SERVER_PHYS] *The server shall be protected against physical attack and against physical access by unauthorised persons.* This objective covers the physical defence of the server against direct physical attack and against intrusion via the administrator interfaces colocated with the server.

6.1.11 [O.SERVER_ADMIN_NOEVIL] *The server administrators shall be worthy of trust that their actions will not breach this security policy*. This objective covers the trust which must be placed in those who must administer the system, and whose privileged access means that electronic measures alone are not sufficient to defend the system, shall they be malicious. Personnel measures are therefore required, to ensure that the administrators are not malicious.

6.1.12 [O.CLIENT_NOT_UNATTENDED] *The client should be under the control of the legitimate user, and not left unattended while it is connected over the network to the server following a legitimate connection*. This objective counters the possibility that an authenticated, privileged user may allow an unauthorised person to access the client by leaving the client unattended (and therefore unsupervised) but connected to the server via the network, allowing the unauthorised person to impersonate the legitimate user.

6.1.13 [O.KEY_DELIVERY_INTERCEPT_DETECT] *Procedures shall be able to detect the occurrence, if a key is intercepted during attempted delivery to a legitimate user*. This objective covers the requirement that keys shall be delivered to legitimate users, and imposes a requirement for a detection mechanism if that key delivery fails.

6.2 Security Requirements Rationale

6.2.1 Chapter 4 of this Protection Profile lists a large number of requirements taken from [CC2]. The following paragraphs map these requirements to a smaller and more manageable set of requirements related to the actual configuration and usage of a system or product such as may be built to implement this Protection Profile. A table giving the mappings between objectives, threats, policies and assumptions is given in section 6.4 of this document.

6.2.2 **Authentication of the client to the server by means of a cryptographically-generated key derived from machine characteristics.** This requirement, which derives from the objectives O.SERVICE, O.TRANSPARENT and O.NO_LEAK, derives from the requirement to identify and authenticate privileged users without requiring identification and authentication from "public" users; it maps to the following [CC2] requirements:

a. FIA_ATD.1 User Attribute Definition;

b. FIA_SOS.1 Verification of Secrets

c. FIA_SOS.2 TSF Generation of Secrets;

d. FIA_USB.1 User-Subject Binding.

6.2.3 **Re-authentication of the client key against the machine characteristics on each use, and invalidation of the client key if it fails re-authentication, or if the client key has been revoked by an administrator.** This requirement, which maps to the objective O.DETECT, derives from a requirement to maintain the currency of identification and authentication information, and to detect client keys which have been copied from one machine to another; it maps to the following [CC2] requirements:

a. FIA_AFL.1 Authentication Failure Handling;

b. FIA_UAU.3 Unforgeable Authentication;

c. FPT_RVM.1 Non-bypassability of the TSP.

6.2.4 **Provision of an interface to an untrusted network, which appears from the client side to be a website, whose apparent content is dependent on the privileges accorded to the client.** This requirement, which maps to the objectives O.NET, O.SERVICE and O.NO_LEAK, derives from a requirement for the website content to be appropriate to the privileges (if any) granted to each user, including "public" users; the requirement maps to the following [CC2] requirements:

a. FDP_ACC.2 Complete Access Control;

b. FDP_ACF.1 Security Attribute Based Access Control;

c. FDP_IFC.2 Complete Information Flow Control;

d. FDP_IFF.1 Simple Security Attributes.

6.2.5 **Presentation of a "public" view of the website to those users who do not have the client software, or who have the client software but no client key, or whose client key has been invalidated.** This requirement, which maps to the objectives O.SERVICE and O.NO_LEAK, derives from a requirement not to require identification and authentication from members of the public, and for the protection of the website to be unobtrusive (so that the protected website appears to be the same as any other website, as far as the client can see); the requirement maps to the following [CC2] requirements:

a. FDP_ACC.2 Complete Access Control;

b. FDP_ACF.1 Security Attribute Based Access Control;

c. FDP_IFC.2 Complete Information Flow Control;

d. FDP_IFF.1 Simple Security Attributes;

e. FIA_AFL.1 Authentication Failure Handling;

f. FIA_UAU.1 Timing of Authentication;

g. FIA_UID.1 Timing of Identification.

