Tivoli Access Manager for Operating Systems 5.1 Security Target

Document Version Number 1.6.5

Document Creation Date: 2004-10-21

Document Update Date: 2006-02-01

Authors: Clemens Wittinger, David Ochel

Owner: Ron Willard

Note: This document will become a public document at the end of the evaluation

The responsibility for using the latest version of this document lies with the user of the document.

To request changes to this document, notify the document owner.

Table of Contents

History

This track of document changes is a part of the standard document tracking process.

Version	Date	Summary of Changes
0.1	2004-08-12	First version after draft reviews
0.2	2005-04-05	Work in progress at beginning of evaluation project: moved to CC 2.2; clarified usage of “users” and “administrators” throughout document; added boundary/component diagram; further completion of TOE configuration and underlying systems; identification of guidance; resolved redundancies in SPD and added threat against separation, further specified threat agents and assets; added O.SOF and OE.SEPARATION; removed support of TLSv1; removed FIA_UAU.1, FIA_UAU.5; splitted SFRs for underlying PS and RM systems; removed SFRs for non-IT environment; updated mapping tables in rationale; clarifications and augmentations in all chapters.
0.3	2005-04-14	Changes to evaluated configuration in 2.7; introduced A.PAM; clarified credential derivation in FIA_USB.1; refined FIA_USB.1
0.4	2005-04-27	Corrected section 1.2; editorial changes in 2.2; removed redundancies and provided clarifications in 2.7; clarified assets in 3.2; corrected application notes for FAU_SAR.1(2), FIA_AFL.1(2); updated password policy in FIA_SOS.1; refined SFRs in 5.2 to reflect IT environment; clarified access control support in 5.2.3; updated 5.3; clarifications in 6.1.2; correction in 6.1.3.9; updated 6.1.4.1 and 6.1.4.8; added “TOE environment” to glossary; further editorial changes.
0.5	2005-04-27	Clarification in 2.2; allow usage of pdadmin from TAMOS RM in 2.5, A.ADMIN; allow LDAP replicas in 2.7; throughout ST clarification that LDAP supports the authentication of administrators and the TOE enforces it; corrected identification of operations in chapter 5; corrected FAU_SEL.1(2); removed FAU_STG.4; updated 8.2.1; completed 8.2.2, 8.2.4 and following chapters; various editorial changes.
0.6	2005-05-02	Added clarifications on LDAP support for administrator authentication and assumptions for the support of LDAP replicas (A.FRIENDLY_LDAP, A.DIR_PROT, OE.FRIENDLY_DS, OE.REPLICAS).
0.7	2005-05-06	Editorial changes.
0.8	2005-05-18	Changes due to evaluation results.
0.9	2005-05-25	More changes due to evaluation results.
1.0	2005-06-06	Improved mapping for A.FRIENDLY_OS and performed refinements on FPT_SEP.1 and FPT_STM.1 for the IT environment to clarify the TOE’s dependency on the specified mechanisms.
1.1	2005-06-15	Clarified kernel versions in section 2.7.
1.2	2005-07-22	Corrected version of HP UX in 2.7. Added various user space programs to 6.1.3.10 and 6.1.4.8. Augmented OE.SEC_INTEGRATE to address protection of communication with LDAP server.

1.3	2005-08-03	Changed TAMOS fixpack level and SLES kernel versions in 2.7.
1.4	2005-10-10	Updates with regard to the evaluated configuration.
1.5	2005-10-11	Further clarifications with regard to the evaluated configuration in section 2.7
1.6	2005-11-04	Addressing comments from CB. Further adaptations to product capabilities.
1.6.1	2005-11-07	One typo is fixed.
1.6.2	2005-11-17	Clarified PS/RM selection in 2.7.
1.6.3	2005-11-17	Added Release Notes to TOE in 2.7; Changed FIA_AFL.1(1) and (2) to define minimum lockout period to match SOF analysis.
1.6.3	2005-12-14	Approved by Ron Willard. Official approval notice posted in the AMOS 2003 Teamroom
1.6.4	2006-01-25	Corrected typo in 6.1.3.1; added note on component leveling in section 5.1.1.
1.6.4	2006-01-27	Approved by Ron Willard. Official approval notice posted in the AMOS 2003 Teamroom
1.6.5	2006-02-01	Fixed enumeration in SFRs.

1 ST introduction

This document defines the Security Target for the Common Criteria evaluation of Tivoli Access Manager for Operating Systems, developed by International Business Machines Corporation (IBM). The sponsor for this evaluation at assurance level EAL3 is IBM.

1.1 ST Identification

Title: Tivoli Access Manager for Operating for Systems 5.1 Security Target, Version 1.6.5

Keywords: Access Control, ISO 10181-3, aznAPI

1.2 ST Overview

IBM Tivoli Access Manager for Operating Systems is a specific implementation of the access control framework defined by the ISO 10181-3 [ISO 10181-3] standard and the Authorization API (aznAPI) [AZNAPI].

Throughout the document the following terminology will be used to identify the TOE and its subsystems:

- “TAMOS product”, “TOE” - IBM Tivoli Access Manager for Operating Systems
- “TAMOS”, “TAMOS Resource Manager”, “TAMOS RM” - the Resource Manager (see section 2.3)
- “Policy Server” - the Policy Server

The TOE is an authorization solution for operating systems. The TOE's authorization services allow an organization to securely control user access to protected information and resources. By providing a centralized, flexible, and scalable access control solution, the TOE allows highly secure and well-managed operating systems infrastructures to be built.

In addition to its state-of-the-art security policy management feature, the TOE supports authentication decision enforcement for administrative users, the application of login and password policies to user accounts of underlying resources, authorization, data security, secure communication and resource management capabilities.

At its core, the TOE provides role based access control: the Tivoli Access Manager authorization service, accessed through a standard authorization API, provides permit and deny decisions on access requests from native Tivoli Access Manager resource managers, such as the TAMOS Resources Manager, and third-party applications.

The TAMOS RM is part of the TOE. It is the resource manager/authorization evaluator responsible for managing and protecting operating systems information and resources – it applies a centrally managed authorization and authentication policy to operating system resources. The following features are provided by the TAMOS RM:

- Authentication Policy Services – extends the native operating system authentication process by applying centrally managed login and password policies.
- Access control – protects operating system resources like files, directories, programs against unauthorized access.

1.3 CC Conformance Claim

This Security Target is based upon the

Common Criteria for Information Technology Security Evaluation, January 2004, Version 2.2, Part 1-3, CCIMB-2004-01-001, CCIMB-2004-01-002, CCIMB-2004-01-003 [CC]

For the evaluation, the following methodology is used:

Common Methodology for Information Technology Security Evaluation, January 2004, Version 2.2, CCIMB-2004-01-004 [CEM]

This Security Target is

- Part 2 extended
- Part 3 conformant

The evaluation assurance level is EAL3, augmented by ALC_FLR.1

1.4 Strength of Function

The claimed strength of function for this TOE is: **SOF-medium**

1.5 Definition of Terms

Authorized **users**	Individuals who have been successfully identified by the TOE and may access resources as defined by the access control policy of the TOE. This includes “unauthenticated” users where the TOE policy allows unauthenticated users access to resources. See also “Note”.
Unauthenticated **users**	Individuals who can not be identified by the TOE but are part of the user community allowed to access resources available to unauthenticated users. Note that all users of resources will be authenticated by the underlying operating system – the fact that the TOE does not recognize them as individual users makes them “unauthenticated” users from the perspective of the TOE. See also “Note”.
Base **administrators**	Individuals who have successfully authenticated themselves to the TOE as administrators and are allowed to perform administrative tasks via the *pdadmin* interface commands within their administrative responsibilities. See also “Note”.
Resource Manager **administrators** / TAMOS RM **administrators**	Individuals who have been successfully authenticated by the underlying operating system and have been identified by the TOE as administrators are allowed to perform administrative tasks via the TAMOS RM management commands within their administrative responsibilities. See also “Note”.
Note: **Administrators and Users**	Note that in the definition of the SFRs in chapter 5 the terms (a) “**users**” and (b) “**administrators**” are used to distinguish between (a) all users of the managed resource, including TAMOS RM administrators and (b) Base administrators only, if not noted otherwise.

2 TOE Description

The TOE consists of IBM Tivoli Access Manager for Operating Systems, Version 5.1 (TAMos). The following sections provide a description of the structure of the TOE, its boundaries, and an overview of the security functions provided by the TOE.

2.1 Abstract view: Access Control Framework according to ISO 10181-3 and the Open Group Authorization API

The TOE is a specific implementation of the access control framework defined by the ISO 10181-3 [ISO 10181-3] standard and the Authorization API (aznAPI) [AZNAPI]. The TOE uses the overall access control model and the interface described in those two standards. To explain those ideas we provide a short summary of them.

ISO 10181-3 contains the following figure to explain the general access control model:

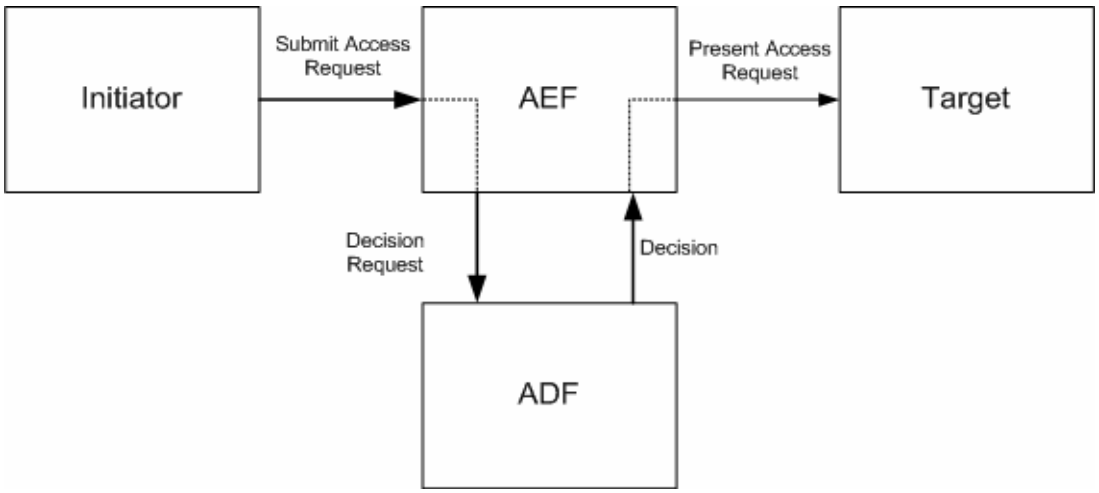

Figure 1: ISO 10181-3 fundamental access control functions

In this model an initiator submits an access request to the “Access Enforcement Function” (AEF) of a system. This function passes the request to an “Access Decision Function” (ADF), which makes the decision based on the rules of the access control system which may be based on:

- The identity and attributes of the initiator
- The identity and attributes of the resource being requested
- Contextual information (e. g. time and date, number of request from the same initiator, information from other systems)

Separating the access enforcement from the access decision function, as well as separating the access enforcement function from the actual target, allows the implementation of highly flexible access control and management systems in distributed environments. ISO 10181-3 actually is a general framework for such kind of access control and management system.

