1.

Adaptive Server Anywhere Security Target

Version 1.0

04/11/06

Prepared for:

iAnywhere™ Solutions, Inc.

a Sybase company

One Sybase Drive
Dublin, CA 94568

Prepared By:

Science Applications International Corporation

Common Criteria Testing Laboratory

7125 Columbia Gateway Drive, Suite 300
Columbia, MD 21046

LIST OF TABLES

1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is Adaptive Server Anywhere developed by iAnywhere Solutions, Inc. a subsidiary of Sybase, Inc. (hereafter referred to as Sybase). Adaptive Server Anywhere is a relational database management system (RDBMS) server that operates in the context of a commercial operating system, providing services to both local and remote clients.

The Security Target contains the following additional sections:

- TOE Description (Section 2)
- Security Environment (Section 3)
- Security Objectives (Section 4)
- IT Security Requirements (Section 5)
- TOE Summary Specification (Section 6)
- Protection Profile Claims (Section 7)
- Rationale (Section 8).

1.1 Security Target, TOE and CC Identification

ST Title – Sybase Adaptive Server Anywhere Security Target

ST Version – Version 1.0

ST Date – 04/11/06

TOE Identification – Sybase Adaptive Server Anywhere (versions 9.0.1 and 9.0.2) component of SQL Anywhere Studio 9; specifically:

- Adaptive Server Anywhere version 9.0.2 build 3221 for Microsoft Windows XP, Windows 2000, and Windows 2003 Server.
- Adaptive Server Anywhere version 9.0.2 build 3219 for Sun Solaris 8, and Redhat Linux Advanced Server 2.1.
- Adaptive Server Anywhere version 9.0.1 build 2085 for Microsoft Windows XP, Windows 2000, Windows 2003 Server, Sun Solaris 8, and Redhat Linux Advanced Server 2.1.

CC Identification – Common Criteria for Information Technology Security Evaluation, Version 2.1, August 1999, ISO/IEC 15408.

1.2 Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security functional requirements, Version 2.1, August 1999, ISO/IEC 15408-2.
    - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance requirements, Version 2.1, August 1999, ISO/IEC 15408-3.
    - Part 3 Conformant

- EAL 3 augmented with ALC_FLR.2

1.3 Conventions, Terminology, Acronyms

This section specifies the formatting information used in the Security Target.

1.3.1 Conventions

The following conventions have been applied in this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
  - Iteration: allows a component to be used more than once with varying operations. In the ST, iteration is indicated by a letter in parenthesis placed at the end of the component. For example FDP_ACC.1a and FDP_ACC.1b indicate that the ST includes two iterations of the FDP_ACC.1 requirement, a and b.
  - Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]).
  - Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., [***selection***]).
  - Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …").
- Explicit Security Functional Requirements are identified with the following symbol suffix: "_EXP", for example FTA_MCS_EXP.
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

2. TOE Description

The Target of Evaluation (TOE) is Sybase Adaptive Server Anywhere (versions 9.0.1 and 9.0.2) component of SQL Anywhere Studio 9 - configured and operated according to the guidance documents identified later in this Security Target. Note that the rest of this discussion focuses on the Adaptive Server Anywhere (ASA) portion of SQL Anywhere Studio 9 and it should be understood that only the ASA security features of SQL Anywhere Studio has been subject to evaluation in the context of this Security Target.

ASA is designed to execute as a set of applications in the context of a number of commercially available operating systems. For the purpose of evaluation, ASA was evaluated in the context of and tested on Microsoft Windows 2000, XP and Server 2003, Sun Solaris 8, and Redhat Linux Advanced Server 2.1.

2.1 TOE Overview

ASAprovides relational database technology designed to support multiple operating systems as well as allowing it to operate efficiently with limited memory, CPU power, and disk space. Core features such as the query optimizer and the data caching mechanism are designed specifically to operate without extravagant use of resources. At the same time, ASA contains the features needed to take advantage of workgroup servers, including support for many users, scalability over multiple CPUs, and advanced concurrency features.

ASA runs as applications on top of an operating system and depends on the services exported by the operating system to function. ASA uses operating system services for process creation and manipulation; device and file processing; shared memory creation and manipulation; provision of the network stack up through the TCP layer; and security requests such as inter-process communication. The hardware upon which the operating system runs is completely transparent to ASA - ASA sees only the operating system's user interfaces.

2.2 TOE Architecture

The components that make up the majority of the Target of Evaluation (TOE) are:

- ASA Server - an operating system process that is running the ASA Server executable. The server is multi-threaded and may be running on several processors simultaneously.
- Operating system files - The database is stored in one or more operating system files. A number of operating system files are also used by the ASA Server for configuration.

In addition to the components identified above, the TOE includes a set of programs that provide the interfaces necessary for TOE administration.

2.2.1 Physical Boundaries

There are two mechanisms available to communicate with the ASA Server: the Command Sequence protocol (CmdSeq) and the Tabular Data Stream (TDS) Protocol.

These protocols are used to request ASA Server services. They are used by untrusted client processes, via routines in provided libraries, to communicate with the Server. Administrators interface with the ASA Server via utility programs provided to facilitate ASA administration. These utility programs use available library routines, just like other untrusted clients, to interface to CmdSeq and/or TDS, which in turn communicates with the ASA Server over TCP/IP or other network protocols implemented by the hosting operating system.

The CmdSeq and TDS protocols facilitate communication with the ASA Server via messages and streams. The functionality of the ASA Server is reflected by the type of messages/streams and their contents.

2.2.2 Logical Boundaries

The TOE logically supports the following security functions at its available interfaces (CmdSeq, TDS, and administrator programs):

- Security audit
- User data protection
- Identification and authentication
- Security Management,
- Protection of the TSF, and
- TOE access.

Note that while ASA may include additional functions, only these security functions, only to the extend described in this ST, have been subject to evaluation.

2.2.2.1 Security audit

ASA has an audit mechanism that is invoked for access checks, authentication attempts, administrator functions, and at other times during its operation. When invoked, the date, time, responsible individual and other details describing the event are recorded to the audit trail.

The Audit log is stored in the transaction log which is protected from unauthorized access or modification. The dbtran utility can be used by authorized administrators to extract the audit records from the transaction log file, including searching by user identities. The resulting text file can then be used by the administrator in any manner to effectively review the audit trail.

2.2.2.2 User data protection

ASA implements a Discretionary Access Control Policy over applicable database objects - tables, views, stored procedures and user-defined functions. Note that there are other database objects that are either always private, always public, or are part of one of the afore-mentioned objects. Regardless, each object has an owner which is the creator of the object. Object owners have special permissions, while other users can subsequently be granted specific access permissions allowing corresponding operations on the object.

2.2.2.3 Identification and authentication

ASA provides its own identification and authentication mechanism in addition to the underlying operating system. Users must provide a valid username and password before they can access any security-related functions. Once identified and authenticated, all subsequent actions are associated with that user and policy decisions are based on the user's identity, group memberships and corresponding authorities.

2.2.2.4 Security management

ASA provides SQL statements and built-in procedures necessary to manage users and associated attributes, access privileges, and other security functions such as audit. The functions are restricted based on user authorities, originally restricted to administrators only. While all of the administrative functions are available through and restricted at the ASA Server interfaces, utility programs are provided to facilitate ASA administrators.

ASA associated authorities with users, including a Database Administrator (DBA) authority that can manage the behavior of the applicable security functions. The DBA(s) is considered an authorized administrator (or trusted user) and all other users are simply referred to as users (or untrusted users).

2.2.2.5 Protection of the TSF

ASA protects itself and ensures that its policies are enforced in a number of ways. While there is dependence on the underlying operating system to separate its process constructs, enforce file and memory access restrictions, and to provide communication services, ASA protects itself by keeping its context separate from that of its users and also by making effective use of the operating system mechanisms to ensure that memory and files used by ASA have the appropriate access settings. Furthermore, ASA interacts with users through well-defined interfaces designed to ensure that the ASA security policies are always enforced.

2.2.2.6 TOE access

ASA allows authorized administrators to define stored procedures (presented in the guidance documentation) that will be activated by events when users connect. Once defined and activated the stored procedures can restrict the number of concurrent sessions, specific user identities, and whether the session is allowed at the current time.

2.3 TOE Documentation

Sybase offers a series of documents that describe the installation process for Adaptive Server Anywhere as well as guidance for subsequent use and administration of the applicable security features. Refer to Section 6 for information about these and other documentation associated with Adaptive Server Anywhere.

3. Security Environment

The security environment for the functions addressed by this specification includes threats, security policies, and usage assumptions, as discussed below.

3.1 Organizational Policies

P.ACCOUNTABILITY	The users of the TOE shall be held accountable for their actions within the TOE.
P.AUTHORIZED_USERS	Access controls will ensure that only those users who have been authorized to access the protected information within the TOE will be able to do so.
P.NEED_TO_KNOW	The TOE must limit the access to information in protected resources to those authorized users who have a need to know that information.
P.ROLES	The TOE shall provide an authorized administrator role for secure administration of the TOE. This role shall be separate and distinct from other authorized users.

3.2 Threats

T.ADMIN_ERROR	An authorized administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.
T.AUDIT_COMPROMISE	A process or user may cause audit data to be inappropriately accessed (viewed, modified or deleted), or prevent future records from being recorded, thus masking an attacker's actions.
T.MASQUERADE	An unauthorized user, process, or external IT entity may masquerade as an authorized entity to gain access to data or TOE resources.
T.RESIDUAL_DATA	A user or process may gain unauthorized access to data through reallocation of TOE resources from one user or process to another.
T.SYSACC	A malicious process or user may gain unauthorized access to the authorized administrator account, or that of other trusted personnel.
T.TSF_COMPROMISE	A malicious user or process may cause the TOE, configuration data, or sensitive user data to be inappropriately accessed (viewed, modified or deleted) allowing a breach in the TSF security policies.
T.UNAUTH_ACCESS	A user may gain unauthorized access (view, modify, delete) to user data.
T.UNDETECTED_ACTIONS	Failure of the IT operating system to detect and record unauthorized actions may occur.
T.UNIDENTIFIED_ACTIONS	Failure of the authorized administrator to identify and act upon unauthorized actions may occur.

3.3 Assumptions

A.NO_EVIL	Authorized administrators are non-hostile, appropriately trained and follow all administrator guidance.
A.NO_GENERAL_PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on DBMS servers, other than those services necessary for the operation, administration and support of the DBMS.
A.PHYSICAL	It is assumed that appropriate physical security is provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.
A.ROBUST_ENVIRONMENT	It is assumed that the IT environment protects the TOE (and its resources) and provides time stamps with at least the same degree of assurance as that claimed by the TOE.

4. Security Objectives

This section defines the security objectives for the TOE and its environment. These objectives are suitable to counter all identified threats and cover all identified organizational security policies and assumptions. The TOE security objectives are identified with 'O.' inserted at the beginning of the name and the environment objectives are identified with 'OE.' inserted at the beginning of the name.

4.1 Security Objectives for the TOE

O.ACCESS	The TOE will ensure that users gain only authorized access to it and to the resources that it controls.
O.ADMIN_ROLE	The TOE will provide authorized administrator roles to isolate administrative actions.
O.AUDIT_GENERATION	The TOE will provide the capability to detect and create records of security relevant events associated with users.
O.AUDIT_PROTECTION	The TOE will provide the capability to protect audit information.
O.AUDIT_REVIEW	The TOE will provide the capability to selectively view audit information.
O.DISCRETIONARY_ACCESS	The TOE will control access to resources based upon the identity of users or groups of users.
O.INTERNAL_TOE_DOMAINS	The TSF will maintain internal domains for separation of data and queries belonging to concurrent users.
O.MANAGE	The TOE will provide all the functions and facilities necessary to support the authorized administrators in their management of the security of the TOE.
O.PROTECT	The TOE will provide mechanisms to protect user data and resources.
O.RESIDUAL_INFORMATION	The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated.
O.TOE_PROTECTION	The TOE will protect itself and its assets from external access, interference and tampering.[1]
O.USER_AUTHENTICATION	The TOE will verify the claimed identity of users.
O.USER_IDENTIFICATION	The TOE will uniquely identify users.