6.2.6 **Transparency to the user of all identification and authentication.** This requirement, which maps to the objective O.TRANSPARENT, derives from a requirements (a) for ease of use, (b) not to require identification and authentication from members of the public, and (c) for the protection of the website to be unobtrusive (so that the protected website appears to be the same as any other website, as far as the client can see); the requirement maps to the following [CC2] requirements:

a. FIA_UAU.1 Timing of Authentication;

b. FIA_UAU.7 Protected Authentication Feedback;

c. FIA_UID.1 Timing of Identification;

d. FPT_RVM.1 Non-bypassability of the TSP.

6.2.7 **Provision of a server-side audit log and of real-time alarms to the administrators on the server-side, triggered in the event of authentication failures or attempted intrusions.** This requirement, which maps to the objective O.DETECT and O.NO_NET_INTRUDER_COMMANDS, derives from a requirement for administrators to be alerted, so that they can take appropriate action in the event of an incident, and so that the appropriate records can be kept in the event of an incident; the requirement maps to the following [CC2] requirements:

a. FAU_ARP.1 Security Alarms;

b. FAU_GEN.1 Audit Data Generation;

c. FAU_SAA.1 Potential Violation Analysis;

d. FAU_SAR.1 Audit Review;

e. FAU_SAR.2 Restricted Audit Review;

f. FAU_SAR.3 Selectable Audit Review;

g. FAU_SEL.1 Selective Audit;

h. FAU_STG.1 Protected Audit Trail Storage;

i. FIA_AFL.1 Authentication Failure Handling;

j. FPT_STM.1 Reliable Time Stamps.

6.2.8 **Remote storage of the website content, administrative access to website content, and generation of web pages separately for each request for access.** This requirement, which maps to the objective O.SERVER_SECURE, derives from requirements to protect the website content against unauthorised modification from the client side, but to allow it to be managed from the server side; it maps to the following [CC2] requirements:

a. FDP_RIP.1 Subset Residual Information Protection;

b. FMT_MTD.1 Management of TSF Data.

6.2.9 **Remote execution of the TSF applications, and refreshment of the webroot directory, as a defence against file modification and deposition attacks.** This requirement, which maps to the objective O.SERVER_SECURE, derives from requirements to protect the execution of the security functions, and to protect the webroot, against unauthorised modification from the client side; it maps to the following [CC2] requirements:

a. FPT_SEP.1 TSF Domain Separation.

6.2.10 **Management Facilities for Server-Side Administrators.** This requirement, which maps to the objective O.ADMIN, derives from the requirement for server-side administrators to be able to assign appropriate rights to privileged users (and to each other), and to be able to revoke rights if it becomes inappropriate for a particular user to hold particular rights (or if a client key is reported stolen, the report arriving via a route not under the TOE’s control). This requirement corresponds to the following [CC2] requirements:

a. FMT_MSA.1 Management of Security Attributes;

b. FMT_MSA.3 Static Attribute Initialisation;

c. FMT_REV.1 Revocation;

d. FMT_SMR.1 Security Roles.

6.2.11 It is noted that a product or system satisfying this Protection Profile employs unconventional handling (by the standards of other products and systems) of unidentified “public” users, and of those users whose identification and authentication fails. As a result, the following dependencies are rendered irrelevant, although they are in fact satisfied:

a. FIA_AFL.1 (Authentication Failure Handling) has a dependency on FIA_UAU.1 (Timing of Authentication);

b. FIA_UAU.1 (Timing of Authentication) has a dependency on FIA_UID.1 (Timing of Identification);

c. FIA_UAU.7 (Protected Authentication Feedback) has a dependency on FIA_UAU.1 (Timing of Authentication).