The Open Group now defines a standard for an application programming interface (API) for

the interface between the Access Enforcement Function (AEF) and the Access Decision Function (ADF) which allows AEF and ADF components from different vendors to co-operate. The following figure shows the aznAPI system structure as defined in [AZNAPI]:

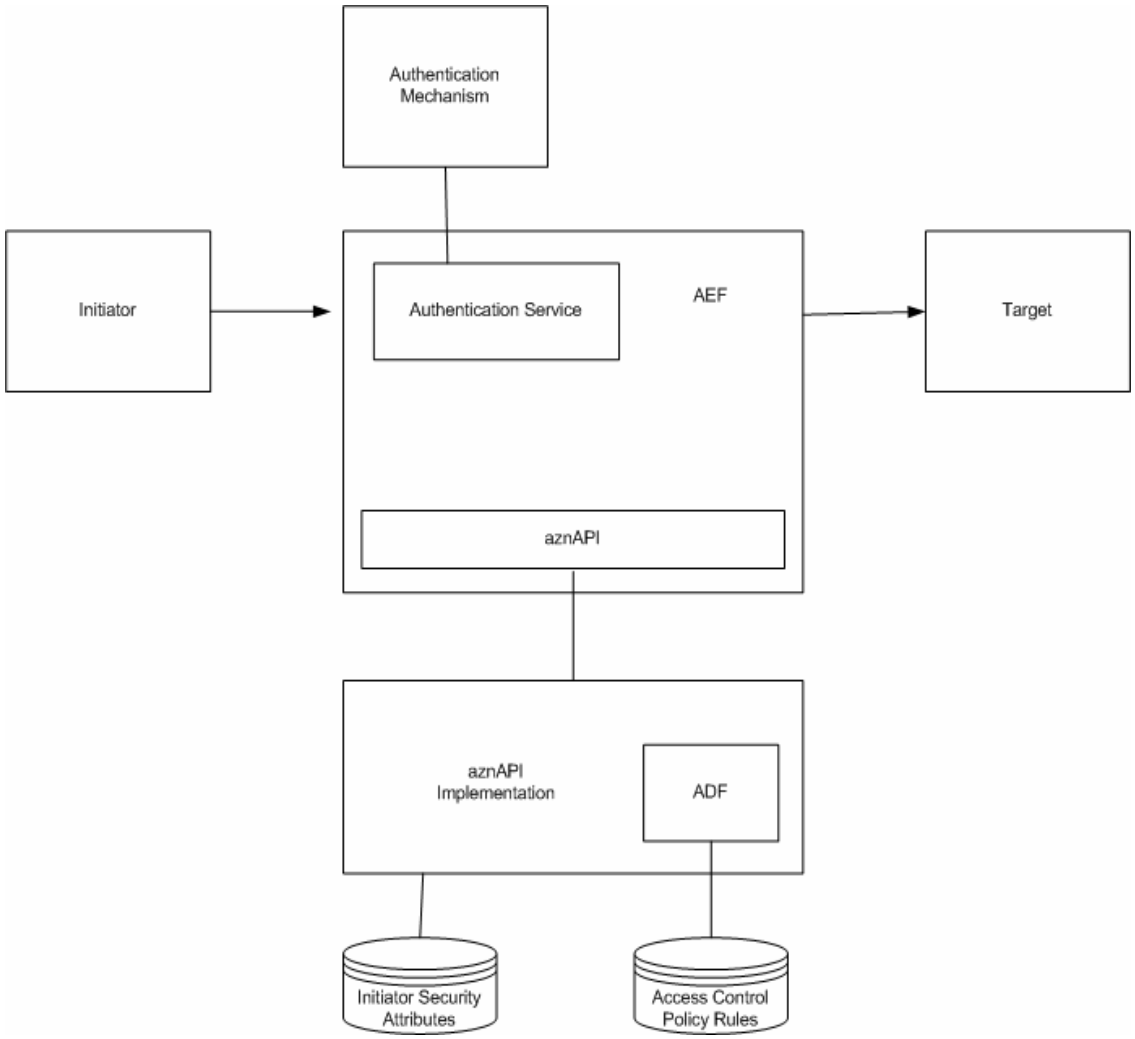

Figure 2: aznAPI System Structure as defined in the Open Group Standard

In this model the initiator submits his access request to the AEF, which then (if required by the policy) authenticates the identity of the initiator using the authentication service within the AEF. This authentication service may use an external authentication mechanism (e. g. a directory server storing user attributes and credentials).

The request together with the initiator attributes is then passed via the aznAPI interface to the implementation of the aznAPI, which in turn may store or request from an external entity additional security attributes of the initiator of the request. The attributes together with the information passed via the aznAPI about the request (including the target of the request) as well as the information about the initiator of the request is passed to the ADF component, which uses the Access Control Policy Rules stored in some kind of database.

2.2 Mapping the TOE to the aznAPI System Structure

The TOE is a specific implementation of the access control model defined in [ISO 10181-3] and [AZNAPI]. The overall TOE architecture and boundaries are illustrated in figure 3. With relation to the model defined in figure 2 the TOE includes the Access Enforcement Function (AEF) and the Access Decision Function (ADF) together with the Access Control Policy rules. The Authentication Mechanism for users is implemented in the IT environment (the TOE relies on the underlying operating system), while TOE administrators are authenticated by the TOE with the help of an LDAP user registry in the IT environment. The “Initiator Security Attributes” are stored in this LDAP user registry. Also, the target system which hosts the actual resources that are to be protected is not part of the TOE.

In this model a user submits a request for a resource (e. g. accessing a file that is protected by the TOE). This request is intercepted by the TOE, which implements access control checking on top of the native operating system functions. The TOE performs the following actions:

- Checking if the requested resource is known to be not protected. If this is true, the request is passed through to the operating system.
- Checking if the user has the right to access the requested resource for the requested operation. If not, the request is rejected. If yes, the request is passed through to the underlying operating system.

To explain how the access rights are checked an overview on the Tivoli Access Manager components is provided first (please see figure 3 for an architectural overview of the TOE).

The “Resource Manager” is implemented as part of the TOE by the TAMOS RM. This component includes also the “Authorization Evaluator” as a subsystem. The Resource Manager communicates with the “Authorization Evaluator” via the aznAPI.

The “Policy Server” is responsible to define and maintain the access control policy. It uses the “Master Authorization Policy” database to store the access control policy rules.

To speed up the time required to make an access decision, the “Authorization Evaluator” manages a replica of the “Master Authorization Policy”. The Policy Server informs all Authorization Evaluators about modifications to the “Master Authorization Policy” (actually what it does is to use a standard compression utility to compress the whole database and then transfer the whole new database). An Authorization Evaluator can also request the Policy Server to submit a new copy of the Master Authorization Policy (which it does upon startup, since there may be updates it could not obtain e.g. during a down-time). Also the Policy Server can request an Authorization Evaluator to update the replica of the Master Authorization Policy to make sure that the Authorization Evaluator has the latest version.

Administration of the TOE is performed via a workstation or terminal directly connected to the Policy Manager component. Only the command line interface and C language API for administration are part of the evaluated configuration. The C language API may be used by an organization to define its own tools to automate some of the administration tasks. But such tools would then be part of the IT environment and it is the responsibilty of the consumer to ensure that those tools perform their task correctly.

The TAMOS RM can be further managed by commands provided by the TAMOS RM. These commands can be initiated by an appropriately authorized user logged into the underlying operating system the TAMOS RM is running on.

Administration includes the management of the Master Authorization Policy (defining access rules for protected objects) as well as management of the TOE. It should be noted that ac-

cess rights of administrators to administrative objects of the TOE are also stored and maintained in the Master Authorization Policy; this specifically includes the management commands provided by the TAMOS RM.

To enhance authentication capabilities the TOE uses modules that extend the native authentication process of the host operating system by applying a login policy defined within the TOE.

For administrator authentication, the TOE uses a directory server. The directory server provides a repository for user and administrator attributes and credentials. Authentication of users is done by the Resource Manager in combination with host operating system, authentication of administrators is performed by the Policy Server (in the sense of the authentication service in figure 2) and uses the external Directory Server as the authentication mechanism.

The communication link between the TOE and the LDAP server is protected using the SSL v3 protocol. The TOE uses the GSKit library for the implementation of those protocols and their underlying cryptographic functions. The GSKit library is therefore part of the Policy Server and part of the Resource Manager. Also, the communication link between the Policy Server and the different Resource Managers is secured by SSL v3 using the GSKit library.

The Master Authorization Policy as well as the Replica Authorization Policy are databases. The Master Authroization Policy is a database held by the Policy Server and the Replica Authorization Policy is a database held by each Authorization Evaluator.

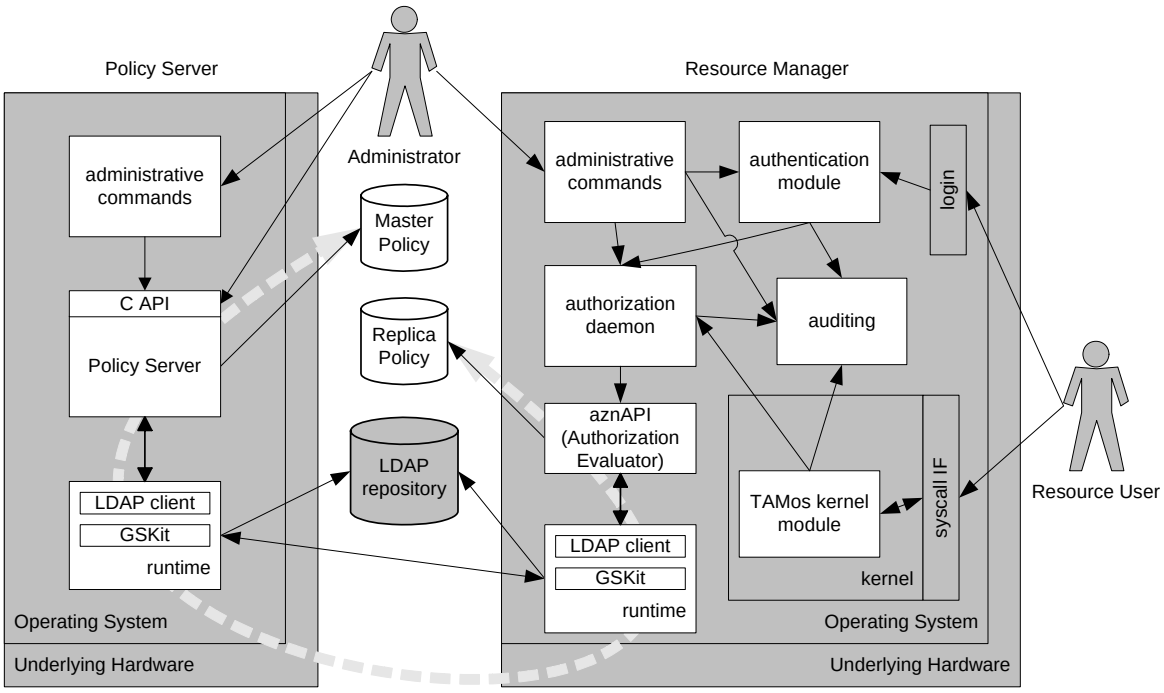

Figure 3: TOE components and boundary –TOE parts are displayed without color (white), the parts of the TOE environment are gray. The dashed line depicts the information flow during policy synchronization.