[1] Note that in a distributed configuration the TOE sends passwords in clear text and as such it is expected that the IT environment would be configured to mitigate the risk of another user obtaining access to that information.

4.2 Security Objectives for the IT Environment

OE.TIME	The IT environment will provide a time source that provides reliable time stamps.
OE.TOE_PROTECTION	The IT environment will provide protection to the TOE and its assets from external access, interference and tampering.

4.3 Security Objectives for the Environment

OE.ADMIN_GUIDANCE	The TOE will provide authorized administrators with the necessary information for secure management of the TOE.
OE.CONFIG	The TOE will be installed, configured, managed and maintained in accordance with its guidance documentation and applicable security policies and procedures by appropriately trained and trusted administrator personnel.
OE.INSTALL	The TOE will be delivered with the appropriate installation guidance to establish and maintain TOE security.
OE.NO_GENERAL_PURPOSE	There will be no general-purpose computing capabilities (e.g., compilers or user applications) available on DBMS servers, other than those services necessary for the operation, administration and support of the DBMS.
OE.PHYSICAL	Physical security will be provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.
OE.ROBUST_ENVIRONMENT	The IT environment that supports the TOE for enforcement of its security objectives will be of at least the same level of assurance as the TOE.
OE.SELF_PROTECTION	The IT environment and its assets will be protected from external interference, tampering or unauthorized disclosure.
OE.TRUST_IT	Each IT entity the TOE relies on for security functions (i.e., protection of the TOE and its resources and reliable time information) will be installed, configured, managed, maintained and provide the applicable security functions in a manner appropriate to the IT entity, and consistent with the security policy of the TOE and the relationship between them.

5. IT Security Requirements

This section defines the security functional and security assurance requirements for the TOE and associated IT environment components. Note that in addition to these requirements, Adaptive Server Anywhere also satisfies a minimum strength of function 'SOF-medium'. The only applicable (i.e., probabilistic or permutational) security functions are FIA_SOS.1, FIA_UAU.2, and FIA_UID.2 which are all levied on the TOE.

5.1 TOE Security Functional Requirements

The following table describes the SFRs that are candidates to be satisfied by Adaptive Server Anywhere.

Requirement Class	Requirement Component
FAU: Security audit	FAU_GEN.1: Audit data generation
	FAU_GEN.2: User identity association
	FAU_SAR.1: Audit review
	FAU_SAR.3: Selectable audit review
	FAU_SEL.1: Selective audit
	FAU_STG.1: Protected audit trail storage
	FAU_STG.3: Action in case of possible audit data loss
FDP: User data protection	FDP_ACC.1: Subset access control
	FDP_ACF.1: Security attribute based access control
	FDP_RIP.2: Full residual information protection
FIA: Identification and authentication	FIA_AFL.1: Authentication failure handling
	FIA_ATD.1: User attribute definition
	FIA_SOS.1: Verification of secrets
	FIA_UAU.2: User authentication before any action
	FIA_UID.2: User identification before any action
	FIA_USB.1: User-subject binding
FMT: Security management	FMT_MOF.1: Management of security functions behaviour
	FMT_MSA.1: Management of security attributes
	FMT_MSA.2: Secure security attributes
	FMT_MSA.3: Static attribute initialization
	FMT_MTD.1a: Management of TSF data
	FMT_MTD.1b: Management of TSF data
	FMT_MTD.1c: Management of TSF data
	FMT_REV.1a: Revocation
	FMT_REV.1b: Revocation
	FMT_SMF.1: Specification of Management Functions *(per International Interpretation #65)*
	FMT_SMR.1: Security roles
FPT: Protection of the TSF	FPT_RVM.1a: Non-bypassability of the TSP
	FPT_SEP.1a: TSF domain separation
FTA: TOE access	FTA_MCS.1: Basic limitation on multiple concurrent sessions
	FTA_TSE.1: TOE session establishment

Table 1 TOE Security Functional Components

5.1.1 Security audit (FAU)

5.1.1.1 Audit data generation (FAU_GEN.1)

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events: a) Start-up and shutdown of the audit functions; b) All auditable events for the ***[not specified]*** level of audit; and c) **[the auditable actions identified in the following table]**. *(per International Interpretation #202)*

Requirement Component	Auditable Action
FAU_GEN.1: Audit data generation	None
FAU_GEN.2: User identity association	None
FAU_SAR.1: Audit review	None
FAU_SAR.3: Selectable audit review	None
FAU_SEL.1: Selective audit	All modifications to the audit configuration that occur while the audit collection functions are operating.
FAU_STG.1: Protected audit trail storage	None
FAU_STG.3: Action in case of possible audit data loss	None
FDP_ACC.1: Subset access control	None
FDP_ACF.1: Security attribute based access control	Successful requests to perform an operation on an object covered by the SFP.
FDP_RIP.2: Full residual information protection	None
FIA_AFL.1: Authentication failure handling	None
FIA_ATD.1: User attribute definition	None
FIA_SOS.1: Verification of secrets	Rejection by the TSF of any tested secret.
FIA_UAU.2: User authentication before any action	Unsuccessful use of the authentication mechanism.
FIA_UID.2: User identification before any action	Unsuccessful use of the user identification mechanism, including the user identity provided.
FIA_USB.1: User-subject binding	None
FMT_MOF.1: Management of security functions behaviour	None
FMT_MSA.1: Management of security attributes	None
FMT_MSA.2: Secure security attributes	None
FMT_MSA.3: Static attribute initialization	None
FMT_MTD.1a: Management of TSF data	None
FMT_MTD.1b: Management of TSF data	None
FMT_MTD.1c: Management of TSF data	None
FMT_REV.1a: Revocation	None
FMT_REV.1b: Revocation	None
FMT_SMF.1: Specification of Management Functions	Use of the management functions.
FMT_SMR.1: Security roles	Modifications to the group of users that are part of a role.
FPT_RVM.1a: Non-bypassability of the TSP	None
FPT_SEP.1a: TSF domain separation	None
FTA_MCS.1: Basic limitation on multiple concurrent sessions	None
FTA_TSE.1: TOE session establishment	None

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, **[no additional information]**

5.1.1.2 User identity association (FAU_GEN.2)

FAU_GEN.2.1 The TSF shall be able to associate each auditable event with the identity of the user that caused the event.

5.1.1.3 Audit review (FAU_SAR.1)

FAU_SAR.1.1 The TSF shall provide **[the authorized administrator]** with the capability to read **[all audit information]** from the audit records.

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

5.1.1.4 Selectable audit review (FAU_SAR.3)

FAU_SAR.3.1 The TSF shall provide the ability to perform **[*searches* and *sorting*]** of audit data based on **[user identities]**.

5.1.1.5 Selective audit (FAU_SEL.1)

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) **[*event type*]** b) **[no additional attributes]**.

5.1.1.6 Protected audit trail storage (FAU_STG.1)

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 **[*prevent*]** unauthorised modifications to the audit records in the audit trail. *(per International Interpretations #141 and #202)*

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

FAU_STG.3.1 The TSF shall take **[action to prevent additional auditable events]** if the audit trail exceeds **[its maximum capacity]**.

5.1.2 User data protection (FDP)

5.1.2.1 Subset access control (FDP_ACC.1)

FDP_ACC.1.1 The TSF shall enforce the **[Discretionary Access Control Policy]** on **[all database subjects; the following database objects: tables, views, stored procedures and user-defined functions; and, all operations on the identified database objects by database subjects]**.

5.1.2.2 Security attribute based access control (FDP_ACF.1)

FDP_ACF.1.1 The TSF shall enforce the **[Discretionary Access Control Policy]** to objects based on the following: **[database subject attributes: user identity, group memberships and authorities; and, database object attributes: owner and access control lists (ACLs)]**. *(per International Interpretation #103)*

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **[a) if the user identity is equal to the object owner, the requested access is allowed; or b) if the ACL grants the requesting user identity the requested access, the requested access is allowed; or c) if the user identity is a member of a group and the ACL grants the group the requested access, the requested access is allowed; or d) otherwise access is denied, unless access is explicitly authorized in accordance with the rules specified in FDP_ACF.1.3.]**.

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **[a) if the database subject has DBA authority, the requested access is allowed.]**.

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the **[there are no explicit access denial rules]**.

5.1.2.3 Full residual information protection (FDP_RIP.2)

FDP_RIP.2.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the ***[allocation of the resource to]*** all objects.

5.1.3 Identification and authentication (FIA)

5.1.3.1 Authentication failure handling (FIA_AFL.1)

FIA_AFL.1.1 The TSF shall detect when ***[an administrator configurable positive integer within [0 – 32767[2]]]*** unsuccessful authentication attempts occur related to **[user identification]**. *(per International Interpretation #111)*

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall **[prevent subsequent authentication of the identified user**].

5.1.3.2 User attribute definition (FIA_ATD.1)

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users: **[user identity, authentication data, group memberships and authorities]**.

5.1.3.3 Verification of secrets (FIA_SOS.1)

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet **[the following: a) for each attempt to use the authentication mechanisms, the probability that a random attempt will succeed is less than one in 5,000,000,000,000,000; and b) any feedback given during each attempt to use the authentication mechanism will reduce the probability of the above metric by only one.]**.

5.1.3.4 User authentication before any action (FIA_UAU.2)

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

5.1.3.5 User identification before any action (FIA_UID.2)

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

5.1.3.6 User-subject binding (FIA_USB.1)

FIA_USB.1.1 The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: **[user identity, group memberships, and authorities]**. *(per International Interpretation #137)*

FIA_USB.1.2 The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: **[subject security attributes are derived from TSF data maintained for each defined user after a successful connection with the defined user identity]**. *(per International Interpretation #137)*

FIA_USB.1.3 The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: **[subject security attributes cannot change after initial assignment, other than authorities and group memberships, and then only through use of the GRANT and REVOKE statements by authorized users]**. *(per International Interpretation #137)*

[2] In the context of this requirement, a value of '0' indicates that the account is locked (i.e., no authentication attempts will succeed).

5.1.4 Security management (FMT)

5.1.4.1 Management of security functions behaviour (FMT_MOF.1)

FMT_MOF.1.1 The TSF shall restrict the ability to **[*disable* and *enable*]** the functions **[related to the specification of events to be audited]** to **[authorized administrators]**.

5.1.4.2 Management of security attributes (FMT_MSA.1)

FMT_MSA.1.1 The TSF shall enforce the **[Discretionary Access Control Policy]** to restrict the ability to **[*[manage]*]** the security attributes **[of database subjects]** to **[authorized administrators]**.

5.1.4.3 Secure security attributes (FMT_MSA.2)

FMT_MSA.2.1 The TSF shall ensure that only secure values are accepted for security attributes.

5.1.4.4 Static attribute initialization (FMT_MSA.3)

FMT_MSA.3.1 The TSF shall enforce the **[Discretionary Access Control Policy]** to provide **[*restrictive*]** default values for security attributes that are used to enforce the SFP. *(per International Interpretations #201 and #202)*

FMT_MSA.3.2 The TSF shall allow the **[no user role]** to specify alternative initial values to override the default values when an object or information is created.

5.1.4.5 Management of TSF data (FMT_MTD.1a)

FMT_MTD.1a.1 The TSF shall restrict the ability to **[*include* or *exclude*]** the **[audited events]** to **[authorized administrators]**.