6.3 Coverage of Threats

6.3.1 Section 2.3 of this Protection Profile indicates 11 threats. These threats are countered as follows:

a. Threat T.SERVER_PHYS is the threat of physical attack against the server. This is countered by the assumption A.SERVER_PHYS, that the server location is secure.

b. Threat T.SERVER_ADMIN_EVIL is the threat of subversion of the installation by the administrators. This is countered by the assumption A.SERVER_ADMIN_NOEVIL, that the administrators are sufficiently trustworthy. In addition, there are accountability and audit facilities on the server side.

c. Threat T.CLIENT_IMPERSONATION is the threat of impersonation of a privileged user by another person. This is countered by the identification and authentication features related to the client key. The threat of theft of a machine with a valid client key is countered by the assumption A.CLIENT_THEFT_REPORT (that the genuine user will report the theft) and the organisational policy P.PRIVILEGE_REVOCATION (that the administrators, on hearing the report of the theft, will revoke the client key). The organisational policy P.COMPROMISED_CLIENT is also relevant, and states that the information available to a client known to be compromised shall be restricted to that legitimately available to the public.

d. The threats T.SERVER_INTEGRITY_ATTACK and T.SERVER_DENIAL_ATTACK are countered by the functionality where the webroot and the website content are stored remotely and periodically refreshed. In addition, there are audit logs and real-time alarms; the organisational policy P.ALERT and the assumption A.ALERT state that the administrators will act on such alarms, and the organisational policy P.AUDIT and the assumption A.SERVER_AUDIT_CHECK state that the administrators will check the audit logs and will act on the results. Assumption A.BACKUP states that the backup facilities available on the server machine will be used to create a 'clean' version of the website which can be used to replace the original, shall all other precautions fail.

e. The threat T.CLIENT_UNATTENDED is countered by education of privileged users, so that they do not leave their machines unattended; since the identification and authentication relate to the machine and not directly to the user, this is countered by an organisational policy P.CLIENT_NOT_UNATTENDED.

f. The threats T.NET_INTERCEPT, T.REPLAY and T.NET_EVIL_COMMAND are countered by using encryption when required, and by identifying the client by its client key (as well as by its IP address). The server is able (given a compatible browser at the client machine) to encrypt data either at its own request or at the request of the client machine (e.g. when emulating the https protocol). The protection given by the encryption is exactly what would be achieved by using the same protocol to access a conventional website.

g. The threat T.KEY_DELIVERY_INTERCEPT is countered by the procedure P.KEY_DISTRIBUTION and by the assumptions A.CLIENT_KEY_USE_ON_ISSUE and A.CLIENT_KEY_FAILURE_REPORT (i.e. a procedure for issuing the client key to the legitimate user, an assumption that the legitimate user will use the client key as soon as it is received, and an assumption that the user will report any failure of the client key to operate as expected, as would occur if the legitimate user is locked out by an intruder equipped with that client key). The organisational policies P.COMPROMISED_CLIENT and P.CLIENT_REVALIDATION are also relevant, and state that the information available to a client known to be compromised shall be restricted to that legitimately available to the public, and that a legitimate user shall be able to get his client key replaced if it becomes in any way invalidated.

h. The threat T.SERVER_INSECURE, relating to insecure server configuration, is countered by the procedure P.INSTALL (the installation procedure gives warnings of insecure configuration of the server platform) and the assumption A.ALERT (that the administrators will take appropriate action on receiving such a warning).

6.3.2 A table giving the mappings between objectives, threats, policies and assumptions is given in section 6.4 of this document.