The TOE architecture (showing also the directory server and the servers holding the resources, although they are not part of the TOE) is shown in Figure 3. Please refer to Figure 4 for detailed schematics of the resource manager (section 2.3)

The TOE now maps in the following way to the system structure shown in Figure 2 as de-

fined in [AZNAPI]:

- The “Initiator” maps to the client
- The “Access Enforcement Function” (AEF) **and** the “Authentication Service” map to the Resource Manager (part of TAMOS RM)
- The “Access Decision Function” (ADF) maps to the “Authorization Evaluator” (part of TAMOS RM)
- The “aznAPI” maps to the “aznAPI”
- The “aznAPI” implementation maps to the “Authorization Evaluator” **and** the “Policy Server”.
- For administrative authentication within the “Policy Server”
    - The “Authentication Mechanism” **and** the “Initiator Security Attributes” map to the “Directory Server” (which is not part of the TOE but part of the TOE environment)
- For user authentication within the “Resource Manager” (part of TAMOS RM)
    - The “Authentication Mechanism” maps to the underlying operating system
    - The “Initiator Security Attributes” maps to the “Directory Server” and the “Policy Server”
- The “Access Control Policy Rules” map to the “Master Authorization Policy” **and** the “Replica Authorization Policy”
- The “Target” maps to the “Target”

2.3 Resource Manager (TAMOS RM)

The Resource Manager shown in the previous figure as part of the TOE is responsible to protect the resources on the host system (operating system resources).

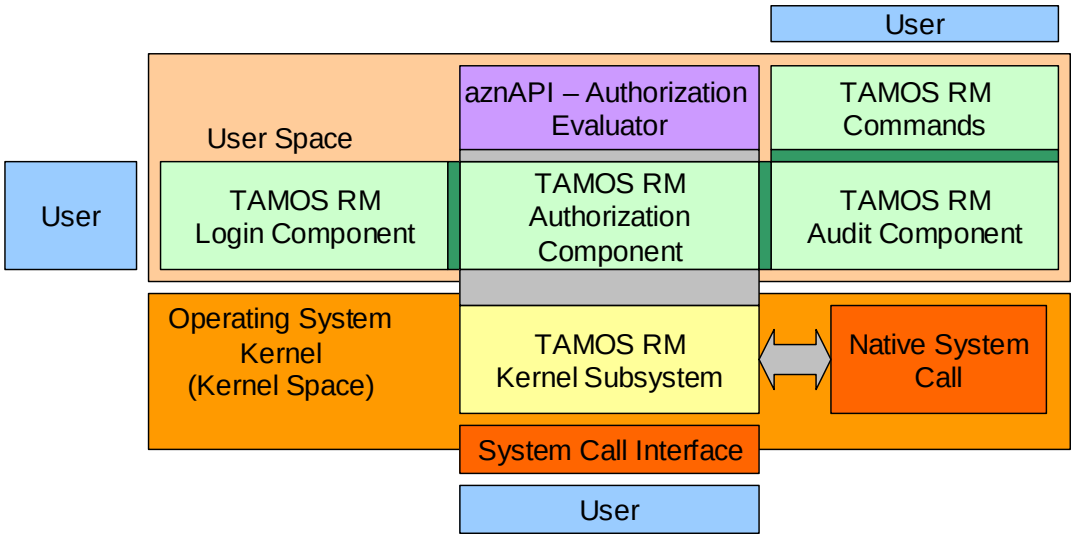

Figure 4: Resource Manager Structure

Figure 4 shows the structure of TAMOS RM comprising

- The TAMOS RM authorization component
- The TAMOS RM login component
- The TAMOS RM audit component
- The TAMOS RM commands component

and

a. The TAMOS RM kernel subsystem

all of which are part of the TOE.

The authorization mechanism of the TOE is capable of performing and enforcing access control decisions for users of a managed resource with respect to the following object types:

b. file system resources

c. outgoing and incoming network connections

d. login and password control services

e. changes of user and group identity

f. sudo commands

2.3.1 Authorization Component

The Authorization Daemon (pdosd) is a multi-threaded daemon process that processes all authorization requests for the TAMOS RM. Authorization requests are initiated when the kernel intercepts a system call and sends a request for an authorization decision to the pdosd daemon. The pdosd daemon processes that request and sends a decision of permit or deny back to the kernel, which, in turn, either allows the system call to continue or returns an error to the caller.

The pdosd daemon relies on the Authorization Evaluator (aznAPI) to manage and store the security policy in the policy database. The pdosd daemon also uses the Tivoli Access Manager user registry (LDAP) for storage and retrieval of user credentials, which are used when making an access decision.

Users do not authenticate directly to the TAMOS RM, but TAMOS RM extends the operating system authentication process (see section 2.3.3).

When a user logs into a TAMOS RM protected endpoint (i.e. the underlying operating system), after the user has successfully been authenticated by the native authentication mechanism, the user's credentials are retrieved from the user registry via the aznAPI.

2.3.2 Audit Compontent

The TOE provides auditing for security-relevant events. TAMOS RM provides extensive auditing capabilities. Audit levels can be set globally, for a specific protected resource, or on a per-user basis. For global- and resource-level auditing, the permit and deny levels can be further qualified so that they are only in effect for specified TAMOS RM actions. All of the TAMOS RM components as well as the commands generate audit data.

2.3.3 Login Component

The Login Component of TAMOS RM is used to enforce the login and password policies that are supported by TAMOS RM. The mechanism that is used to insert TAMOS RM's login subsystem in the login path to allow further authorization decisions is a loadable authentication module on AIX and a pluggable authentication module (PAM) on all other supported platforms.

The supported policies allow to define time-of-day restrictions, restricting login to individual local and remote terminals, and login-activity-related policies such as password expiry, automatically disabling accounts after a number of consecutive failed logins, and automatically disabling inactive accounts.

2.3.4 Commands Component

The commands component of TAMOS RM consists of administrative commands that are used to manage, view, modify, or control various aspects of the TAMOS RM.

2.3.5 Kernel Module

The TAMOS kernel module intercepts system calls and enforces policy defined within the TOE on the actions being performed (see section 2.3.6).

2.3.6 Authorization Flow

1. An operating system resource is being accessed by a caller.
2. The TAMOS Kernel Module intercepts the system call and checks to determine if the object the call is trying to access is subject to policy by means of querying the TAMOS Authorization Component or using a local cache. If not, the TAMOS Kernel Subsystem passes control to the native operating system call.
3. The kernel sends a request to the TAMOS Authorization Component, passing on the accessing user’s identification, the type and name of the operating system object, the action being performed and the executable used to perform the action.
4. The TAMOS Authorization Component uses this information from the kernel to retrieve relevant user credentials, construct the protected object name, and map the specified actions to the relevant TAMOS RM permissions. It then calls the Authorization Evaluator with this information to determine if this access should be allowed.
5. The Authorization Evaluator returns an access decision. The TAMOS Authorization Component uses this information to make a final decision if access should be allowed.
6. An audit record is eventually generated indicating, amongst others, the object, the initiator, the operation and the outcome of the decision.
7. The TAMOS Authorization Component passes the result back to the TAMOS Kernel Module.
8. The TAMOS Kernel Module, depending on the result, denies the request signalling an error condition to the caller, or passes the request to the native operating system call implementation.

2.4 Authorization Evaluator

This component is running as a part of a TAMOS RM system. When the Resource Manager (TAMOS RM) calls functions of the aznAPI to check if the client has the right to access the resource in the intended way, the Authorization Evaluator will check the local Replica Authorization Policy to decide if the request can be granted or not.

The Authorization Evaluator is also responsible for synchronizing the local replica of the Master Authorization Policy with the Policy Server. The Authorization Evaluator provides a mechanism to ask the Policy Server if its version is still up-to-date (which it always does on start-up and continuously during operation) and will store a new version of the replica database on demand of the Policy Server. In addition the Authorization Evaluator will serve additional system management commands coming from the Policy Server.

The Authorization Evaluator sets up a secured communication channel with the Policy Server using the SSL v3 protocol with client and server authentication. The Policy Server will take the role of the server while the Authorization Evaluator will take the role of a client. Before an Authorization Evaluator can set up such a communication channel, it needs to have its public key certified by the Policy Server (which acts as a certification authority for SSL client certificates within the TOE).

2.5 Policy Server

This component is responsible for the management of the Master Authorization Policy. It provides a separate interface for administrators (pdadmin interface) as a command line interface as well as a C API on the Policy Server. To perform administrative actions an administrator has to identify and authenticate via these interfaces. Only password based authentication is possible at this interface.

An administrator using a remote terminal or remote workstation to connect to the Policy Server for administration must ensure that the remote terminal or workstation is in a secured environment and managed securely. Management is performed by setting up a secured connection to communicate with a shell of the operating system on the Policy Server. There the administrator invokes the pdadmin command line interface and authenticates him- or herself to the TOE. Note that the security measures to protect the remote terminal or remote workstation as well as the security measures used to protect the communication link between the remote terminal or workstation are not part of the TOE but have to be assured in the TOE environment. Alternatively, administrators can invoke the pdadmin command on a TAMOS RM.

Administrators can now define and/or modify rules in the Master Authorization Policy as well as perform administrative actions for the remote Authorization Evaluator or Resource Manager components. Whenever administrators modify the Master Authorization Policy, a request is sent to all Authorization Evaluator components for synchronization of their replica database.

The Policy Server uses the Master Authorization Policy to store access control rules for system management objects and uses the Directory Server to store attributes and credentials for administrators. Management of the TOE can only be performed via the pdadmin interface of the Policy Server, since the “management objects” are not known to the Resource Manager or the Authorization Evaluator.

The Policy Server generates audit records for administrative actions.

2.6 GSKit

GSKit is a library that implements the SSL Version 3 protocol to secure the communication between the TOE and other trusted systems (including the LDAP server) as well as the communication between distributed parts of the TOE. SSL is also used to secure the communication between the TOE and client systems. GSKit is part of each Resource Manager/Authorization Evaluator system as well as the Policy Server. Since it provides functions to secure the communication between the involved machines, it is part of the TSF. On the Policy Server this component also provides the functions to generate and sign the certificates for the public keys of the TOE.

The functions provided by GSKit include those required for the generation of public/private RSA key pairs, generation of X.509 V3 certificate and certificate management (signing, distribution and revocation). RSA key pairs are required for the SSL ciphersuites supported by the TOE.

Within the TOE the Policy Server component is used as a Certification Authority for the certificates of the other TOE components. When a new TAMOS RM system is added to the TOE, during the installation of those components an RSA key pair is generated and the key pair and the PDCA certificate (the CA certificate of the Policy Server) is imported via the pdadmin interface. Afterwards the new TAMOS RM system and the Policy Server component can set up a trusted communication path with mutual authentication using the SSL v3 protocol with client and server authentication.