5.1.4.6 Management of TSF data (FMT_MTD.1b)

FMT_MTD.1b.1 The TSF shall restrict the ability to **[*query* and *clear*]** the **[audit records]** to **[authorized administrators]**.

5.1.4.7 Management of TSF data (FMT_MTD.1c)

FMT_MTD.1c.1 The TSF shall restrict the ability to **[*set* and *reset*]** the **[user authentication data]** to **[authorized administrators and the user associated with the authentication data]**.

5.1.4.8 Revocation (FMT_REV.1a)

FMT_REV.1a.1 The TSF shall restrict the ability to revoke security attributes associated with the **[*subjects*]** within the TSC to **[authorized administrators]**. *(per International Interpretation #201)*

FMT_REV.1a.2 The TSF shall enforce the rules **[: the enforcement of subject attribute changes shall take immediately on completion of the revocation operation]]**.

5.1.4.9 Revocation (FMT_REV.1b)

FMT_REV.1b.1 The TSF shall restrict the ability to revoke security attributes associated with the **[*objects]*** within the TSC to **[authorized users (only for database objects they own or database objects for which they have been granted subject access privileges allowing them to revoke security attributes)]**. *(per International Interpretation #201)*

FMT_REV.1b.2 The TSF shall enforce the rules **[: the enforcement of object attribute changes shall take effect before the next access attempt related to that object]**.

5.1.4.10 Specification of Management Functions *(per International Interpretation #65)* (FMT_SMF.1)

FMT_SMF.1.1 The TSF shall be capable of performing the following security management functions: **[starting and stopping the audit function, selection of the audited events, review of audit data, and**

management of database subjects and authentication data]. *(per International Interpretation #65)*

5.1.4.11 Security roles (FMT_SMR.1)

FMT_SMR.1.1 The TSF shall maintain the roles **[authorized administrators and users]**.
FMT_SMR.1.2 The TSF shall be able to associate users with roles.

5.1.5 Protection of the TSF (FPT)

5.1.5.1 Non-bypassability of the TSP (FPT_RVM.1a)

FPT_RVM.1a.1 The TSF shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

5.1.5.2 TSF domain separation (FPT_SEP.1a)

FPT_SEP.1a.1 The TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.
FPT_SEP.1a.2 The TSF shall enforce separation between the security domains of subjects in the TSC.

5.1.6 TOE access (FTA)

5.1.6.1 Basic limitation on multiple concurrent sessions (FTA_MCS_EXP.1)

FTA_MCS_EXP.1.1 The TSF shall be able to restrict the maximum number of concurrent sessions that belong to the same user.
FTA_MCS_EXP.1.2 The TSF shall enforce, by default, no limit to the number of sessions per user.

5.1.6.2 TOE session establishment (FTA_TSE.1)

FTA_TSE.1.1 The TSF shall be able to deny session establishment based on **[user identity and time]**.

5.2 IT Environment Security Functional Requirements

The following table describes the SFRs that are candidates to be satisfied by the IT environment of Adaptive Server Anywhere.

Requirement Class	Requirement Component
FPT: Protection of the TSF	FPT_RVM.1b: Non-bypassability of the TSP
	FPT_SEP.1b: TSF domain separation
	FPT_STM.1: Reliable time stamps

Table 2 IT Environment Security Functional Components

5.2.1 Protection of the TSF (FPT)

5.2.1.1 Non-bypassability of the TSP (FPT_RVM.1b)

FPT_RVM.1b.1 The TSF shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

5.2.1.2 TSF domain separation (FPT_SEP.1b)

FPT_SEP.1b.1 The TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.
FPT_SEP.1b.2 The TSF shall enforce separation between the security domains of subjects in the TSC.

5.2.1.3 Reliable time stamps (FPT_STM.1)

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

5.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are the EAL 3 augmented with ALC_FLR.2 components as specified in Part 3 of the Common Criteria. No operations are applied to the assurance components.

Requirement Class	Requirement Component
ACM: Configuration management	ACM_CAP.3: Authorisation controls
	ACM_SCP.1: TOE CM coverage
ADO: Delivery and operation	ADO_DEL.1: Delivery procedures
	ADO_IGS.1: Installation, generation, and start-up procedures
ADV: Development	ADV_FSP.1: Informal functional specification
	ADV_HLD.2: Security enforcing high-level design
	ADV_RCR.1: Informal correspondence demonstration
AGD: Guidance documents	AGD_ADM.1: Administrator guidance
	AGD_USR.1: User guidance
ALC: Life cycle support	ALC_DVS.1: Identification of security measures
	ALC_FLR.2: Flaw reporting procedures
ATE: Tests	ATE_COV.2: Analysis of coverage
	ATE_DPT.1: Testing: high-level design
	ATE_FUN.1: Functional testing
	ATE_IND.2: Independent testing - sample
AVA: Vulnerability assessment	AVA_MSU.1: Examination of guidance
	AVA_SOF.1: Strength of TOE security function evaluation
	AVA_VLA.1: Developer vulnerability analysis

Table 3 EAL 3 augmented with ALC_FLR.2 Assurance Components

5.3.1 Configuration management (ACM)

5.3.1.1 Authorisation controls (ACM_CAP.3)

ACM_CAP.3.1d The developer shall provide a reference for the TOE.
ACM_CAP.3.2d The developer shall use a CM system.
ACM_CAP.3.3d The developer shall provide CM documentation.
ACM_CAP.3.1c The reference for the TOE shall be unique to each version of the TOE.
ACM_CAP.3.2c The TOE shall be labelled with its reference.
ACM_CAP.3.3c The CM documentation shall include a configuration list and a CM plan.
ACM_CAP.3.4c The configuration list shall describe the configuration items that comprise the TOE.
ACM_CAP.3.5c The CM documentation shall describe the method used to uniquely identify the configuration items.

ACM_CAP.3.6c The CM system shall uniquely identify all configuration items.
ACM_CAP.3.7c The CM plan shall describe how the CM system is used.
ACM_CAP.3.8c The evidence shall demonstrate that the CM system is operating in accordance with the CM plan.
ACM_CAP.3.9c The CM documentation shall provide evidence that all configuration items have been and are being effectively maintained under the CM system.
ACM_CAP.3.10c The CM system shall provide measures such that only authorised changes are made to the configuration items.
ACM_CAP.3.11c The configuration list shall uniquely identify all configuration items that comprise the TOE. *(per International Interpretation #3)*
ACM_CAP.3.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.1.2 TOE CM coverage (ACM_SCP.1)

ACM_SCP.1.1d The developer shall provide a list of configuration items for the TOE. *(per International Interpretation #4)*
ACM_SCP.1.1c The list of configuration items shall include the following: implementation representation and the evaluation evidence required by the assurance components in the ST. *(per International Interpretation #4)*
ACM_SCP.1.2c *(this element has been deleted per International Interpretation #4)*
ACM_SCP.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.2 Delivery and operation (ADO)

5.3.2.1 Delivery procedures (ADO_DEL.1)

ADO_DEL.1.1d The developer shall document procedures for delivery of the TOE or parts of it to the user.
ADO_DEL.1.2d The developer shall use the delivery procedures.
ADO_DEL.1.1c The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.
ADO_DEL.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.2.2 Installation, generation, and start-up procedures (ADO_IGS.1)

ADO_IGS.1.1d The developer shall document procedures necessary for the secure installation, generation, and start-up of the TOE.
ADO_IGS.1.1c The installation, generation and start-up documentation shall describe all the steps necessary for secure installation, generation and start-up of the TOE. *(per International Interpretation #51 (rev 1))*
ADO_IGS.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADO_IGS.1.2e The evaluator shall determine that the installation, generation, and start-up procedures result in a secure configuration.

5.3.3 Development (ADV)

5.3.3.1 Informal functional specification (ADV_FSP.1)

ADV_FSP.1.1d The developer shall provide a functional specification.
ADV_FSP.1.1c The functional specification shall describe the TSF and its external interfaces using an informal style.
ADV_FSP.1.2c The functional specification shall be internally consistent.
ADV_FSP.1.3c The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing details of effects, exceptions and error messages, as appropriate.
ADV_FSP.1.4c The functional specification shall completely represent the TSF.

ADV_FSP.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_FSP.1.2e The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

5.3.3.2 Security enforcing high-level design (ADV_HLD.2)

ADV_HLD.2.1d The developer shall provide the high-level design of the TSF.
ADV_HLD.2.1c The presentation of the high-level design shall be informal.
ADV_HLD.2.2c The high-level design shall be internally consistent.
ADV_HLD.2.3c The high-level design shall describe the structure of the TSF in terms of subsystems.
ADV_HLD.2.4c The high-level design shall describe the security functionality provided by each subsystem of the TSF.
ADV_HLD.2.5c The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.
ADV_HLD.2.6c The high-level design shall identify all interfaces to the subsystems of the TSF.
ADV_HLD.2.7c The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.
ADV_HLD.2.8c The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing details of effects, exceptions and error messages, as appropriate.
ADV_HLD.2.9c The high-level design shall describe the separation of the TOE into TSP-enforcing and other subsystems.
ADV_HLD.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_HLD.2.2e The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

5.3.3.3 Informal correspondence demonstration (ADV_RCR.1)

ADV_RCR.1.1d The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided.
ADV_RCR.1.1c For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.
ADV_RCR.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.4 Guidance documents (AGD)

5.3.4.1 Administrator guidance (AGD_ADM.1)

AGD_ADM.1.1d The developer shall provide administrator guidance addressed to system administrative personnel.
AGD_ADM.1.1c The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.
AGD_ADM.1.2c The administrator guidance shall describe how to administer the TOE in a secure manner.
AGD_ADM.1.3c The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.
AGD_ADM.1.4c The administrator guidance shall describe all assumptions regarding user behaviour that are relevant to secure operation of the TOE.
AGD_ADM.1.5c The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.
AGD_ADM.1.6c The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.
AGD_ADM.1.7c The administrator guidance shall be consistent with all other documentation supplied for evaluation.

AGD_ADM.1.8c The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.

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

5.3.4.2 User guidance (AGD_USR.1)

AGD_USR.1.1d The developer shall provide user guidance.

AGD_USR.1.1c The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.

AGD_USR.1.2c The user guidance shall describe the use of user-accessible security functions provided by the TOE.

AGD_USR.1.3c The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.

AGD_USR.1.4c The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behaviour found in the statement of TOE security environment.

AGD_USR.1.5c The user guidance shall be consistent with all other documentation supplied for evaluation.

AGD_USR.1.6c The user guidance shall describe all security requirements for the IT environment that are relevant to the user.

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

5.3.5 Life cycle support (ALC)

5.3.5.1 Identification of security measures (ALC_DVS.1)

ALC_DVS.1.1d The developer shall produce development security documentation.

ALC_DVS.1.1c The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

ALC_DVS.1.2c The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.

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

ALC_DVS.1.2e The evaluator shall confirm that the security measures are being applied.

5.3.5.2 Flaw reporting procedures (ALC_FLR.2)

ALC_FLR.2.1d The developer shall provide flaw remediation procedures addressed to TOE developers. *(per International Interpretation #94)*

ALC_FLR.2.2d The developer shall establish a procedure for accepting and acting upon all reports of security flaws and requests for corrections to those flaws. *(per International Interpretation #62)*

ALC_FLR.2.3d The developer shall provide flaw remediation guidance addressed to TOE users. *(per International Interpretation #94)*

ALC_FLR.2.1c The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

ALC_FLR.2.2c The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC_FLR.2.3c The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC_FLR.2.4c The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.