6.4 Mapping of Objectives, Threats, Policies and Assumptions

6.4.1 The following table gives a mapping between the objectives (O), policies (P), assumptions (A) and threats (T).

Objectives	Procedures / Assumptions	Threats
O.SERVER_PHYS	A.SERVER_PHYS	T.SERVER_PHYS
O.SERVER_ADMIN_NOEVIL	A.SERVER_ADMIN_NOEVIL, P.AUDIT, A.SERVER_AUDIT_CHECK, P.ALERT, A.ALERT, P.PRIVILEGE	T.SERVER_ADMIN_EVIL
O.DETECT, O.TRANSPARENT, O.NO_LEAK	A.CLIENT_THEFT_REPORT, P.PRIVILEGE_REVOCATION, P.COMPROMISED_CLIENT, P.PUBLIC	T.CLIENT_IMPERSONATION
O.SERVICE, O.NET	P.ALERT, A.ALERT, P.AUDIT, A.SERVER_AUDIT_CHECK, A.BACKUP, P.PRIVILEGE	T.SERVER_INTEGRITY_ATTACK, T.SERVER_DENIAL_ATTACK
O.CLIENT_NOT_UNATTENDED	P.CLIENT_NOT_UNATTENDED	T.CLIENT_UNATTENDED
O.NO_NET_INTRUDER_COMMANDS, O.NET	P.ALERT, A.ALERT, P.AUDIT, A.SERVER_AUDIT_CHECK, A.BACKUP	T.NET_INTERCEPT, T.REPLAY, T.NET_EVIL_COMMAND
O.KEY_DELIVERY_INTERCEPT_DETECT	P.KEY_DISTRIBUTION, A.CLIENT_KEY_USE_ON_ISSUE, A.CLIENT_KEY_FAILURE_REPORT, P.COMPROMISED_CLIENT, P.CLIENT_REVALIDATION	T.KEY_DELIVERY_INTERCEPT
O.SERVER_SECURE	P.INSTALL, A.ALERT	T.SERVER_INSECURE
O.ADMIN	P.SERVER_LEAST_PRIVILEGE	T.SERVER_DENIAL_ATTACK, T.SERVER_INTEGRITY_ATTACK

6.5 Completeness of Resolution of Assignments and Selections in Criteria in [CC2]

6.5.1 The following table indicates the state of completeness of the assignments and selections left incomplete in the various criteria in [CC2]. All of these criteria have been given in the list at section 4.1.1.7 of this document, together with a rationale for those criteria where choices still have to be made. Some criteria are complete as they stand in [CC2], and these are omitted from this table.

Criterion	Reference	Completion state
FAU_ARP.1	4.1.1.7(a)	Assignment COMPLETE
FAU_GEN.1	4.1.1.7(b)	Selection and 2 assignments COMPLETE
FAU_SAA.1	4.1.1.7(c)	2 assignments COMPLETE
FAU_SAR.1	4.1.1.7(d)	2 assignments COMPLETE
FAU_SAR.3	4.1.1.7(f)	Selection and assignment INCOMPLETE
FAU_SEL.1	4.1.1.7(g)	Selection and assignment INCOMPLETE
FAU_STG.1	4.1.1.7(h)	Selection COMPLETE
FDP_ACC.2	4.1.1.7(i)	1st assignment INCOMPLETE; 2nd assignment

		COMPLETE
FDP_ACF.1	4.1.1.7(j)	5 assignments INCOMPLETE
FDP_IFC.2	4.1.1.7(k)	1st assignment INCOMPLETE; 2nd assignment COMPLETE
FDP_IFF.1	4.1.1.7(l)	7 assignments INCOMPLETE
FDP_RIP.1	4.1.1.7(m)	Selection INCOMPLETE; assignment COMPLETE
FIA_AFL.1	4.1.1.7(n)	3 assignments COMPLETE
FIA_ATD.1	4.1.1.7(o)	Assignment COMPLETE
FIA_SOS.2	4.1.1.7(q)	1st assignment INCOMPLETE; 2nd assignment COMPLETE
FIA_UAU.1	4.1.1.7(r)	Assignment COMPLETE
FIA_UAU.3	4.1.1.7(s)	2 selections COMPLETE
FIA_UAU.7	4.1.1.7(t)	Assignment COMPLETE
FIA_UID.1	4.1.1.7(u)	Assignment COMPLETE
FMT_MSA.1	4.1.1.7(w)	Selection and 4 assignments COMPLETE
FMT_MSA.3	4.1.1.7(x)	Selection and 2 assignments COMPLETE
FMT_MTD.1	4.1.1.7(y)	Selection and 2 assignments COMPLETE
FMT_REV.1	4.1.1.7(z)	Selection and 2 assignments COMPLETE
FMT_SMR.1	4.1.1.7(aa)	Assignment COMPLETE