2.7 TOE configuration

The following describes the specifics of the configuration of Tivoli Access Manager for Operating Systems, Version 5.1 that conforms to the description in this Security Target and is henceforth called the evaluated configuration:

a. The evaluated configuration has one Policy Server system and one or more Resource Manager/Authorization Evaluator systems:
    - The Policy Server component of the TOE is installed and operated on a dedicated system within a physically protected environment. Optionally, the TAMOS RM can be installed on the Policy Server.
    - Resource Manager and Authorization Evaluator are always installed and operated on the same system. The evaluated configuration does not include Authorization Evaluator components running on a machine separate from the Resource Manager that uses them.
    - All Resource Manager/Authorization Evaluator systems operate independent from each other and are only connected to the central Policy Server.
    - The Policy Server and all the Resource Manager/Authorization Evaluator systems only use the operating systems platform combinations as defined in Table 1.

b. The following components and/or configurations are not part of the evaluated configuration and must not be used:
    - The use of the Web Portal Manager and integration with the Tivoli Desktop for the administration of the TOE is not supported. Instead only the command line interfaces of pdadmin and TAMOS RM and the pdadmin C API are supported in the evaluated configuration.

- No Application Development Kit is installed.
- Active Directory is not supported. Only LDAP is supported as interface to the user registry. Multiple LDAP replicas are supported, whereas the number of LDAP masters in the environment is restricted to one.
- No hardware encryption device is used. The cryptographic services are fully provided by the software implementation of the GSKit component.
- Language packs other than English are not supported. Only the English language pack for the TOE is evaluated.
- Non-certified authentication mechanisms and non-password based authentication mechanisms are not supported. On underlying operating systems managed by TAMOS RM the TOE supports only password-based authentication mechanisms listed in the Administrator Guide for the TAMOS ROM as being certified. For administrators requesting access via the pdadmin interface for the Policy Server, only password-based authentication is supported.
- Weak or no encryption of internal communications. Communication between the LDAP server and the TOE as well as the communication between the Policy Server and the Resource Manager/Authorization Evaluator systems is protected using the SSLv3 protocol with one of the ciphersuites defined in this Security Target. The use of unencrypted communication is disabled in the TOE.

c. The following components of the TAMOS product are not evaluated as part of the TOE, but can be used and are then considered part of the IT environment:
   - the log router daemon operating on the audit log files for remote distribution
   - the Tivoli Enterprise Console daemon providing audit data for remote access

d. The following components of the TAMOS product are part of the TOE, but do not contain security functionality that is subject to evaluation:
   - the integrated Watchdog functionality of the TOE
   - The TCB mechanisms of the TAMOS product, such as checksum verification of files being defined as a member of the TCB,

To set up the evaluated configuration compliant with the description above the user needs to follow the guidance provided in the TAMos 5.1 Common Criteria Guide.

The system components to be installed are:

1. Policy Server:
   a. Global Security Toolkit (GSKit) 7.0.3.3
   b. Tivoli Directory Server 5.2 Client
   c. Tivoli Access Manager 5.1 runtime
   d. Tivoli Access Manager 5.1 policy server
   e. Fixpack 06 for Tivoli Access Manager 5.1
2. Resource Manager/Authorization Server (TAMOS RM)
   a. Global Security Toolkit (GSKit) 7.0.3.3
   b. Tivoli Directory Server 5.2 Client
   c. Tivoli Access Manager 5.1 runtime (including the authorization evaluator)

d. Tivoli Access Manager for Operating Systems 5.1
e. Fixpack 06 for Tivoli Access Manager 5.1
f. Fixpack 17 for Tivoli Access Manager for Operating Systems 5.1

With one exception, noted in the table below, the Policy Server and all Resource Manager/Authorization Evaluator within an evaluated configuration use the same operating system platform as underlying system (but may run on different machines). The platforms with their corresponding TAMos resource manager modules that are covered by this evaluation are the following:

- IBM AIX 5.2
  a. 32-bit (SMP / UP)
  b. 64-bit (SMP / UP)
- Sun Solaris 2.8
  a. 32-bit (SMP / UP)
  b. 64-bit (SMP / UP)
- HP-UX 11i V1
  a. 64-bit (SMP / UP)
- RedHat Enterprise Linux AS/WS Version 3 Update 2 on i386 architectures (RHEL3)
  a. kernel-2.4.21-15.EL.i686.rpm (UP)
  b. kernel-smp-2.4.21-15.EL.i686.rpm (SMP)
- Novell SUSE Linux Enterprise Server 8 Service Pack 3 on i386 architectures (SLES8)
  a. k_deflt-2.4.21-295.i586 (UP)
  b. k_smp-2.4.21-295.i586 (SMP)

Resource Manager(s) can be run in arbitrary combinations of their variations listed above, e.g. it is permitted to operate a Policy Server on AIX 5.2.with Resources Managers running on AIX 5.2 32-bit SMP, AIX 5.2 32-bit UP, AIX 5.2 64-bit SMP and AIX 5.2 64-bit UP in the same environment.

This yields to the system combinations outlined in the table below being covered by this evaluation.

Combination	Policy Server	TAMOS Resource Manager(s)
1	AIX 5.2	AIX 5.2
2	Solaris 2.8	Solaris2.8
3	HP-UX 11i	HP-UX 11i
4	RHEL3	RHEL3
5	SLES8	SLES8, Solaris 2.8 (64bit SMP/UP)

Table 1: Valid Platform Combinations

The following guidance documents are part of the TOE:

- IBM Tivoli Access Manager for Operating Systems Release Notes

- IBM Tivoli Access Manager for Operating Systems Installation Guide
- IBM Tivoli Access Manager for Operating Systems Administration Guide
- IBM Tivoli Access Manager for Operating Systems Problem Determination Guide
- IBM Tivoli Access Manager Base Installation Guide
- IBM Tivoli Access Manager Base Administration Guide
- IBM Tivoli Access Manager for e-business Release Notes
- IBM Tivoli Access Manager for e-business Administration C API Developer Reference
- IBM Tivoli Access Manager for e-business Command Reference
- IBM Tivoli Access Manager Error Message Reference
- IBM Tivoli Access Manager for e-business Problem Determination Guide
- IBM Global Security Kit Secure Sockets Layer Introduction and iKeyman User's Guide

2.8 TOE User Types

The TOE deals with several logically distinct types of users:

1. Administrative users ("administrators")
    a. **Base administrators** managing the central policy using the *pdadmin* command or the administration C API. These users are authenticated by the Policy Server in combination with the LDAP server.
    b. **Resource manager administrators** using the administrative commands on the TAMOS resource manager. Those are specifically authorized operating system users that are able to manage TAMOS RM specific aspects. They are authenticated by the native operating system's mechanisms. The authorization of these administrators is managed by the TOE using its central policy.
2. Non-administrative users ("users")
    a. **Resource users** accessing the operating system resources. These users are authenticated by the native operating system's mechanisms. TAMOS RM applies a centrally managed login policy to the native operating system login process. The authorization of these users is managed by the TOE using the central policy.

Note that from a technical point of view there is no distinction between administrative and non-administrative users, they are commonly referred to as users. The distinction is made on the logical level based on the fact whether users are privileged to perform administrative tasks for the TOE or not. To this extent, "root" users on the underlying operating system resource are non-administrative users from the TOE's perspective.

Note also that in the definition of the SFRs in chapter 5 the terms (a) "**users**" and (b) "**administrators**" are used to distinguish between (a) all users of the managed resource, including TAMOS RM administrators and (b) Base administrators only, if not noted otherwise.

2.9 TOE Boundary

Figure 3 shows the boundary of the TOE. It shows that the Policy Server, the. Resource Manager/Authorization Evaluator, the Master Authorization Policy database and the Replica

Authorization Policy database are part of the TOE. It also shows that the host operating system, the LDAP server and the managed resources (client operating systems) are all part of the TOE environment.

2.10 TOE Security Model

2.10.1 Components

The TOE Security Model has the following components:

1. A User Registry (LDAP)

   The user registry contains all users and groups allowed to participate in the Tivoli Access Manager secure domain.

   Note: The LDAP Server is not part of the TOE!

2. A Master Authorization Policy Database

   This database contains a representation of all resources in the domain (= protected object space). The security administrator can define Access Control Lists (ACL) and Protected Object Policies (POP) and Access Control Extended Attributes for those resources that require protection.

3. An Authentication Policy Service (TAMOS RM)

   This service extends the verification of the claimed identity of a user by applying a login policy that takes into account various attributes stored in the user registry and policy database. The user's identification information is obtained from the User Registry. If a user's authentication information could not be obtained from the User Registry, this user is treated as unauthenticated by TAMOS RM.

   Note: The login policy does not cover the verification of the user's authentication secrets.

4. An Administration Authentication Service (Policy Server)

   This service verifies the claimed identity of a Base administrator. All administrators that are going to be authenticated must have an entry in the User Registry.

   When an administrator is successfully authenticated a set of identification information (credentials including user identity, group membership and security attributes) is extracted from the information stored in the User Registry and maintained for the user within the TOE.

5. An Authorization Service

   For each attempted access this service verifies if the user attempting access has the right to access the resource in the intended way. This is done by comparing the user’s credentials with the rules defined for the resource in the Authorization Policy Database. The Authorization Service is called by the Resource Manager and returns “yes” or “no” depending on the evaluation of the rules from the database.

   For Resource Manager Administrators and Resource Users the Authorization Evaluator creates the user credentials used in the TOE’s authorization evaluation by deriving the user name from the underlying system’s user registry by using the users’ numerical ID intercepted during login and matching this user name to the corresponding entry in the policy data base.

6. An Audit Service

   A configurable number of events will generate an audit record that allows tracing of the time the event happened and the user that caused the event.

7. An Administration Interface for the Policy Server

   This interface is used to administer the Policy Server and the central authorization policy.

8. An Administration Interface for TAMOS RM

   This interface is used to administer specific security functions of the TAMOS RM.

9. Configuration Files

   A number of configuration files are used by the components of the TOE. The settings of those files define the behavior of the security functions of the TOE. Configuration files need to be defined correctly at the installation time of the TOE to ensure a secure initial configuration. The administrator will maintain configuration files by:

   1. Using the administration commands of the pdadmin interface (Policy Server).
   2. Directly modifying the files (Policy Server).
   3. Using administrative commands (TAMOS RM).

2.10.2 Security Functionality

The TOE has the following security functionality:

1. Authentication of Base administrators

   Administrators allowed to administer the TOE via the pdadmin interfaces (C API and command line interface) are identified and authenticated.

2. Authentication Policy for Resource Manager administrators and Resource Users

   The authentication process of the underlying operating system for these users is extended by applying a TOE authentication policy.

   Note that the verification of the user's authentication secrets is not part of the authentication policy's application.

3. Assigning credentials to authenticated administrators

   The credentials of authenticated administrator are eventually created from the information stored in the user registry within the LDAP server.

4. Assigning credentials to authenticated users

   The credentials of authenticated users are eventually created from the information stored in the user registry within the LDAP server. If no user record is defined in the user registry, the user is treated as unauthenticated.