ALC_FLR.2.5c The flaw remediation procedures documentation shall describe a means by which the developer receives from TOE users reports and enquiries of suspected security flaws in the TOE. *(per International Interpretation #94)*

ALC_FLR.2.6c The procedures for processing reported security flaws shall ensure that any reported flaws are corrected and the correction issued to TOE users. *(per International Interpretation #94)*

ALC_FLR.2.7c The procedures for processing reported security flaws shall provide safeguards that any corrections to these security flaws do not introduce any new flaws. *(per International Interpretation #94*

ALC_FLR.2.8c The flaw remediation guidance shall describe a means by which TOE users report to the developer any suspected security flaws in the TOE. *(per International Interpretation #94)*

ALC_FLR.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.6 Tests (ATE)

5.3.6.1 Analysis of coverage (ATE_COV.2)

ATE_COV.2.1d The developer shall provide an analysis of the test coverage.

ATE_COV.2.1c The analysis of the test coverage shall demonstrate the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.

ATE_COV.2.2c The analysis of the test coverage shall demonstrate that the correspondence between the TSF as described in the functional specification and the tests identified in the test documentation is complete.

ATE_COV.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.6.2 Testing: high-level design (ATE_DPT.1)

ATE_DPT.1.1d The developer shall provide the analysis of the depth of testing.

ATE_DPT.1.1c The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design.

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

5.3.6.3 Functional testing (ATE_FUN.1)

ATE_FUN.1.1d The developer shall test the TSF and document the results.

ATE_FUN.1.2d The developer shall provide test documentation.

ATE_FUN.1.1c The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.

ATE_FUN.1.2c The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.

ATE_FUN.1.3c The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.4c The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.5c The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.

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

5.3.6.4 Independent testing - sample (ATE_IND.2)

ATE_IND.2.1d The developer shall provide the TOE for testing.

ATE_IND.2.1c The TOE shall be suitable for testing.

ATE_IND.2.2c The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

ATE_IND.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2.2e The evaluator shall test a subset of the TSF as appropriate to confirm that the TOE operates as specified.

ATE_IND.2.3e The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

5.3.7 Vulnerability assessment (AVA)

5.3.7.1 Examination of guidance (AVA_MSU.1)

AVA_MSU.1.1d The developer shall provide guidance documentation.

AVA_MSU.1.1c The guidance documentation shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AVA_MSU.1.2c The guidance documentation shall be complete, clear, consistent and reasonable.

AVA_MSU.1.3c The guidance documentation shall list all assumptions about the intended environment.

AVA_MSU.1.4c The guidance documentation shall list all requirements for external security measures (including external procedural, physical and personnel controls).

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

AVA_MSU.1.2e The evaluator shall repeat all configuration and installation procedures to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.

AVA_MSU.1.3e The evaluator shall determine that the use of the guidance documentation allows all insecure states to be detected.

5.3.7.2 Strength of TOE security function evaluation (AVA_SOF.1)

AVA_SOF.1.1d The developer shall perform a strength of TOE security function analysis for each mechanism identified in the ST as having a strength of TOE security function claim.

AVA_SOF.1.1c For each mechanism with a strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the minimum strength level defined in the PP/ST.

AVA_SOF.1.2c For each mechanism with a specific strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the specific strength of function metric defined in the PP/ST.

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

AVA_SOF.1.2e The evaluator shall confirm that the strength claims are correct.

5.3.7.3 Developer vulnerability analysis (AVA_VLA.1)

AVA_VLA.1.1d The developer shall perform a vulnerability analysis. *(per International Interpretation #51 (rev 1))*

AVA_VLA.1.2d The developer shall provide vulnerability analysis documentation. *(per International Interpretation #51 (rev 1))*

AVA_VLA.1.1c The vulnerability analysis documentation shall describe the analysis of the TOE deliverables performed to search for obvious ways in which a user can violate the TSP. *(per International Interpretation #51 (rev 1))*

AVA_VLA.1.2c The vulnerability analysis documentation shall describe the disposition of obvious vulnerabilities. *(per International Interpretation #51 (rev 1))*

AVA_VLA.1.3c The vulnerability analysis documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE. *(per International Interpretation #51 (rev 1))*

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

AVA_VLA.1.2e The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure obvious vulnerabilities have been addressed.

6. TOE Summary Specification

This chapter describes the security functions and associated assurance measures.

6.1 TOE Security Functions

6.1.1 Security audit

The ASA Server provides its own audit mechanism. ASA maintains two audit logs. The primary audit log is the ASA transaction log file and is used to record auditable events that occur within the running ASA Server. The secondary audit log is the utility audit log file and is used by ASA utility programs to record auditable events regardless of whether the ASA Server is currently running.

Each audit record identifies the event type, responsible user (except for failed login attempts), data and time of the event, an indication of success or failure, and other information specific to each audit event.

Enabling or disabling the audit mechanism using the dbtran utility program requires an authority check[3] so that only authorized administrators can perform these functions.

The audit logs are protected with the rest of the ASA data files and can be accessed by the DBA using ASA utility programs. Furthermore, the audit logs can be imported back into a database and the SQL select command can be used to search and sort based on any attributes within the audit records, including user identities.

The general classes of auditable actions are listed below. All actions that require a role are auditable, such as those that require System Administrator or System Security Officer (i.e., any authorized administrator). Unsuccessful attempts to perform a trusted operation by an untrusted subject also result in the generation of an audit record. The auditable actions include:

- Enabling and disabling of the audit mechanism
- Successful and failed attempts to perform functions requiring DBA authority (i.e., authorized administrator actions)
- All successful and failed login attempts
- All successful and failed DDL and DML statements
- All permission checks

ASA allows authorized administrators to enable and disable the audit function as a whole and also to configure audit levels to control which auditable events will be audited when audit is enabled.

When the available audit log space (i.e., available disk space) is exhausted, ASA will stop processing requests upon encountering a fatal error while trying to perform the I/O operation.

The Security audit function is designed to satisfy the following security functional requirements:

- FAU_GEN.1: The audit events as well as the audit record content enumerated above represent a superset of that required.
- FAU_GEN.2: A user identity is associated with each audit event that involves a user.
- FAU_SAR.1: The files containing the audit log are accessible by authorized administrators who can view them with a utility program or use the SQL select command after importing the audit records into a database.

[3] Dbtran requires the administrator to provide a valid user name and password when executing the dbtran command and access controls on public database options further restricts the ability to enable and disable audit to the authorized administrator.

- FAU_SAR.3: The audit logs can be searched and sorted based on user identity and other audit records contents using the SQL select command after importing them into a database.
- FAU_SEL.1: A system stored procedure can be used to select auditable events to be audited based on audit event type as well as individual users in the case of object access attempts.
- FAU_STG.1: The files storing the audit log are protected so that only authorized administrators have any access.
- FAU_STG.3: The TOE does not allow auditable functions to complete when there is no available audit storage space.

6.1.2 User data protection

The ASA Server implements a discretionary access control (DAC) policy for object access based on:

- user identities,
- group memberships,
- authorities,
- object owners, and
- Access Control Lists (ACLs).

The ASA Server objects directly subject to this policy are tables, views, stored procedures and user-defined functions. Other objects are always public (e.g., global variables and messages), always private (e.g., temporary tables, connection variables), or are part of one of the protected objects (e.g., triggers and defaults).

The Server stores the access permissions for the applicable objects in access control lists (ACLs) and provides the ability to grant and revoke ANSI Standard SQL permissions[4]. Note that ACLs are stored in system database tables and simply represent permissions of specific users and groups to specific objects. There is no notion of explicitly denying access, so while granting access adds access permissions to an ACL, revoking access removes access permissions from an ACL. The user identity, groups and authorities (i.e., Database Administrator (DBA) Authority) associated with the user identity, are used by the DAC mechanism to validate the user's permission to access the applicable objects.

Only an authorized administrator can create or delete a database. The authorized administrator can subsequently grant users the permission to create tables, views, and stored procedures, as well as other capabilities in the database. For tables and views, the grantable permissions are insert, select, update, delete, alter (tables only), and references (tables only). When an authorized administrator or user (with RESOURCE authority) creates a table, the user becomes the table owner and inherits the ability to perform any operation on that table.

If a user has DAC permission on a view, the user also has implied DAC permission on all objects upon which the view depends (via the view only) provided that the owner of the view also has the corresponding permissions on all of the objects upon which the view depends. Note that in order to creqate a view the user must have at least SELECT permission on the corresponding objects (e.g., tables).

For stored procedures and user-defined functions the only permission is execute. If a user has execute permission on a stored procedure, the user can access all objects referenced by the stored procedure provided the owner of the stored procedure or user-defined function has permissions on them.

[4] The ANSI Standard SQL permissions are:

Tables, Views	ALTER, DELETE, INSERT, REFERENCES, SELECT, UPDATE
Stored Procedures, User-defined Functions	EXECUTE

Permissions can be granted with or without the grant option. The grant option permits the user to subsequently grant the specific permission to other users.

Users and Groups

While user identities can be used in ACLs to assigned specific access permissions to specific users, ASA also supports a 'group' feature. A group is a special identity that is allowed to have members. Note that both users and groups can be members of groups and each user or group can be a member of multiple groups. Membership in a group can be granted by the authorized administrator or by the group user ID.

Groups provide a convenient way to grant and revoke permissions to more than one user in a single statement.

Ownership and DAC Permissions

While ASA has no concept of a database owner, all of the system stored procedures and system tables are owned by SYS. However, ASA does not allow any user to connect to SYS and as a result no user can directly update the system tables or change a system stored procedure.

When a user in the database creates an object, that user becomes the object owner. For example, if user Joe creates table1, then Joe is the table owner (TBO) of table1.

In general the owner of an object has all access permissions to the object regardless of explicitly granted or revoked access permissions. This access will persist as long as the applicable user remains the owner of the object.

Residual Information Protection

When a database is dropped, the associated files are deleted and new files are created in conjunction with the creation of a new database.

When a table or index is created, pages are allocated in the database. Although the data areas of these pages are not zeroed out before use, the page header information is updated whenever a new page is used for the object. The information in the database's allocation pages, the allocation map for the object, and the page headers, ensures that only data which has been written out may be accessed.

When a table is dropped, all rows in the system tables of the database in which the table resides which reference the table and its associated indexes are deleted. The allocation bitmaps for all extents allocated to the table and its indexes are zeroed out so that there is no access to those pages. Truncating a table has the effect of deleting all data rows for the table and deallocating the associated data pages and extents from the table. When rows are 'shrunk' they are rewritten in place and the other rows are moved around so as to leave no gaps. This effectively results in a truncated table which is handled as indicated above. When rows are 'expanded', new rows are written and the old rows are marked as deleted and the associated space is available for reallocation.

The ASA Server object reuse policy on memory segments is write-before-read.

The User data protection function is designed to satisfy the following security functional requirements:

- FDP_ACC.1: All database subjects are subject to the DAC policy for all available operations on tables, views, stored procedures, and user-defined functions
- FDP_ACF.1: Database objects have owners, ACLs and can define groups (the 'public' group is always defined) and these attributes are compared against user identities in order to determine whether the request operation should be allowed. Alternately, a user may have a role that explicitly grants the requested access regardless of the normal access check. If both of these checks fail, access will be refused.
- FDP_RIP.2: Objects are cleared when allocated, either by zeroing the data structures or by overwriting the data structures with their new contents prior to being accessible.

6.1.3 Identification and authentication

The ASA Server supports an identification and authentication mechanism in addition to any that might be provided by the underlying operating system. In order to access the ASA Server, a login account, including a login name and password, must be created for the user. User accounts can be established as either regular user accounts or trusted

user accounts via the assignment of authorities as described in relation to the Security Management function (below). Login name, password (hashed) and an internal Server identifier are stored and protected in a system table.

To login to the Server, the user provides the login name and password to the Server. The Server hashes the password and compares the resulting value to that stored in the system table. If either the login name or password is incorrect the login request will fail and no functions will be made available.