5. Access Control to protected objects of the underlying system

   Those objects are protected as defined in the policy defined by an administrator on the Policy Manager.

6. Access Control to TOE management objects

   A flexible management model can limit the administration capabilities of administrative users to defined sections of the protected object space.

7. Auditing of activities

   The TOE is capable of auditing defined events.

8. Secure Communication

   The TOE employs secure communication channels to communicate between its entities.

3 TOE Security Environment

3.1 Assumptions

The description of assumptions describes the security aspects of the environment in which the TOE will be used or is intended to be used. This includes the following:

- information about the intended usage of the TOE, including such aspects as the intended application, potential asset value, and possible limitations of use; and
- information about the environment of use of the TOE, including physical, personnel, and connectivity aspects.

A.ADMIN	The system administrative personnel are not careless, willfully negligent, or hostile, and will follow and abide by the instructions provided by the administrator guidance. Base administrators will perform administration activities from a secure environment using terminals and/or workstations they trust via secured connections to the Policy Server. All administrative commands themselves will be executed on the Policy Server or on a TAMOS RM.
A.BOOT	During operating system startup it is ensured that the TAMOS RM is started before user logins can occur.
A.DIR_PROT	The directory server used by the TOE provides protection mechanisms against unauthorized access to TSF data stored in the directory. This includes the assumptions that queries are properly authenticated, that replicas are held consistent according to a well-defined policy, and that communication between TOE and LDAP server is SSL v3 encrypted. .
A.FRIENDLY_OS	The underlying operating system of a resource manager works as specified. In particular, the operating system kernel is assumed to be well behaved with regard to the TSF parts operating in kernel mode. It does not alter, hinder or otherwise influence the kernel mode operation of the TOE, it rather supports them.
A.OS_CONF_MGMT	The operating systems of the machines running the TOE are assumed to be configured and maintained by trained and trustworthy personnel such that the underlying systems provide a reliable basis for the operation of the TOE software.
A.PAM	The authentication mechanisms in the underlying operating system for the TAMOS RM effectively identify the operating system users and associate them with correct user IDs. Furthermore, the PAM mechanism (or loadable authentication module mechanism on AIX) enforces the invocation of the TOE's login module, if so configured.

A.PHYS_PROT	The machines running the TOE software need to be protected against unauthorized physical access and modification. All machines running parts of the TOE software require this protection.
A.PWD_SAFE	Administrators and other users have to protect their passwords used for authentication to the TOE such that no unauthorized access to them is possible.
A.USER	Users of the TOE are not hostile and do not try to deliberately attack the TSF. Especially, they do not possess greater attack potential as assumed in the description of the threat environment for the TOE.
A.USER_PASSWORD	The underlying operating system for a resource manager ensures that users are authenticated.
A.FRIENDLY_LDAP	The LDAP server performs its functions as specified.

3.2 Threats

This section identifies the security threats originating from the TOE environment. The TOE counters the general threat of unauthorized access to information, where “access” includes disclosure, modification and destruction.

The **assets** to be protected are comprised of the information stored, processed or transmitted by the TOE, and the information that the TOE mediates access to. The term “information” is used here to refer to all data used by the TOE to enforce the TOE security policies, regardless of the location of such data, and to information in the TOE environment that the TOE mediates access to.

The assets to be protected are therefore:

- TSF data, especially
    - The Authorization Policy database (including replicas and caches of this database) which stores the data upon which access decisions are taken.
    - The user registry hosted on an LDAP server in the IT environment, which contains the user and group definitions utilized by the TOE.
- User data of the managed resources
    - The TOE is not directly and solely responsible for the protection of data stored on managed resources, but it contributes to their protection by making and enforcing decisions on access to those resources, making them assets in terms of the TSP.

The **threat agents** can be categorized as:

- Authorized users of the managed resource

  Those are authorized users of the TOE and the underlying operating system, i.e. individuals who have successfully authenticated themselves against the underlying operating system and are authorized to access resources in the underlying operating system as mediated by the TOE’s access control policy. Note that those users do not necessarily need to be known to the TOE, in which case the TOE will treat them as “unauthenticated” with respect to the defined access control policies – this still requires that they have been authenticated successfully by the underlying operating system.

- Authorized administrators of the TOE

  These individuals have successfully authenticated themselves to the TOE and may perform administrative tasks via the pdadmin interface within their administrative responsibilities. A special case are TAMOS RM Administrators, which have been authenticated by the underlying operating system of a resource manager and are granted special privileges to use the management functions of the TOE on the managed resource.

- Network-based attackers

  This class of threat agents does not possess any accounts on the managed resource or the TOE itself, but has access to the network or parts of the network that connects policy server, resource managers, and/or LDAP repositories.

The threat agents are assumed to originate from a well managed user community in a non-hostile working environment, and hence the TOE protects against threats of inadvertent or casual attempts to breach the system security. The TOE is not intended to be applicable to circumstances in which protection is required against determined attempts by hostile and well funded attackers to breach system security. An example of an environment meeting the intended assurance level of this evaluation is a company, providing operating system access to employees, well protected from external attacks and with an overall user community that can be assumed to be non-hostile.

System administrators of the TOE as well as those for the underlying operating system and the LDAP repository in the IT environment are assumed to be trustworthy and follow the instructions provided to them with respect to the secure configuration and operation of the systems under their responsibility. Also, assumptions on the protection of the policy server are made. (See section 3.1.)

The threats listed below are grouped according to whether or not they are countered by the TOE. Those that are not countered by the TOE are to be countered by environmental or external mechanisms.

3.2.1 Threats to be countered by the TOE

T.BYPASS	An authorized user of a managed resource or network-based attacker accesses resources protected by the TOE in a way that bypasses the TSF, exploiting non-TSF portions of the TOE.
T.COM_ATT	An attacker intercepts the communication between the TOE and an external entity or between distributed parts of the TOE in order to get access to protected information, to impersonate as an authorized user or part of the TOE or to manipulate the data transmitted between the TOE and an external or internal entity.
T.UAACTION	An undetected violation of the TSP may be caused as a result of an authorized user of a managed resource or network-based attacker attempting to perform actions that they are not authorized to do.
T.UAUSER	An authorized user of a managed resource or a network-based attacker may impersonate an authorized user of the TOE. This includes the threat of an authorized user that tries to impersonate as another authorized user without knowing the authentication credentials.

3.2.2 Threat to be countered by the TOE environment

TE.BYPASS An authorized user of a managed resource or network-based attacker accesses resources protected by the TOE in a way that bypasses the TSF due to the absence of protection mechanisms in the underlying system.

3.3 Organizational Security Policies

The following organizational security policies are deemed appropriate in a security environment for the TOE:

P.ACCOUNTABILITY The administrators of the system shall be held accountable for their actions within the system.

P.ADM_DELEGATION Specific administration tasks as well as management operations to defined subsets of the resources protected by the TOE may be delegated to administrators that are only allowed to perform the management tasks within their defined area of responsibility and are not able to extend this area themselves.

4 Security Objectives

This section defines the security objectives of the TSF and its supporting environment. Security objectives are categorized as IT security objectives for the TOE or the IT environment as well as non-IT security objectives to be met by organizational means in the TOE environment.

4.1 Security Objectives for the TOE

O.ACC_ADM	The TSF must control the definition and management of access control rules, policies and other security function properties and restrict those activities to authorized administrators. The TSF must allow restricting the rights of some administrators to define access control rules for a subset of the protected object space only.
O.AUDITING	The TSF must record the security relevant actions of users and administrators of the TOE. The TSF must present this information to authorized administrators.
O.AUTHENT_ADMIN	The TSF must enforce the authentication of Base administrators which request access to the TOE and its resources. **Note**: The authentication decision is derived from an external LDAP server.
O.AUTHORIZATION	The TSF must ensure that only authorized administrators and users gain access to the TOE and the resources it protects. **Note:** The access control rules may also allow unauthenticated users, i.e. such users that are not known to the TOE, to access resources explicitly defined to be accessible to unauthenticated users.
O.SEC_COM	Communication between physically distributed parts of the TOE must be secured to ensure the integrity and confidentiality of the communication.
O.SOF	The TSF must enforce a password policy for the TOE's administrators and users that protects against attacks of attackers with moderate attack potential.

4.2 IT Security Objectives for the Environment

4.2.1 IT Security Objectives for the underlying operating system

OE.OS_AUTH	The underlying system of a resource manager must reliably authenticate users.
OE.OS_CFG_PROT	The underlying operating systems for Policy Server and resource manager must provide protection for files comprising the TOE and for shared memory containing TSF or user data against unauthorized access.

OE.OS_TIME The underlying operating systems within IT environment must provide a reliable time source.

OE.SEPARATION The underlying system for a resource manager must provide proper separation mechanisms protecting the TOE and itself from interference and tampering by untrusted subjects.

4.2.2 IT Security Objectives for the LDAP server

OE.DS_ACCESS_CTRL The LDAP server must provide authentication and access control mechanisms to prohibit unauthorized access to directory entries containing TSF data. This access control must be enforced when importing and exporting data.

OE.FRIENDLY_DS The LDAP server must perform according to its specification, in particular the bind and attribute compare functions and the replication mechanisms.

4.3 Non-IT Security Objectives for the Environment

OE.CREDEN Those responsible for the TOE must ensure that all access credentials, such as passwords or other authentication information, are protected by the users in a manner which maintains IT security objectives. The TOE environment must ensure that authentication data of TAMOS RM users, i.e. users logging in to managed resources, are adequately protected against disclosure when in transit.

OE.INSTALL Those responsible for the TOE must ensure that the TOE is delivered, installed, managed, and operated in a manner which maintains IT security objectives.

OE.OS_OPERATE The operating systems of the machines running the TOE must be configured and maintained such that the underlying systems provide a reliable basis for the operation of the TOE software. The operating systems are configured such that no unauthorized access to functions provided by the operating system software (including network daemons) is possible either locally or via any network connection.

OE.PHYSICAL Those responsible for the TOE must ensure that those parts of the TOE critical to security policy are protected from physical attack that might compromise IT security objectives.

OE.REPLICAS In case LDAP replicas are used in the IT environment, those responsible for the TOE must ensure that the LDAP replication mechanism – in particular with respect to the availability of consistent TSF data across all replicas – conforms with a well-defined policy

OE.SEC_INTEGRATE Those responsible for the TOE must ensure that the TOE is integrated into the overall system in a way that prohibits direct access to resources to be protected by the TOE in a way that bypasses the TOE and its security functions.

This includes that the communication between remote components of the TOE and supporting infrastructure in the IT environment (i.e. with the LDAP repository) is protected against unauthorized interception, either by organizational/physical means (for example, a dedicated network) or by logical means (for example, SSL/TLS).

This includes further that those responsible for the TOE shall seek assurance that the underlying system of a resource manager works as specified.

OE.USER Those responsible for the TOE shall control the user community that can request access to resources protected by the TOE. This includes a configuration where the client systems allowed to submit requests to the TOE are controlled (for example, a company internal network with a known and controlled user community protected against unauthorized access from external networks).

5 IT Security Requirements

This chapter defines the security requirements for the TOE as well as the TOE environment.

Chapter 5.1 defines the security requirements for the TOE itself, separated into security functional requirements and security assurance requirements. Those requirements use the appropriate Common Criteria functional and assurance components with all the required operations performed. Operations are marked in bold and italics. In addition some refinements of SFRs as defined in the Common Criteria had to be made. Those are marked in bold, italics and underlined. Iterations are marked by appending an additional identifier to the SFR reference.

Chapter 5.2 defines the security requirements for the IT environment, separate for each component within the environment. Here only security functional requirements are defined using the Common Criteria functional components where appropriate. Not all operations have been performed for those components to allow for the necessary flexibility in the selection of the products used in the environment. The security functional requirements defined in this section try to identify a minimum set of requirements needed to get a secure Tivoli Access Manager Environment.

Chapter 5.3 then defines the security requirements for the non-IT environment. They are expressed without using the Common Criteria functional components because they are not suitable to describe non-technical security requirements

Note that in the definition of the SFRs the terms (a) "**users**" and (b) "**administrators**" are used to distinguish between (a) all users of the managed resource, including TAMOS RM administrators and (b) Base administrators only, if not noted otherwise. This reflects the actual technical view based on which the roles known to the TOE are implemented.

5.1 Extended Components Definition

5.1.1 FAU_GEN.3-TAMOS

Rationale: This explicitly statement requirement is introduced to reflect the audit capabilities of the TAMOS RM which could not be expressed using functional requirements from CC part 2. Since this explicitly stated requirement is derived from FAU_GEN.1, the assurance requirements applicable to FAU_GEN.1 apply FAU_GEN.3-TAMOS as well.

Component leveling: FAU_GEN.3-TAMOS is not hierarchical to FAU_GEN.1 or FAU_GEN.2.

Management: FAU_GEN.3-TAMOS – No management activities are foreseen.

Audit: No actions are identified.

FAU_GEN.3-TAMOS Audit data generation

FAU_GEN.3-TAMOS.1 The TSF shall be able to generate an audit record of the following auditable events:

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

b) [assignment: other specifically defined auditable events]

FAU_GEN.3-TAMOS.2 The TSF shall record within each audit record at least the following information:

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

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

Dependencies: FPT_STM.1

5.2 TOE Security Requirements

5.2.1 TOE Security Functional Requirements

FAU_GEN.1 Audit data generation

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

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

b) All auditable events for the ***not specified*** level of audit; and

c) ***The following defined events:***

a. Policy Server:

- ***creation of user by administrator***
- ***user locked by administrator***
- ***user unlocked by administrator***
- ***all commands of administrators that result in a modification of the policy database***
- ***locking of User ID (after three consecutive unsuccessful authentication attempts)***

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 ***for Base administration commands: parameters passed to the command.***

Application note: The “not specified level of audit” refers to the fact that no other audit records than the ones specified here are subject to evaluation.

FAU_GEN.2 User identity association

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

FAU_GEN.3-TAMOS Audit data generation

FAU_GEN.3-TAMOS.1 The TSF shall be able to generate an audit record of the following auditable events:

a) All auditable events for the ***not specified*** level of audit; and

b) ***The following defined events:***

a. TAMOS RM:

- ***successful and unsuccessful authentication attempts with a userid / password combination***
- ***failed authorization for access to a protected resource***
- ***locking of User ID (after three unsuccessful authentication attempts within a defined time interval)***
- ***unlocking of User ID***
- ***refresh of a user's credentials***

FAU_GEN.3-TAMOS.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 ***not specified.***

Application note: The "not specified level of audit" refers to the fact that no other audit records than the ones specified here are subject to evaluation.

FAU_SAR.1(1) Audit review (Policy Server)

FAU_SAR.1.1 The TSF shall provide ***authorized administrators*** with the capability to read ***all 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.

Application Note: The TOE does not provide a direct interface to read the audit trail. Instead the administrator has to use a tool outside of the TOE to read the audit records. The information in the audit files is human readable even when read with a program like an editor.

FAU_SAR.1(2) Audit review (TAMOS RM)

FAU_SAR.1.1 The TSF shall provide ***authorized users*** with the capability to read ***all 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.

Application Note: Audit records are stored in a binary log file format and tools are provided to extract human-readable information from the log files.

FAU_SEL.1(1) Selective audit (Policy Server)

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) object identity, host identity

b) audit event category (authn, azn, mgmt)

Application Note: The audit categories can be defined on a per-server basis, which satisfies the selection of "host identity" in a). POP can be used to define auditing on a per object basis.

FAU_SEL.1(2) Selective audit (TAMOS RM)

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) object identity, subject identity

b) event outcome (for objects: permit, deny, all, none; for users: permit, deny, loginpermit, logindeny, all, none)

FAU_STG.1 Protected audit trail storage

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

FAU_STG.1.2 The TSF shall be able to ***prevent*** modifications to the audit records.

Application Note: The protection from unauthorized deletion is achieved with the TOE setting the access permissions to the audit files appropriately. Prevention of modifications also is based on the access rights to the audit files and by the fact that the TOE itself does not provide any function that could be used to modify the audit records once stored in the audit file. This security functional requirement of course also relies on the appropriate protection of the TOE itself against unauthorized access in the TOE environment.

FCS_CKM.1(1) Cryptographic key generation (Symmetric algorithms)

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm ***as defined in the SSL v3 standard*** and specified cryptographic key sizes ***128 bit (RC4), 168 bit (TDES)*** that meet the following: ***generation and exchange of session keys as defined in the SSL v3 standard with the cipher suites defined in FCS_COP.1(2).***

Application Note: Generation of symmetric keys is defined in section 6.2 in the SSL v3 standard.

FCS_CKM.1(2) Cryptographic key generation (RSA)

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm ***product specific*** and specified cryptographic key sizes ***1024 bit*** that meet the following: ***not specified***

Application Note: The SSL v3 specification does not define how the RSA key pair is generated. This is up to the implementation. Almost all implementations of the SSL v3 standard have their own algorithm for RSA key pair generation (if they support cipher

suites that use RSA). Therefore the key generation and algorithm and the standard to follow are not defined here. Only the required key size is specified.

FCS_CKM.2(1) Cryptographic key distribution (RSA public keys)

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method ***digital certificates*** that meets the following: ***certificate format as defined in X.509 Version 3***.

Application Note: This requirement addresses the exchange of public RSA keys as part of the SSL client and server authentication. For a definition of the certificate format see [X.509].

FCS_CKM.2(2) Cryptographic key distribution (Symmetric keys)

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method ***Secure Socket Layer handshake using RSA encrypted exchange of session keys*** that meets the following: ***SSL Version 3 (Internet Draft dated November 1996, Netscape Communication)***.

Application Note: This requirement addresses the exchange of SSL session keys as part of the SSL handshake protocol.

FCS_COP.1(1) Cryptographic operation (RSA)

FCS_COP.1.1 The TSF shall perform ***digital signature generation and digital signature verification*** in accordance with a specified cryptographic algorithm ***RSA*** and cryptographic key sizes ***1024 bit*** that meet the following: ***SSL Version 3 (Internet Draft dated November 1996, Netscape Communication)***.

Application Note: This requirement addresses the RSA digital signature generation and verification operations using the RSA algorithm as required by the SSL session establishment protocol (provided a cipher suite including RSA is used). Note that the details of the signature format like the use of the PKCS#1 block type 1 and block type 2 are defined in the SSL Version 3 standard.

FCS_COP.1(2) Cryptographic operation (Symmetric operations)

FCS_COP.1.1 The TSF shall perform ***encryption and decryption*** in accordance with a specified cryptographic algorithm ***RC4 and TDES*** and cryptographic key sizes ***128 bit (RC4) or 168 bit (TDES)*** that meet the following: ***SSL Version 3 (Internet Draft dated November 1996, Netscape Communication) and the following cipher suites: SSL_RSA_WITH_RC4_128_SHA, SSL_RSA_WITH_RC4_128_MD5, SSL_RSA_WITH_3DES_EDE_CBC_SHA as defined in the SSL v3 standard.***

Application Note: GSKit supports also other cipher suites that use RSA based key exchange listed in the SSLv3 standard. The cipher suites listed above are the ones supported in the evaluated configuration.

FDP_ACC.2(1) Complete access control

FDP_ACC.2.1 The TSF shall enforce the ***Object-Space access control policy*** on ***us-***

ers as subjects and objects in the TAMOS RM protected object space and all operations among subjects and objects covered by the SFP.

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

FDP_ACC.2(2) Complete access control

FDP_ACC.2.1 The TSF shall enforce the ***management access control policy*** on ***administrators as subjects and objects in the management protected object space*** and all operations among subjects and objects covered by the SFP.

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

FDP_ACF.1(1) Security attribute based access control

FDP_ACF.1.1 The TSF shall enforce the ***security attribute based Object-Space access control policy*** to objects based on the following:

Users as subjects and objects in the TAMOS RM protected object space controlled by access control lists (ACL), protected object policies (POP) and Access Restriction Extended Attributes Policy (AREAP).

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

Users have the requested type of access to a protected object in the Object-Space under the following conditions:

1. ***the user has been successfully authenticated and identified and the user has the "traverse" right for all objects from the root object down the path to the requested object and***
    - ***the user has an entry in the ACL associated with the object that contains the requested type of access, or***
    - ***the user is member of a group that has an entry in the ACL associated with the object that contains the requested type of access, or***
    - ***the ACL associated with the object has an entry of type "any-other" that contains the requested type of access***
2. ***the user has been successfully authenticated but unsuccessfully identified and a traverse right exists for all objects from the root object down the path to the requested object for unauthenticated users and***
    - ***the ACL associated with the object has both an entry of type "any-other" and an entry of type "unauthenticated" where the requested access right is contained in both entries***

The "ACL associated with the object" is the ACL of the object if the object has an explicit ACL or the ACL inherited from the next object

up on the path to the root that has an explicit ACL.

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: ***none***

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the

1. ***rules defined by the Protected Object Policy, if such a Protected Object Policy has been defined for the requested object. Protected Object Policies can deny access based on***
   - ***the time-of-day***
2. ***rules defined by the Access Restriction Extended Attributes Policy, if such an Access Restriction Extended Attributes Policy has been defined for the requested object. Access Restriction Extended Attributes Policy can deny access based on***
   - ***the accessor***
   - ***the permission set***
   - ***the program used***

FDP_ACF.1(2) Security attribute based access control

FDP_ACF.1.1 The TSF shall enforce the security attribute based ***management access control policy*** to objects based on the following:

Administrators as subjects and objects in the management protected objects space controlled by access control lists (ACL) and protected object policies (POP).

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

Users have the requested type of access to a protected object in the Object-Space under the following conditions:

1. ***the user has been successfully authenticated and***
   - ***the user has the "traverse" right for all objects from the root object down the path to the requested object and***
   - ***the user has an entry in the ACL associated with the object that contains the requested type of access, or***
   - ***the user is member of a group that has an entry in the ACL associated with the object that contains the requested type of access, or***
   - ***the ACL associated with the object has an entry of type "any-other" that contains the requested type of access***
2. ***the user has not been authenticated and***
   - ***a traverse right exist for all objects from the root object down the path to the requested object for unauthenticated users and***
   - ***the ACL associated with the object has both an entry of type***

"any-other" and an entry of type "unauthenticated" where the requested access right is contained in both entries

The "ACL associated with the object" is the ACL of the object if the object has an explicit ACL or the ACL inherited from the next object up on the path to the root that has an explicit ACL.