The administrator is provided guidance in the administrator guide to define restrictions (e.g., minimum password length) to ensure that the chance of guessing a password is sufficiently small (i.e., less than one in 5,000,000,000,000,000). Note that when a failure occurs, the TOE does not indicate whether the identity or password was incorrect. Note also that an authorized administrator can create a login event and user-defined stored procedure (to make a decision based on the queried number of failed connections) that can be used to effectively disable a user account after a prescribed number of authentication failures[5] (see Section 6.1.6, TOE Access) until an authorized administrator re-enables the user account. Note that this user-defined stored procedure is documented in the provided guidance and must be defined by the Administrator.

As a result of a successful login, a subject (i.e., a connection object) is created on behalf of the client and is represented by a unique ASA Server identifier.

In addition to the user's identifiers and password, any groups and authorities assigned to the user are also stored in the system tables. Note that groups are used to simplify access control management and authorities define special privileges available to the user (e.g., DBA - Database Administrator).

The Identification and authentication function is designed to satisfy the following security functional requirements:

- FIA_AFL.1: Login events in combination with user-defined stored procedures can be used to disable a user account after an authorized administrator defined number of authentication failures.
- FIA_ATD.1: The TOE defines user identities, authentication data, groups, and authorities within system tables.
- FIA_SOS.1: The administrator is provided guidance necessary to configure the authentication parameters (e.g., minimum password length) necessary to comply with this requirement.
- FIA_UAU.2: The TOE offers no TSF-mediated functions until the user is authenticated.
- FIA_UID.2: The TOE offers no TSF-mediated functions until the user is identified.
- FIA_USB.1: Each user is identified and authenticated when a connection is made to the TOE and a connection structure is allocated with the user attributes for the duration of the connection.

6.1.4 Security management

ASA maintains a number of special system tables, system stored procedures, and public options to control how it operates. All system tables and system stored procedures are owned by the special user designator SYS, while public options are owned by the special user designator PUBLIC. System tables provide no more than read access to other users. System stored procedures generally allow execute access, but they are designed to restrict their own functions by making operation decisions based on the invoking user. For example, the stored procedure used to change passwords ensures the user is either an authorized administrator or the user associated with the password to be changed before making the requested change. Similarly, each system stored procedure allowing management of a security functions will succeed only when invoked by an authorized administrator. The values of public options are readable by any user, but are only settable by users with DBA authority.

Among other things, the system tables, accessible using system stored procedures, are responsible to define user accounts including authentication data (i.e., passwords), group memberships, and authorities (i.e., Database Administrator – DBA[6] or Resource[7] authority) with user accounts. Audit data is stored and protected in separate

[5] Note that the failure count does not reset after a period of time or anything and hence is based on the total cumulative number of failures, until the count is reset by the Administrator.
[6] DBA authority offers full permissions on all objects inside the database (other than objects owned by SYS) and allows the user to grant other users the permission to create objects and execute commands within the database.

files associated with ASA (this data is accessible using ASA utility programs). Audit parameters are stored in public options. Since when a database is created all of the security related management data is restricted to authorized administrators (i.e., the DBA). Furthermore, as indicated above access to the security data via system stored procedures is controlled by checks implemented within those procedures.

System tables store all meta information about the database including all security information. These tables are accessible read-only. The information in these system tables is changed through the use of SQL statements for managing the database (Data Definition Language – DDL). These tables are not modifiable with Data Manipulation Statements – DML (Insert, Update, Delete). Furthermore, any changes made to system table data will be effective the next time the data within the table is accessed (e.g., a new user connection) and will also be applied immediately to any connected user.

Access to other database objects, such as databases contents (e.g., tables) is subject to the Discretionary Access Control Policy and any user with sufficient privilege, according to that policy, can manage the associated access attributes. Unlike the system table data changes, changes to access attributes of database objects will be used during the next attempt to access that object. Any database object that is created is initially accessible only by the creator who can subsequently change the access permissions at their discretion.

The Security management function is designed to satisfy the following security functional requirements:

- FMT_MOF.1: The ability to enable and disable the audit related functions is restricted to an authorized administrator through access controls on public database options.
- FMT_MSA.1: The ability to manage database subject attributes is restricted to an administrator.
- FMT_MSA.2: ASA rejects invalid passwords and as such prevents the introduction of 'insecure' data.
- FMT_MSA.3: By default every database object is created with the creator as the owner. Subsequently, access can be granted to other users, but there is no method to specify access other than the default during creation.
- FMT_MTD.1a: The ability to configure audited events is restricted to an authorized administrator through access restrictions enforced by a system stored procedure.
- FMT_MTD.1b: The ability to delete or review audited events is restricted to an authorized administrator by requiring the administrator to provide a valid user name and password when executing the dbtran command, or by requiring physical access to the transaction log file.
- FMT_MTD.1c: The ability to set and reset subject authentication data is restricted to an authorized user through access controls explicitly granted to that user.
- FMT_REV.1a: The ability to manage database subject attributes is restricted to an authorized administrator through discretionary access controls. This information is stored in system tables. These tables are used to determine subject attributes each time a user connects to the ASA Server. However, when a subject attribute is revoked, ASA ensures that the change is effective immediately.
- FMT_REV.1b: The ability to manage database object attributes is based upon the user's access to the database object. The owner of an object can always manipulate its attributes, as can the authorized administrator. Other users can do so only went the applicable privileges have been granted.
- FMT_SMF.1: Administrators are able to perform all management functions, including: start and stop the audit function, select the auditable events to be audited, review the audit data, and manage database subjects including authentication data using SQL statements, system stored procedures, and ASA utility programs.
- FMT_SMR.1: Each user account can be assigned zero or more system-defined authorities. Any user account that is assigned one or more system-defined authorities is considered an 'authorized administrator' and other user accounts are considered simply 'users'.

[7] Resource authority allows a user to create any kind of object within a database rather than requiring granting permissions on individual CREATE statements.

6.1.5 Protection of the TSF

ASA instantiates itself as a process within task constructs provided by the underlying operating system. It retains exclusive control of its process and separates and differentiates subjects (representing clients) using separate TDS and CmdSeq connection objects.

A connection object is created at the time that a client user logs in. A connection object is deleted when a client user disconnects or when ASA closes a connection. ASA closes connections when instructed by a DBA or when shutting down. When ASA receives a request over a connection, it assigns a task from its task pool to execute the request. If no task is available, the request is queued. Tasks are not permanently assigned to connections. The connection object is represented inside ASA by a connection structure. The connection structure contains a pointer to the associated user definition that contains the security attributes of the user. The connection structure also contains links to pointers to connection structures for the session, presentation, and transport layers.

In addition to protecting its own process, ASA protects its memory and files using features provided by the underlying operating system. Specifically, it ensures that the security properties (i.e., operating system access controls) of those objects do not allow access by other operating system processes. This serves to both protect ASA itself as well as to ensure that any attempts to access the database constructs realized by ASA must be made through ASA. Furthermore, ASA has been carefully designed to offer well-defined interfaces that ensure that access to protected resources is subject to the applicable ASA security policies.

The Protection of the TSF function is designed to satisfy the following security functional requirements:

- FPT_RVM.1a: The TOE uses protected media (disk and memory) to store the objects it instantiates to ensure that any access attempts must go through ASA where the appropriate access rules are enforced.
- FPT_SEP.1a: The TOE instantiates itself as a process which it protects from inappropriate access. The TOE separates clients based on individual protocol connections.

6.1.6 TOE access

ASA allows authorized administrators to define login events that can invoke user-defined stored procedures (created by the authorized administrator) that are activated each time a client connects to the ASA Server. The combination of login events and Administrator-defined stored procedures can effectively be used to deny access to the ASA Server based on criteria defined by the authorized administrator. Among the criteria that can be configured are user identities to reject, disallowed time periods, and the maximum number of current sessions the user has. When the login event causes a Adminsitrator-defined stored procedure to be activated, it checks the applicable attributes against the defined criteria and if any of the rules match, the connection is rejected.

The TOE access function is designed to satisfy the following security functional requirements:

- FTA_MCS_EXP.1: Login events in combination with user-defined stored procedures can be used to restrict the number of concurrent client sessions.
- FTA_TSE.1: Login events in combination with user-defined stored procedures can be used to restrict specific users or sessions based on the current time.

6.2 TOE Security Assurance Measures

6.2.1 Configuration management

The configuration management measures applied by Sybase ensure that configuration items are uniquely identified, and that documented procedures are used to control and track changes that are made to the TOE. Sybase ensures changes to the implementation representation are controlled. Sybase performs configuration management on the TOE implementation representation, design documentation, tests and test documentation, user and administrator guidance, delivery and operation documentation, life-cycle documentation, vulnerability analysis documentation, and configuration management documentation.

These activities are documented in:

- Sybase Adaptive Server Anywhere Configuration Management Plan, Revision 0.6, February 16, 2006

The Configuration management assurance measure satisfies the following EAL 3 augmented with ALC_FLR.2 assurance requirements:

- ACM_CAP.3
- ACM_SCP.1

6.2.2 Delivery and operation

Sybase provides delivery documentation and procedures to identify the TOE, secure the TOE during delivery, and provide necessary installation and generation instructions. Sybase's delivery procedures describe all applicable procedures to be used to prevent in appropriate access to the TOE. Sybase also provides documentation that describes the steps necessary to install Adaptive Server Anywhere in accordance with the evaluated configuration.

These activities are documented in:

- Sybase Adaptive Server Anywhere Delivery and Operation Procedures, Revision 0.1, May 26, 2004
- Supplement for Installing Adaptive Server for Common Criteria Configuration, Document ID: DC00080-01-1252-01, Last revised: April 2006 (http://www.ianywhere.com/developer/product_manuals/sqlanywhere/sqlanywhere_cc_configuration.pdf)

The Delivery and operation assurance measure satisfies the following EAL 3 augmented with ALC_FLR.2 assurance requirements:

- ADO_DEL.1
- ADO_IGS.1

6.2.3 Development

Sybase has numerous documents describing all facets of the design of the TOE. In particular, they have a functional specification that describes the accessible TOE interfaces; a high-level design that decomposes the TOE architecture into subsystems and describes each subsystem and their interfaces; and, correspondence documentation that explains how each of the design abstractions correspond from the TOE summary specification in the Security Target to the subsystems.

These activities are documented in:

- Adaptive Server Anywhere Architecture Summary, Revision 1.3, 09/21/2005
- Adaptive Server Anywhere Command Sequence Specification, Revision 0.1, 06/15/2005
- Adaptive Server Anywhere Security Functional Requirements, Revision 0.4, 05/06/2005
- SQL Correspondence, Rev 2.0, 09/26/2005
- TDS Correspondence, Rev 2.0, 06/23/2005
- TDS 5.0 Functional Specification, Version 3.4, 08/2005

The Development assurance measure satisfies the following EAL 3 augmented with ALC_FLR.2 assurance requirements:

- ADV_FSP.1
- ADV_HLD.2
- ADV_RCR.1

6.2.4 Guidance documents

Sybase provides administrator and user guidance on how to utilize the TOE security functions and warnings to administrators and users about actions that can compromise the security of the TOE.

These activities are documented in:

- Supplement for Installing Adaptive Server for Common Criteria Configuration, Document ID: DC00080-01-1252-01, Last revised: April 2006 (http://www.ianywhere.com/developer/product_manuals/sqlanywhere/sqlanywhere_cc_configuration.pdf)
- Sybase ASA SQL reference, January 2004 (9.0.1) and October 2004 (9.0.2)
- Sybase ASA error messages, January 2004 (9.0.1) and October 2004 (9.0.2)
- Sybase ASA database administration guide, January 2004 (9.0.1) and October 2004 (9.0.2)
- SQL Anywhere Studio Security Guide, January 2004 (9.0.1) and October 2004 (9.0.2)

The Guidance documents assurance measure satisfies the following EAL 3 augmented with ALC_FLR.2 assurance requirements:

- AGD_ADM.1
- AGD_USR.1

6.2.5 Life cycle support

Sybase ensures the adequacy of the procedures used during the development and maintenance of the TOE through its life-cycle. Sybase includes security controls on the development environment that are adequate to provide the confidentiality and integrity of the TOE design and implementation that is necessary to ensure the secure operation of the TOE. In addition, Sybase identifies and tracks reported flaws, ensuring that they are addressed and corrections and corrective measures are made available as applicable.

These activities are documented in:

- Sybase Adaptive Server Anywhere Life Cycle Document Revision 0.1, 28 September 2004
- Video Files (as documented in CC_Video_Script), Version 1.0, February 16, 2006

The Life cycle support assurance measure satisfies the following EAL 3 augmented with ALC_FLR.2 assurance requirements:

- ALC_DVS.1
- ALC_FLR.2

6.2.6 Tests

Sybase has a test plan that describes how each of the necessary security functions is tested, along with the expected test results. Sybase has documented each test as well as an analysis of test coverage and depth demonstrating that the security aspects of the design evident from the functional specification and high-level design are appropriately tested. Actual test results are created on a regular basis to demonstrate that the tests have been applied and that the TOE operates as designed.

These activities are documented in:

- Sybase ASA Security Function Test Documentation, Revision 0.8, December 2, 2005
- SQL Correspondence, 09/26/2005 (test coverage)
- Test Scripts as referenced by Security Function Test Documentation, 1/20/2006
- Test Results as referenced by test scripts, 2/10/2006

The Tests assurance measure satisfies the following EAL 3 augmented with ALC_FLR.2 assurance requirements:

- ATE_COV.2
- ATE_DPT.1
- ATE_FUN.1
- ATE_IND.2

6.2.7 Vulnerability assessment

The TOE administrator and user guidance documents describe the operation of Adaptive Server Anywhere and how to maintain a secure state. These guides also describe all necessary operating assumptions and security requirements outside the scope of control of the TOE. They have been developed to serve as complete, clear, consistent, and reasonable administrator and user references.

Sybase has conducted a strength of function analysis wherein all permutational or probabilistic security mechanisms have been identified and analyzed resulting in a demonstration that all of the relevant mechanisms fulfill the minimum strength of function claim, SOF-Medium.

Sybase performs regular vulnerability analyses of the entire TOE (including documentation) to identify obvious weaknesses that can be exploited in the TOE.

These activities are documented in:

- • Sybase ASA Vulnerability Analysis, Revision 1.2, January 10, 2006

The Vulnerability assessment assurance measure satisfies the following EAL 3 augmented with ALC_FLR.2 assurance requirements:

- AVA_MSU.1
- AVA_SOF.1
- AVA_VLA.1

7. Protection Profile Claims

There are no Protection Profile claims.

8. Rationale

This section provides the rationale for completeness and consistency of the Security Target. The rationale addresses the following areas:

- Security Objectives;
- Security Functional Requirements;
- Security Assurance Requirements;
- Strength of Functions;
- Security Functional Requirement Dependencies;
- Explicitly Stated Requirements;
- TOE Summary Specification; and
- PP Claims.

8.1 Security Objectives Rationale

This section shows that all secure usage assumptions, organizational security policies, and threats are completely covered by security objectives. In addition, each objective counters or addresses at least one assumption, organizational security policy, or threat.

8.1.1 Security Objectives Rationale for the TOE and Environment

This section provides evidence demonstrating the coverage of organizational policies and usage assumptions by the security objectives.

	P.ACCOUNTABILITY	P.AUTHORIZED_USERS	P.NEED_TO_KNOW	P.ROLES	T.ADMIN_ERROR	T.AUDIT_COMPROMISE	T.MASQUERADE	T.RESIDUAL_DATA	T.SYSACC	T.TSF_COMPROMISE	T.UNAUTH_ACCESS	T.UNDETECTED_ACTIONS	T.UNIDENTIFIED_ACTIONS	A.NO_EVIL	A.NO_GENERAL_PURPOSE	A.PHYSICAL	A.ROBUST_ENVIRONMENT
O.ACCESS		X	X						X		X						
O.ADMIN_ROLE				X													
O.AUDIT_GENERATION	X					X						X					
O.AUDIT_PROTECTION						X						X					
O.AUDIT_REVIEW	X												X				
O.DISCRETIONARY_ACCESS			X								X						
O.INTERNAL_TOE_DOMAINS											X						
O.MANAGE					X				X				X				
O.PROTECT			X								X						
O.RESIDUAL_INFORMATION								X									
O.TOE_PROTECTION										X							

O.USER_AUTHENTICATION							X		X							
O.USER_IDENTIFICATION	X		X				X		X							
OE.TIME	X											X				
OE.TOE_PROTECTION										X						
OE.ADMIN_GUIDANCE					X				X				X			
OE.CONFIG														X		
OE.INSTALL					X											
OE.NO_GENERAL_PURPOSE															X	
OE.PHYSICAL						X			X	X	X	X				X	
OE.ROBUST_ENVIRONMENT																	X
OE.SELF_PROTECTION						X					X						
OE.TRUST_IT																	X

Table 4 Environment to Objective Correspondence

8.1.1.1 P.ACCOUNTABILITY

The users of the TOE shall be held accountable for their actions within the TOE.

This Organizational Policy is satisfied by ensuring that:

- O.AUDIT_GENERATION: Enforcement of this policy requires all user actions be recorded.
- O.AUDIT_REVIEW: Enforcement of this policy requires all recorded actions must be available for review by the authorized administrator.
- O.USER_IDENTIFICATION: Enforcement of this policy requires all users to be uniquely identified.
- OE.TIME: Enforcement of this policy requires all recorded actions must have reliable timestamps.

8.1.1.2 P.AUTHORIZED_USERS

Access controls will ensure that only those users who have been authorized to access the protected information within the TOE will be able to do so.

This Organizational Policy is satisfied by ensuring that:

- O.ACCESS: The TOE will provide mechanisms to allow only authorized users to access the TOE, mainly Discretionary Access controls.

8.1.1.3 P.NEED_TO_KNOW

The TOE must limit the access to information in protected resources to those authorized users who have a need to know that information.

This Organizational Policy is satisfied by ensuring that:

- O.ACCESS: The authorized administrator will be able to change a user's security attributes when that user no longer needs to access certain information.
- O.DISCRETIONARY_ACCESS: Enforcement of this policy requires the resources to be protected according to the rules of the discretionary access control policy.
- O.PROTECT: Enforcement of this policy requires the protection of resources.
- O.USER_IDENTIFICATION: Enforcement of this policy requires access decision to be based on unique user identities.

8.1.1.4 P.ROLES

The TOE shall provide an authorized administrator role for secure administration of the TOE. This role shall be separate and distinct from other authorized users.

This Organizational Policy is satisfied by ensuring that:

- O.ADMIN_ROLE: The TOE has the objective of providing an authorized administrator role for secure administration. The TOE may provide other roles as well, but only the role of authorized administrator is required.

8.1.1.5 T.ADMIN_ERROR

An authorized administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.

This Threat is countered by ensuring that:

- O.MANAGE: Improper administration could result if the TOE does not provide the proper administration tools. There is always the possibility that the administrator will make an honest mistake. This threat should be mitigated as long as the TOE provides the necessary administrator support.
- OE.ADMIN_GUIDANCE: Improper administration could result if the authorized administrator is unknowledgeable. There is always the possibility that the administrator will make an honest mistake. This threat should be mitigated as long as the authorized administrator is provided with knowledge necessary to carry out administrative duties.
- OE.INSTALL: The authorized administrator is provided with necessary installation instructions from the developer that details how to securely install the TOE.

8.1.1.6 T.AUDIT_COMPROMISE

A process or user may cause audit data to be inappropriately accessed (viewed, modified or deleted), or prevent future records from being recorded, thus masking an attacker's actions.

This Threat is countered by ensuring that:

- O.AUDIT_GENERATION: The TOE will generate an audit log.
- O.AUDIT_PROTECTION: The TOE must also provide protection for its audit data.
- OE.PHYSICAL: The environment must address the possible compromise of audit data due to physical means.
- OE.SELF_PROTECTION: The IT environment must also protect itself and its assets.

8.1.1.7 T.MASQUERADE

An unauthorized user, process, or external IT entity may masquerade as an authorized entity to gain access to data or TOE resources.

This Threat is countered by ensuring that:

- O.USER_AUTHENTICATION: Unique user identification must be supported by the objective of requiring all users of the TOE to prove their claimed identity.
- O.USER_IDENTIFICATION: Addressing the threat of a process or user masquerading as a different process or user produces an objective of uniquely identifying each user.

8.1.1.8 T.RESIDUAL_DATA

A user or process may gain unauthorized access to data through reallocation of TOE resources from one user or process to another.

This Threat is countered by ensuring that:

- O.RESIDUAL_INFORMATION: When data is deleted from memory or storage (e.g., disk drive) it is often left intact and not truly erased. Subsequent users who have that same memory space allocated to their processes might be able to observe other users' data that is residual in that memory/storage. Addressing this threat yields the objective that prohibits users from accessing data that had been stored in system resources previously allocated to other users.

8.1.1.9 T.SYSACC

A malicious process or user may gain unauthorized access to the authorized administrator account, or that of other trusted personnel.

This Threat is countered by ensuring that:

- O.ACCESS: The threat of the wrong individual gaining unauthorized access to the authorized administrator's account logically is addressed by the TOE.
- O.MANAGE: The TOE will provide mechanisms for the authorized administrator to set the security attributes for users so they are not allowed admin access.
- O.USER_AUTHENTICATION: The threat of unauthorized access may be mitigated by requiring the authorized administrator to be authenticated.
- O.USER_IDENTIFICATION: The threat of unauthorized access may be mitigated by requiring the authorized administrator to be uniquely identified.
- OE.ADMIN_GUIDANCE: Authorized administrators will have to know to check this information at each login. The authorized administrator must also be aware that he/she must protect the authentication information that allows access to the authorized administrator account.
- OE.PHYSICAL: The threat of the wrong individual gaining unauthorized access to the authorized administrator's account is addressed by physical means when appropriate.

8.1.1.10 T.TSF_COMPROMISE

A malicious user or process may cause the TOE, configuration data, or sensitive user data to be inappropriately accessed (viewed, modified or deleted) allowing a breach in the TSF security policies

This Threat is countered by ensuring that:

- O.TOE_PROTECTION: The TSF is protected under the TOE objective for TOE protection.
- OE.TOE_PROTECTION: The IT environment and indirectly the TSF are protected under the environmental objective for TOE protection.
- OE.PHYSICAL: The IT environment will protect the TSF from a compromise through physical means.

8.1.1.11 T.UNAUTH_ACCESS

A user may gain unauthorized access (view, modify, delete) to user data.

This Threat is countered by ensuring that:

- O.ACCESS: The TOE must satisfy the objective of ensuring that only authorized users may gain access to the TOE and the resources it protects, and that users are not allowed to access protected data for which they are not authorized.
- O.DISCRETIONARY_ACCESS: Access to user data is controlled by a discretionary policy.
- O.INTERNAL_TOE_DOMAINS: The TOE maintains internal domains to keep data and processes of concurrent users separate, so users cannot observe or interfere with other users' data or queries.
- O.PROTECT: Addressing the threat of other unauthorized access results in the objective of protecting the user data.
- OE.PHYSICAL: The threat of unauthorized physical access is addressed by the environment.
- OE.SELF_PROTECTION: The threat of unauthorized physical access is addressed by the environment.

8.1.1.12 T.UNDETECTED_ACTIONS

Failure of the IT operating system to detect and record unauthorized actions may occur.

This Threat is countered by ensuring that:

- O.AUDIT_GENERATION: Non-physical actions are detected and a record is made.
- O.AUDIT_PROTECTION: To prevent removing evidence of unauthorized actions, the audit records need to be protected from unauthorized modification.
- OE.TIME: All audit records include reliable timestamps.