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: ***none***

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the

Rules defined by the Protected Object Policy, if a Protected Object Policy has been defined for the requested object. Protected Object Policies can deny access based on

- ***the time-of-day***

FIA_AFL.1(1) Authentication failure handling (Policy Server)

FIA_AFL.1.1 The TSF shall detect when ***three*** unsuccessful authentication attempts occur related to ***password based authentication attempts of administrators***.

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall ***disable further login attempts of that user for a time interval of not less than 180 seconds as defined by the administrator in the disable-time-interval configuration parameter***.

FIA_AFL.1(2) Authentication failure handling (TAMOS RM)

FIA_AFL.1.1 The TSF shall detect when ***three*** unsuccessful authentication attempts occur related to ***password based authentication attempts of users***.

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall ***disable further login attempts of that user for a time interval of not less than 180 seconds as defined by the administrator in the Login-LockMinutes configuration parameter in the /OSSEAL/policy-branch/Login policy space***.

FIA_ATD.1(1) User attribute definition

FIA_ATD.1.2 The TSF shall maintain the following list of security attributes belonging to individual users: ***user identifier, registry identifier (distinguished name), list of groups the user belongs to.***

Application Note: The user's common name and surname are not security attributes. The user attributes are stored in an external LDAP server.

FIA_ATD.1(2) *Administrator* attribute definition

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual administrators: ***user name, registry identifier (distinguished name), password, list of groups the user belongs to.***

Application Note: The term "user" in the CC SFR FIA_ATD.1 has been refined by "administrator" to

differentiate the attribute definition of users and administrators. The administrator's common name and surname are not seen as security attributes. The administrator attributes are stored in an external LDAP server.

FIA_SOS.1 Verification of secrets

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

1. ***Minimum password length is 8 characters***
2. ***Minimum number of alphabetic characters is 4***
3. ***Minimum number of non-alphabetic character is 1***
4. ***Maximum number of repeated characters is 2***

Application Note: Those parameters are configurable by the administrator.

Application Note: This requirement is valid for both the Policy Server and TAMOS RM enforced authentication policy.

FIA_UAU.2 *Administrator* authentication before any action (Policy Server)

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

Application Note: The term "user" in the CC SFR FIA_UAU.2 has been refined by "administrator" to differentiate the authentication policy of users and administrators. While there is a possibility of unauthenticated users, all administrators (which use a different interface for authentication) are required to authenticate successfully before performing any administrative action on the TOE.

FIA_UID.1 Timing of identification

FIA_UID.1.1 The TSF shall allow ***access as defined in the bitwise 'and' of the 'any-other' and 'unauthenticated' entry in an object's ACL on behalf of the user*** to be performed before the user is identified.

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

Application Note: An ACL may contain an entry that defines the access modes allowed for anonymous users, i.e. users that are not identified and authenticated.

FIA_UID.2 *Administrator* identification before any action (Policy Server)

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

Application Note: The term "user" in the CC SFR FIA_UID.2 has been refined by "administrator" to differentiate the authentication policy of users and administrators. While there is a possibility of unauthenticated users, all administrators (which use a different interface for authentication) are required to authenticate successfully before performing any administrative action on the TOE.

FIA_USB.1 User-subject binding

FIA_USB.1.1 The TSF shall associate the ***following*** user security attributes with subjects acting on behalf of that user:

1. ***the user identity associated with auditable events;***
2. ***the user and group identities used to enforce the security attribute based Object-Space access control policy and the management access control policy.***

Application Note: The TSS specifies the security attributes that are of relevance for the TSF and how they are associated with users.

FMT_MOF.1 Management of security functions behavior

FMT_MOF.1.1 The TSF shall restrict the ability to modify the behavior of the functions authentication, audit, and authorization to authorized administrators.

FMT_MSA.1 (1) Management of security attributes

FMT_MSA.1.1 The TSF shall enforce the ***management access control policy*** to restrict the ability to ***modify or delete*** the security attributes ***ACL entries*** to ***users or groups having 'control' access for the ACL***.

FMT_MSA.1 (2) Management of security attributes

FMT_MSA.1.1 The TSF shall enforce the ***management access control policy*** to restrict the ability to ***change_default, modify or delete*** the security attributes ***ACL, POP and Access Restriction Extended Attributes*** to ***authorized administrators***.

FMT_MSA.2 Secure security attributes

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

FMT_MSA.3 Static attribute initialization

FMT_MSA.3.1 The TSF shall enforce the ***management access control policy and Object-Space access control policy*** to provide ***inherited*** default values for security attributes that are used to enforce the *SFP*.

FMT_MSA.3.2 The TSF shall allow the ***administrator authorized to modify the ACL of the container object*** to specify alternative initial values to override the default values when an object or information is created.

Application Note: If no ACL is attached to an object, this object inherits the ACL attached to container object that contains the object. This inheritance rule goes 'upward' in the protect object space tree until a container object with an ACL is reached. This rule is expressed with this requirement.

FMT_MTD.1 Management of TSF data

FMT_MTD.1.1 The TSF shall restrict the ability to ***modify and delete*** the ***user attribute data*** to ***authorized administrators***.

FMT_SMF.1 Specification of Management Functions

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

1. ***User and group management***
2. ***ACL, POP and Access Restriction Extended Attributes management***
3. ***Audit management***
4. ***TOE certificate management***
5. ***Login policy management***
6. ***Password management***

Application Note: Management of attributes for 1), 4), 5) and 6) makes use of an external LDAP server.

FMT_SMR.1 Security roles

FMT_SMR.1.1 The TSF shall maintain the roles ***users and administrators***.

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

Application Note: Administration tasks can be delegated by the initially defined administrator (secmaster) to other administrators that he has created. The tasks a specific administrator is allowed to perform can be defined on a fine-grained basis as described in chapter 6. The term 'administrator' is used in this Security Target for any administrator that has been defined to perform administrative actions via the pdadmin interface.

FPT_ITT.1 Basic internal TSF data transfer protection

FPT_ITT.1.1 The TSF shall protect TSF data from ***disclosure and modification*** when it is transmitted between separate parts of the TOE.

FPT_RVM.1 Non-bypassability of the TSP

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

FPT_TRC.1 Internal TSF consistency

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

FPT_TRC.1.2 When parts of the TOE containing replicated TSF data are disconnected, the TSF shall ensure the consistency of the replicated TSF data upon reconnection before processing any requests for ***access permission***.

Application Note: The documentation describes the process where a notification can be sent out by the policy server to all authorization servers with replica of the master database. It is not described how the policy server checks that the authorization servers have received this notification (e. g. if they are not reachable at the time the notification has been sent out). As an alternative authorization servers can check for database updates by polling the policy server.

FTP_ITC.1 Inter-TSF trusted channel

FTP_ITC.1.1 The TSF shall provide a communication channel between itself and a remote trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

FTP_ITC.1.2 The TSF shall permit ***the TSF*** to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for ***the communication with the LDAP server*** .

Application Note: The trusted channel is established using the SSL v3 protocol with client authentication. In the case of the LDAP server the TOE acts as a client (as seen by the SSL protocol). The SSL v3 protocols define the client as the communication partner that initiates the communication over the trusted channel.

5.2.2 TOE Security Assurance Requirements

The target evaluation assurance level for the product is EAL3 [CC] augmented by ALC_FLR.1.

5.3 Security Requirements for the IT Environment

There are several components used in the IT environment of the TOE that need to satisfy a number of security requirements to ensure a secure operation of the TOE in its environment. Those security functional requirements are defined in this section, separate for each system within the TOE environment. The functional components of the Common Criteria (part 2) have been taken to describe those security functional requirements and the operations within those components have been performed whenever possible and useful. It should be noted that the security functional requirements defined here are a minimum set that need to be satisfied by those components to ensure the secure operation of Tivoli Access Manager within its intended environment. In most cases the components will have more security functions than defined here, allowing for a more flexible type of operation or even providing a higher level of security.

It should be noted that the security requirements for the components in the IT environment are not intended to be complete specifications of all security requirements of such a component. Only those requirements required by the TOE are defined.

5.3.1 LDAP Server

Summary and Justification

The TOE uses an external LDAP Server to store and maintain data related to the users and administrators of the TOE. The LDAP Server is required to protect this data against unauthorized access and disclosure. This requires functions for identification and authentication as well as access control for those directory entries belonging to the TOE. The TOE also uses the LDAP server to determine authentication decisions for Base administrators based on the user registry information stored in the server.

It is not required to have an LDAP Server dedicated to one specific instantiation of Tivoli Access Manager for Operating Systems. But it is required that the subset of the directory “belonging” to a specific instantiation of Tivoli Access Manager for Operating Systems is pro-

tected against unauthorized access of any kind by anybody other than a server belonging to the specific instantiation of Tivoli Access Manager. If one directory server is used for several instantiations of Tivoli Access Manager or other applications, the separation or sharing of directory entries has to be defined by the organization (which also includes a clarification of the question how to manage the directory in the case of shared directory entries).

The following section defines the minimum security functional requirements for the Directory Server using Common Criteria functional requirements components. Not all operations on the components have been performed. Some of them can not be performed without placing unnecessary restrictions on the Directory Server.

FDP_ACC.1 Subset access control

FDP_ACC.1.1 The ***IT environment*** shall enforce the ***Directory Access Control Policy*** on ***servers belonging to a TOE instantiation as subject, directory entries belonging to the TOE instantiation as objects and all operations that create, read, modify or delete those objects***.

FDP_ACF.1 Security attribute based access control

FDP_ACF.1.1 The ***IT environment*** shall enforce the ***Directory Access Control Policy*** to objects based on ***successful authentication of a server that is part of the TOE instantiation***.

FDP_ACF.1.2 The ***IT environment*** shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: ***If the server is successfully authenticated to belong to the TOE instantiation all access to directory records belonging to this instantiation of TOE is allowed***.

FDP_ACF.1.3 The ***IT environment*** shall explicitly authorize access of subjects to objects based on the following additional rules: ***None***.

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

Application Note: The assignment in FDP_ACF.1.4 has not been performed, since it is up to the local security policy of the Directory Server to define such additional restrictions. The security of Tivoli Access Manager for Operating Systems does not rely on such additional restriction but also does not demand that they should not exist.

FDP_ETC.1 Export of user data without security attributes

FDP_ETC.1.1 The ***IT environment*** shall enforce the ***Directory Access Control Policy*** when exporting user data, controlled under the SFP(s), outside of the TSC.

FDP_ETC.1.2 The ***IT environment*** shall export the user data without the user data's associated security attributes.

FDP_ITC.1 Import of user data without security attributes

FDP_ITC.1.1 The ***IT environment*** shall enforce the ***Directory Access Control Policy*** when importing user data, controlled under the SFP, from outside of the TSC.