- OE.PHYSICAL: The threat of undetected physical manipulation of the TOE is addressed by the physical protection in the environment.

8.1.1.13 T.UNIDENTIFIED_ACTIONS

Failure of the authorized administrator to identify and act upon unauthorized actions may occur.

This Threat is countered by ensuring that:
- O.AUDIT_REVIEW: The TOE provides the tools to effectively review audit records.
- O.MANAGE: The TOE provides necessary access to the audit trail.
- OE.ADMIN_GUIDANCE: The guidance provides the information necessary to manage audit data.

8.1.1.14 A.NO_EVIL

Authorized administrators are non-hostile, appropriately trained and follow all administrator guidance.

This Assumption is satisfied by ensuring that:
- OE.CONFIG: Authorized administrators are trained and trusted to properly configure the IT environment so it enforces its security policies.

8.1.1.15 A.NO_GENERAL_PURPOSE

There are no general-purpose computing capabilities (e.g., compilers or user applications) available on DBMS servers, other than those services necessary for the operation, administration and support of the DBMS.

This Assumption is satisfied by ensuring that:
- OE.NO_GENERAL_PURPOSE: The DBMS server must not include any general-purpose commuting or storage capabilities. This will protect the TSF data from malicious processes.

8.1.1.16 A.PHYSICAL

It is assumed that appropriate physical security is provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.

This Assumption is satisfied by ensuring that:
- OE.PHYSICAL: The TOE, the TSF data, and protected user data is assumed to be protected from physical attack (e.g., theft, modification, destruction, or eavesdropping). Physical attack could include unauthorized intruders into the TOE environment, but it does not include physical destructive actions that might be taken by an individual that is authorized to access the TOE environment.

8.1.1.17 A.ROBUST_ENVIRONMENT

It is assumed that the IT environment protects the TOE (and its resources) and provides time stamps with at least the same degree of assurance as that claimed by the TOE.

This Assumption is satisfied by ensuring that:
- OE.ROBUST_ENVIRONMENT: The TOE shall only be installed in an IT environment that is at least as robust (primarily a measure of assurance) as the TOE. The TOE is basic robustness, therefore, the operating system in the environment the TOE depends on for enforcement of its security objectives are also assumed to be basic robustness.
- OE.TRUST_IT: The IT entities in the environment are correctly installed, configured, managed, maintained and provide the applicable security functions.

8.2 Security Requirements Rationale

This section provides evidence supporting the internal consistency and completeness of the components (requirements) in the Security Target. Note that **Table 5** indicates the requirements that effectively satisfy the individual objectives. .

8.2.1 Security Functional Requirements Rationale

All Security Functional Requirements (SFR) identified in this Security Target are fully addressed in this section and each SFR is mapped to the objective for which it is intended to satisfy.

	O.ACCESS	O.ADMIN_ROLE	O.AUDIT_GENERATION	O.AUDIT_PROTECTION	O.AUDIT_REVIEW	O.DISCRETIONARY_ACCESS	O.INTERNAL_TOE_DOMAINS	O.MANAGE	O.PROTECT	O.RESIDUAL_INFORMATION	O.TOE_PROTECTION	O.USER_AUTHENTICATION	O.USER_IDENTIFICATION	OE.TIME	OE.TOE_PROTECTION
FAU_GEN.1			X												
FAU_GEN.2			X												
FAU_SAR.1					X										
FAU_SAR.3					X										
FAU_SEL.1			X												
FAU_STG.1				X											
FAU_STG.3				X											
FDP_ACC.1	X					X			X						
FDP_ACF.1	X					X			X						
FDP_RIP.2									X	X					
FIA_AFL.1												X			
FIA_ATD.1													X		
FIA_SOS.1												X			
FIA_UAU.2												X			
FIA_UID.2													X		
FIA_USB.1			X			X							X		
FMT_MOF.1			X					X							
FMT_MSA.1						X		X							
FMT_MSA.2								X							
FMT_MSA.3						X									
FMT_MTD.1a								X							
FMT_MTD.1b				X				X							
FMT_MTD.1c								X				X			
FMT_REV.1a	X														
FMT_REV.1b									X						
FMT_SMF.1			X			X		X							
FMT_SMR.1		X													
FPT_RVM.1a							X				X				
FPT_SEP.1a							X				X				
FTA_MCS_EXP.1	X														

FTA_TSE.1	X												
FPT_RVM.1b													X
FPT_SEP.1b													X
FPT_STM.1			X									X	

Table 5 Objective to Requirement Correspondence

8.2.1.1 O.ACCESS

The TOE will ensure that users gain only authorized access to it and to the resources that it controls.

This TOE Security Objective is satisfied by ensuring that:

- FDP_ACC.1: The Discretionary Access Control policy applies to all operations between subjects and objects (tables, views, stored procedures and user-defined functions) controlled by the TOE.
- FDP_ACF.1: The subjects and objects under the discretionary access control policy will have certain rules that apply to all accesses between them. The rules are based on subject identities and access control lists associated with database objects that either grant or deny potential request access.
- FMT_REV.1a: Security attributes associated with subjects and objects are the basis for access control. Revocation of these security attributes would modify the access control policy. The authorized administrator should have control over security attributes associated with users (such as user authentication data), being the only role that can revoke them.
- FTA_MCS_EXP.1: The TOE must keep track of what user sessions are currently established and running, associating each established session with a uniquely identified user. The TOE must provide the ability to limit the number of concurrent user sessions.
- FTA_TSE.1: The TOE can restrict access to itself (i.e., session establishment) based on specific user identities and the time.

8.2.1.2 O.ADMIN_ROLE

The TOE will provide authorized administrator roles to isolate administrative actions.

This TOE Security Objective is satisfied by ensuring that:

- FMT_SMR.1: The TOE will establish, at least, an authorized administrator role. The authorized administrator will be given privileges to perform certain tasks that other users will not be able to perform. These privileges include, but are not limited to, access to audit information and security functions.

8.2.1.3 O.AUDIT_GENERATION

The TOE will provide the capability to detect and create records of security relevant events associated with users.

This TOE Security Objective is satisfied by ensuring that:

- FAU_GEN.1: This objective is satisfied in part by the requirement that the TOE generate audit records according to the minimum level of auditing, as defined by the Common Criteria.
- FAU_GEN.2: Each audit record written must be descriptive of the event that caused a record to be generated, and must be associated with the unique identity of the user that caused the event.
- FAU_SEL.1: The TOE enables the authorized administrator to pre-select events to include in the audit log.
- FIA_USB.1: All subjects that act on behalf of users must have a binding that associates the subjects with a user. This is necessary to be able to associate audit records with user identities.
- FMT_MOF.1: The TOE ensures that the authorized administrator role is the only role authorized to manipulate the behavior of the audit generation mechanism.
- FMT_SMF.1: The TOE ensures that the authorized administrator role is able to manipulate the behavior of the audit generation mechanism.

- FPT_STM.1: Reliable time stamps are assumed to be provided by the IT environment.

8.2.1.4 O.AUDIT_PROTECTION

The TOE will provide the capability to protect audit information.

This TOE Security Objective is satisfied by ensuring that:

- FAU_STG.1: The TOE prevents unauthorized deletion or modification of audit records.
- FAU_STG.3: The TOE provides site-configurable options to prevent loss of audit data in the event the audit storage space is exhausted.
- FMT_MTD.1b: Only the authorized administrator has the ability to query or clear audit records.

8.2.1.5 O.AUDIT_REVIEW

The TOE will provide the capability to selectively view audit information.

This TOE Security Objective is satisfied by ensuring that:

- FAU_SAR.1: In order for the authorized administrator to review the audit logs they must be accessible in a suitable form for the authorized administrator to read, which means the authorized administrator should have the appropriate functions needed to interpret the data.
- FAU_SAR.3: The authorized administrator must be able to search and sort on the audit data based on date, time, type of event, event status (success or failure), or user identity. This will allow the authorized administrator to examine specific events more efficiently.

8.2.1.6 O.DISCRETIONARY_ACCESS

The TOE will control access to resources based upon the identity of users or groups of users.

This TOE Security Objective is satisfied by ensuring that:

- FDP_ACC.1: The Discretionary Access Control policy applies to all operations between subjects and objects controlled by the TOE.
- FDP_ACF.1: The subjects and objects under the discretionary access control policy will have certain rules that apply to all accesses between them. The rules are based on subject identities and access control lists associated with database objects that either grant or deny potential request access.
- FIA_USB.1: All subjects that act on behalf of users must have a binding that associates the subjects with a user uniquely.
- FMT_MSA.1: Only authorized administrators may manipulate the security attributes of database users.
- FMT_MSA.3: Default access control attributes are restrictive to prevent accidental (non-discretionary) disclosure of information that should be protected.
- FMT_SMF.1: Authorized administrators must be able to manipulate the security attributes of database users.

8.2.1.7 O.INTERNAL_TOE_DOMAINS

The TSF will maintain internal domains for separation of data and queries belonging to concurrent users.

This TOE Security Objective is satisfied by ensuring that:

- FPT_RVM.1a: The mechanisms providing self-protection are always invoked and not able to be bypassed.
- FPT_SEP.1a: The TSF enforces separation between the security domains within its scope of control.

8.2.1.8 O.MANAGE

The TOE will provide all the functions and facilities necessary to support the authorized administrators in their management of the security of the TOE.

This TOE Security Objective is satisfied by ensuring that:

- FMT_MOF.1: Only the authorized administrator will be able to enable or disable functions of the audit log. This will prevent a malicious user from turning off the audit log while he/she performs a malicious act, then turning it back on when he/she is done.
- FMT_MSA.1: Only authorized administrators may manipulate the security attributes of database users.
- FMT_MSA.2: The TOE rejects invalid and insecure data to help ensure the effectiveness of the security functions.
- FMT_MTD.1a: Only authorized administrators are able to manage the inclusion/exclusion of specific events to be audited.
- FMT_MTD.1b: Only authorized administrators are authorized to query or clear the audit log.
- FMT_MTD.1c: Only authorized administrators are authorized to set or reset user authentication data.
- FMT_SMF.1: The authorized administrator will be able to enable or disable functions of the audit log, select audited events, review audit records, and manage database subjects and authentication data.

8.2.1.9 O.PROTECT

The TOE will provide mechanisms to protect user data and resources.

This TOE Security Objective is satisfied by ensuring that:

- FDP_ACC.1: The Discretionary Access Control policy applies to all operations between subjects and objects (tables, views, stored procedures and user-defined functions) controlled by the TOE.
- FDP_ACF.1: The subjects and objects under the discretionary access control policy will have certain rules that apply to all accesses between them. The rules are based on subject identities and access control lists associated with database objects that either grant or deny potential request access.
- FDP_RIP.2: When data is deleted from memory or storage (e.g., disk drive) it is often left intact and not truly erased. It is then possible for other users with access to the memory to view previously protected data. Therefore when a block of memory is allocated it is necessary to ensure all previous data stored in that block has been made unavailable.
- FMT_REV.1b: The discretionary nature of the policy allows users to modify access control permissions, which are represented by security attributes. Users are allowed to modify the security attributes of subjects and objects as permitted by the Discretionary Access Control policy.

8.2.1.10 O.RESIDUAL_INFORMATION

The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated.

This TOE Security Objective is satisfied by ensuring that:

- FDP_RIP.2: When data is deleted from memory or storage (e.g., disk drive) it is often left intact and not truly erased. It is then possible for other users with access to the memory to view previously protected data. Therefore when a block of memory is allocated it is necessary to ensure all previous data stored in that block has been made unavailable.

8.2.1.11 O.TOE_PROTECTION

The TOE will protect itself and its assets from external access, interference and tampering.

This TOE Security Objective is satisfied by ensuring that:

- FPT_RVM.1a: The TOE is required to allow access to protected objects only after it makes informed access decisions.
- FPT_SEP.1a: The TOE is required to protect itself and separate the contexts of its users.

8.2.1.12 O.USER_AUTHENTICATION

The TOE will verify the claimed identity of users.

This TOE Security Objective is satisfied by ensuring that:

- FIA_AFL.1: To prevent brute force attacks on authentication data, the administrator must specify an upper bound on the number of unsuccessful authentications that will be allowed. Surpassing that threshold could indicate a brute force user authentication attack, and the TOE needs to take appropriate action.
- FIA_SOS.1: User authentication is meaningful only if there is an extremely low probability of success for random attempts to authenticate as an authorized user. The requirement ensures that the secret authentication data is computationally difficult to guess randomly.
- FIA_UAU.2: Users must be authenticated before they can perform any TSF-mediated functions.
- FMT_MTD.1c: The user authentication data is to be set only by an authenticated individual in an authorized role.

8.2.1.13 O.USER_IDENTIFICATION

The TOE will uniquely identify users.

This TOE Security Objective is satisfied by ensuring that:

- FIA_ATD.1: Each database user will have a list of security attributes associated with them. They will have their unique identifier, any groups they may be a part of, for discretionary access control, any security roles they posses, and any other attributes assigned by the ST writer.
- FIA_UID.2: Users must be identified to the TOE before they can perform any TSF-mediated functions.
- FIA_USB.1: All subjects that act on behalf of users must have a binding that associates the subjects with a user uniquely.

8.2.1.14 OE.TIME

The IT environment will provide a time source that provides reliable time stamps.

This IT Environment Security Objective is satisfied by ensuring that:

- FPT_STM.1: The IT environment is required to provide a reliable time source.

8.2.1.15 OE.TOE_PROTECTION

The IT environment will provide protection to the TOE and its assets from external access, interference and tampering.

This IT Environment Security Objective is satisfied by ensuring that:

- FPT_RVM.1b: The IT environment is required to allow access to protected objects only after it makes informed access decisions.
- FPT_SEP.1b: The IT environment is required to protect itself and separate the contexts of its users.

8.3 Security Assurance Requirements Rationale

Adaptive Server Anywhere is targeted at a generalized IT environment with good physical access security and competent administrators. Within such environments it is assumed that attackers will have a relatively low attack potential. As such, EAL 3 (augmented with ALC_FLR.2) is appropriate to provide the assurance necessary to counter the potential for attack. Note also that this security target has defined an environment requiring more security than the U.S. Government Protection Profile Consistency Guidance for Basic Robustness, dated 24 July 2002, and that is comparable to or better than the historical notion of the C2 level of the Trusted Computer System Evaluation Criteria.

8.4 Strength of Function Rationale

Adaptive Server Anywhere is targeted at a generalized IT environment with good physical access security and competent administrators. Within such environments it is assumed that attackers will have a moderate attack potential. As such, a strength of function of ‘medium’ is appropriate for the intended environment. Note that the

only applicable mechanisms (i.e., those that are probabilistic or permutational) are related to identification and authentication (FIA_SOS.1, FIA_UAU.2, and FIA_UID.2).

8.5 Requirement Dependency Rationale

The following table represents an analysis of the dependencies of the security functional requirements (SFRs) in this security target. The first column identifies all of the SFRs in this security target. The TOE SFRs are highlighted in bold, unlike the IT environment SFRs, and all SARs are underlined. The second column identifies the minimum dependencies defined in the Common Criteria v2.1 and associated interpretations[8]. The third column identifies the actual requirements in this security target that correspond to the identified dependencies. Again, the corresponding TOE SFRs are highlighted in bold and SARs are underlined. Notice that this table demonstrates that all of the identified dependencies are satisfied with the exception of the dependency of FMT_MSA.2 on ADV_SPM.1. It also shows that the TOE has some dependencies on the IT environment, but the requirements for the IT environment have been defined such that it is not dependent upon the TOE.

While the Common Criteria defines ADV_SPM.1 as a dependency of FMT_MSA.2, this is not a true dependency. The TOE Summary Specification (TSS) provided in this Security Target (ST) in conjunction with the correspondence between the functional specification and the TSS required by ADV_RCR.1 (included in this ST) essentially require the information identified in ADV_SPM.1.

ST Requirement	CC Dependencies	ST Dependencies
FAU_GEN.1	FPT_STM.1	FPT_STM.1
FAU_GEN.2	FAU_GEN.1 and FIA_UID.1	**FAU_GEN.1** and **FIA_UID.2**
FAU_SAR.1	FAU_GEN.1	**FAU_GEN.1**
FAU_SAR.3	FAU_SAR.1	**FAU_SAR.1**
FAU_SEL.1	FAU_GEN.1 and FMT_MTD.1	**FAU_GEN.1** and **FMT_MTD.1a**
FAU_STG.1	FAU_GEN.1	**FAU_GEN.1**
FAU_STG.3	FAU_STG.1	**FAU_STG.1**
FDP_ACC.1	FDP_ACF.1	**FDP_ACF.1**
FDP_ACF.1	none	none
FDP_RIP.2	none	none
FIA_AFL.1	FIA_UAU.1	**FIA_UAU.2**
FIA_ATD.1	none	none
FIA_SOS.1	none	none
FIA_UAU.2	FIA_UID.1	**FIA_UID.2**
FIA_UID.2	none	none
FIA_USB.1	FIA_ATD.1	**FIA_ATD.1**
FMT_MOF.1	FMT_SMR.1 and FMT_SMF.1	**FMT_SMR.1** and **FMT_SMF.1**
FMT_MSA.1	FMT_SMR.1 and FMT_SMF.1 and (FDP_ACC.1 or FDP_IFC.1)	**FMT_SMR.1** and **FMT_SMF.1** and **FDP_ACC.1**
FMT_MSA.2	ADV_SPM.1 and FMT_MSA.1 and FMT_SMR.1 and (FDP_ACC.1 or FDP_IFC.1)	**FMT_MSA.1** and **FMT_SMR.1** and **FDP_ACC.1**
FMT_MSA.3	FMT_MSA.1 and FMT_SMR.1	**FMT_MSA.1** and **FMT_SMR.1**
FMT_MTD.1a	FMT_SMR.1 and FMT_SMF.1	**FMT_SMR.1** and **FMT_SMF.1**
FMT_MTD.1b	FMT_SMR.1 and FMT_SMF.1	**FMT_SMR.1** and **FMT_SMF.1**
FMT_MTD.1c	FMT_SMR.1 and FMT_SMF.1	**FMT_SMR.1** and **FMT_SMF.1**
FMT_REV.1a	FMT_SMR.1	**FMT_SMR.1**
FMT_REV.1b	FMT_SMR.1	**FMT_SMR.1**
FMT_SMF.1	none	none

[8] The only International Interpretation that affects the dependencies of the SFRs in this security target as of the date of the security target is International Interpretation #65. That interpretations introduces the SFR FMT_SMF.1 and alters FMT_MOF.1, FMT_MSA.1, and FMT_MTD.1 so that they are all dependent upon it.

FMT_SMR.1	FIA_UID.1	**FIA_UID.2**
FPT_RVM.1a	none	none
FPT_SEP.1a	none	none
FTA_MCS_EXP.1	FIA_UID.1	**FIA_UID.2**
FTA_TSE.1	none	none
FPT_RVM.1b	none	none
FPT_SEP.1b	none	none
FPT_STM.1	none	none
ACM_CAP.3	ACM_SCP.1	ACM_SCP.1
ACM_SCP.1	ACM_CAP.3	ACM_CAP.3
ADO_DEL.1	none	none
ADO_IGS.1	AGD_ADM.1	AGD_ADM.1
ADV_FSP.1	ADV_RCR.1	ADV_RCR.1
ADV_HLD.2	ADV_FSP.1 and ADV_RCR.1	ADV_FSP.1 and ADV_RCR.1
ADV_RCR.1	none	none
AGD_ADM.1	ADV_FSP.1	ADV_FSP.1
AGD_USR.1	ADV_FSP.1	ADV_FSP.1
ALC_DVS.1	none	none
ALC_FLR.2	none	none
ATE_COV.2	ADV_FSP.1 and ATE_FUN.1	ADV_FSP.1 and ATE_FUN.1
ATE_DPT.1	ADV_HLD.1 and ATE_FUN.1	ADV_HLD.2 and ATE_FUN.1
ATE_FUN.1	ADV_FSP.1 and AGD_ADM.1 and AGD_USR.1	ADV_FSP.1 and AGD_ADM.1 and AGD_USR.1
ATE_IND.2	none	none
AVA_MSU.1	ADO_IGS.1 and ADV_FSP.1 and AGD_ADM.1 and AGD_USR.1	ADO_IGS.1 and ADV_FSP.1 and AGD_ADM.1 and AGD_USR.1
AVA_SOF.1	ADV_FSP.1 and ADV_HLD.1	ADV_FSP.1 and ADV_HLD.2
AVA_VLA.1	ADV_FSP.1 and ADV_HLD.1 and AGD_ADM.1 and AGD_USR.1	ADV_FSP.1 and ADV_HLD.2 and AGD_ADM.1 and AGD_USR.1

Table 6 Requirement Dependencies

8.6 Explicitly Stated Requirements Rationale

This security target includes one explicitly stated requirement: FTA_MCS_EXP.1. This requirement is very similar to the CC Part 2 FTA_MCS.1 requirement; except that it only requires that the TOE *must be able* to limit concurrent user sessions as opposed to requiring that it always *must* do so. This explicit requirement was necessary since the CC does not provide the flexibility of having an optionally configured mechanism. As such, FTA_MCS_EXP.1 should be considered as an alternate version of FTA_MCS.1 that shares the same requirement class and family as well as dependencies.

8.7 TOE Summary Specification Rationale

Each subsection in Section 6, the TOE Summary Specification, describes a security function of the TOE. Each description is followed with rationale that indicates which requirements are satisfied by aspects of the corresponding security function. The set of security functions work together to satisfy all of the security functions and assurance requirements. Furthermore, all of the security functions are necessary in order for the TSF to provide the required security functionality.

This Section in conjunction with Section 6, the TOE Summary Specification, provides evidence that the security functions are suitable to meet the TOE security requirements. The collection of security functions work together to provide all of the security requirements. The security functions described in the TOE summary specification are all

necessary for the required security functionality in the TSF. **Table 7 Security Functions vs. Requirements Mapping** demonstrates the relationship between security requirements and security functions.

	Security audit	User data protection	Identification and authentication	Security management	Protection of the TSF	TOE access
FAU_GEN.1	X					
FAU_GEN.2	X					
FAU_SAR.1	X					
FAU_SAR.3	X					
FAU_SEL.1	X					
FAU_STG.1	X					
FAU_STG.3	X					
FDP_ACC.1		X				
FDP_ACF.1		X				
FDP_RIP.2		X				
FIA_AFL.1			X			
FIA_ATD.1			X			
FIA_SOS.1			X			
FIA_UAU.2			X			
FIA_UID.2			X			
FIA_USB.1			X			
FMT_MOF.1				X		
FMT_MSA.1				X		
FMT_MSA.2				X		
FMT_MSA.3				X		
FMT_MTD.1a				X		
FMT_MTD.1b				X		
FMT_MTD.1c				X		
FMT_REV.1a				X		
FMT_REV.1b				X		
FMT_SMF.1				X		
FMT_SMR.1				X		
FPT_RVM.1a					X	
FPT_SEP.1a					X	
FTA_MCS_EXP.1						X
FTA_TSE.1						X

Table 7 Security Functions vs. Requirements Mapping

8.8 PP Claims Rationale

See Section 7, Protection Profile Claims.