FDP_ITC.1.2 The ***IT environment*** shall ignore any security attributes associated with the user data when imported from outside the TSC.

FDP_ITC.1.3 The ***IT environment*** shall enforce the following rules when importing user data controlled under the SFP from outside the TSC: [assignment: *additional importation control rules*].

Application Note: The assignment in FDP_ITC.1.3 has not been performed, since it is up to the local security policy of the Directory Server to define those rules. The security of Tivoli Access Manager for Operating Systems does not rely on those rules.

FIA_UAU.2 User authentication before any action

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

FIA_UID.2 User identification before any action

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

FMT_MSA.1 Management of security attributes

FMT_MSA.1.1 The ***IT environment*** shall enforce the ***Directory Access Control Policy*** to restrict the ability to [selection: change_default, query, modify, delete, [assignment: other operations]] the security attributes [assignment: list of security attributes] to [assignment: the authorized identified roles].

Application Note: The Directory Server shall enforce an access control policy on directory entries, but the details of this policy including the rights of the individual roles have to be defined in a Security Target for the Directory Server. This TOE does not define specific requirements for the capability of this policy and the role model of the Directory Server.

FMT_MSA.3 Static attribute initialization

FMT_MSA.3.1 The ***IT environment*** shall enforce the ***Directory Access Control Policy*** to provide [selection: restrictive, permissive, other property] default values for security attributes that are used to enforce the SFP.

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

FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1 The ***IT environment*** shall be capable of performing the following security management functions: [assignment: list of security management functions to be provided by the TSF].

FMT_SMR.1 Security roles

FMT_SMR.1.1 The ***IT environment*** shall maintain the roles [assignment: the authorized identified roles].

FMT_SMR.1.2 The ***IT environment*** shall be able to associate users with roles.

5.3.2 Underlying Operating System of the TOE components (Policy Server)

The underlying operating system for the Policy Server is required to provide a reliable time stamp used for the generation of the date and time in the audit records generated by the TOE. In addition the TOE makes use of a configuration file containing the types of audit events that the TOE shall generate audit records for. This file needs to be edited by an OS administrator and should be protected from unauthorized access and modifications by any other user (if such a user is installed in the underlying OS). The TOE makes no use of any other security function provided by the underlying operating system for the Policy Server.

FIA_UID.1 Timing of identification

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

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

FMT_MTD.1 Management of TSF data

FMT_MTD.1.1 The ***IT environment*** shall restrict the ability to modify the list of events to be audited to ***OS administrators***.

FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1 The ***IT environment*** shall be capable of performing the following security management functions: [assignment: list of security management functions to be provided by the TSF].

FMT_SMR.1 Security roles

FMT_SMR.1.1 The ***IT environment*** shall maintain the roles ***OS administrators and*** [assignment: other authorized identified roles].

FMT_SMR.1.2 The ***IT environment*** shall be able to associate users with roles.

FPT_SEP.1 TSF domain separation

FPT_SEP.1.1 The ***IT environment*** shall maintain a security domain for ***the TOE's*** execution that protects it from interference and tampering by untrusted subjects.

FPT_SEP.1.2 The ***IT environment*** shall enforce separation between the security domains of subjects in the TSC.

FPT_STM.1 Reliable time stamps

FPT_STM.1.1 The ***IT environment*** shall be able to provide reliable time stamps for ***the TOE's*** use.

5.3.3 Underlying Operating System of the TOE components (TAMOS RM)

The underlying operating system for the TAMOS RM is required to provide a reliable time stamp used for the generation of the date and time in the audit records generated by the TOE. In addition the TOE makes use of a configuration file containing the types of audit events that the TOE shall generate audit records for. Domain separation is not only necessary to enforce the underlying system's security mechanisms, but also to support the TOE by allowing it to operate within a protected security domain. Access control mechanisms in the operating system, e.g. for shared memory and other IPC mechanisms, further support the protection of TOE operations.

The underlying operating system plays a key role in the user authentication and identification process. It must provide appropriate measures to fulfill these tasks.

FDP_ACC.1 Subset access control

FDP_ACC.1.1 The ***IT environment*** shall enforce the ***Operating System Policy*** on ***users of the operating system as subjects, file system and IPC objects, and all transactions operating on these objects***.

FDP_ACF.1 Security attribute based access control

FDP_ACF.1.1 The ***IT environment*** shall enforce the ***Operating System Policy*** to objects based on [assignment: security attributes, named groups of security attributes].

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

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

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

Application Note: The rules must allow the administrator of the underlying system to prevent any unauthorized access to files containing TSF or user data belonging to the TOE.

FIA_ATD.1 User attribute definition

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

FIA_UAU.1 Timing of authentication

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

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

FIA_UID.1 Timing of identification

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

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

FIA_USB.1 User-subject binding

FIA_USB.1.1 The ***IT environment*** shall associate the appropriate user security attributes with subjects acting on behalf of that user.

FMT_MSA.1 Management of security attributes

FMT_MSA.1.1 The ***IT environment*** shall enforce the ***Operating System Policy*** to restrict the ability to [selection: change_default, query, modify, delete, [assignment: other operations]] the security attributes [assignment: list of security attributes] to [assignment: the authorized identified roles].

FMT_MSA.3 Static attribute initialization

FMT_MSA.3.1 The ***IT environment*** shall enforce the ***Operating System Policy*** to provide [selection: restrictive, permissive, other property] default values for security attributes that are used to enforce the SFP.

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

FMT_MTD.1 Management of TSF data

FMT_MTD.1.1 The ***IT environment*** shall restrict the ability to modify the list of events to be audited to ***OS administrators***.

FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1 The ***IT environment*** shall be capable of performing the following security management functions: [assignment: list of security management functions to be provided by the TSF].

FMT_SMR.1 Security roles

FMT_SMR.1.1 The ***IT environment*** shall maintain the roles ***OS administrators and*** [assignment: other authorized identified roles].

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

FPT_SEP.1 TSF domain separation

FPT_SEP.1.1 The ***IT environment*** shall maintain a security domain for ***the TOE's*** execution that protects it from interference and tampering by untrusted subjects.

FPT_SEP.1.2 The ***IT environment*** shall enforce separation between the security do-

mains of subjects in the TSC.

Application note: The underlying operating system provides a separation mechanism (kernel or supervisor mode vs. user mode) to prevent unprivileged operations from interfering with the TOE security functions.

FPT_STM.1 Reliable time stamps

FPT_STM.1.1 The ***IT environment*** shall be able to provide reliable time stamps for ***the TOE's*** use.

5.4 Strength of Function (SOF) Claim

The overall SOF claim for this Security Target is SOF-medium. No specific metric is defined and used for any permutational or probabilistic mechanisms.

A claim of SOF-medium is applied to the following security functional requirements:

- FIA_AFL.1(1)
- FIA_AFL.1(2)
- FIA_SOS.1

Please note that the cryptographic algorithms including the cryptographic hash algorithms as well as the key generation process for the keys used in those algorithms are excluded from the strength of function analysis in this evaluation.

6 TOE Summary Specification

6.1 Statement of TOE Security Functions

6.1.1 F.Audit

The TOE components can be individually configured with respect to the audit functions they perform. This is done using a defined configuration mechanism, which defines the type of events to be collected.

The TOE provides the capability to generate audit records for the following events:

1. TAMOS RM
   a. Authentication attempts (successful and unsuccessful).
   b. Authorization failures.
   c. Locking of User ID (after three unsuccessful authentication attempts within a defined time interval).
   d. Unlocking of User ID.
   e. Refresh of User's credentials.
2. Policy Server (pdmgrd):
   a. New user created.
   b. User locked by administrator.
   c. User unlocked by administrator.
   d. Locking of User ID (after three consecutive unsuccessful authentication attempts).
   e. All administrator actions that result in modifications to the policy database.

Each audit event is recorded with the date and time, the identity of the user that caused the event, the type of event and the success or failure. In the case of the Policy Server also all parameters of commands issued by an administrator are audited together with the command.

The Policy Server and TAMOS RM provide functionality to roll over to a new audit file if the current file exceeds an administrator-defined size.

6.1.1.1 Review of Audit Records

The *pdosaudview* command on the TAMOS RM is used to process a collection file generated by the TAMOS RM audit subsystem. The resulting output can be viewed, printed, or analyzed by scripts and other programs. The following parameters can be used to select the audit records in the produced output:

- Action
- Reason
- Outcome
- Accessor name

- Accessor effective name
- Decision type of the aznAPI
- Originating process
- Resource type
- Start- and end-time

Policy Server audit records are written in XML-formatted audit trails that can be reviewed by administrators. In addition, the Access Manager product provides an XML log viewer tool as part of the TOE environment.

6.1.2 F.Authentication

The TOE is capable of authenticating Base administrators by making use of an external LDAP server. Successful authentication is required before administrators can perform any administrative action.

The user authentication process can be extended by applying login and password policies.

6.1.2.1 User Authentication Policies

“Users” are those entities that attempt to access resources via the operating system services. In terms of authentication, this includes the TAMOS RM Administrators, but not Base administrators.

User authentication is performed by the native operating systems in the IT environment. The TOE can be configured to apply an additional login policy.

User Login Policy

The TOE supports policy that verifies a defined set of attributes against a policy's representation of these attributes stored in the user registry.

The TOE uses the user ID provided by the authentication process to decide which authentication policy to apply.

The TOE allows the following attributes to influence the authentication policy:

- A limit of successive failed login attempts. Note: This value is not maintained in the user registry but locally on the Resource Manager instance that performs the authentication process – there is no global counter for failed login attempts at multiple resource managers.
  The required value for this attribute is 3.
- Time-of-day login restrictions define hours of the day and days of the week during which users are permitted to log in. For users defined in the user registry, any user-specific policy overrides any global policy.
- Holiday login restrictions specify additional time-of-day restrictions by defining Holidays. The ability of a user to log in on the holiday is controlled by the ACL attached to the same resource.
- Login location restrictions specify where users can log in. Protected resources are defined under the Terminal branch of the Login resource hierarchy to specify where users can log in. Login locations are referred to as terminals.
- Login activity policy – the TOE provides the ability to define and enforce policy related to

login activity. The TOE login activity policy is applied in addition to any such policy provided natively by the operating system. The more restrictive between the TAMOS RM policy and the operating system policy will apply.

- User exception policy allows to define exceptions to the default login activity policy.
- Inactivity of user
  The required value for this attribute is 256 days.

In addition, the TAMOS RM can apply a password management policy enforcing the following properties:

- Password age
  The required value for this attribute is 186 days.
- Password strength
  Administrator defined parameters such as a minimum length for passwords.

The policies can be applied globally or on a per-user basis. The per-user policy takes precedence over the global policy.

Refer to section 6.1.4.5 and 6.1.4.6 for a description of the management of the policies.

User-subject binding

To acquire the credentials needed to make an authorization decision, the accessing user's native numerical UNIX ID is mapped to a TAMOS RM user. The user's UNIX username is obtained from the system's native user registry; this username is mapped one-to-one to a TAMOS RM user of the same name to retrieve credentials from the user registry. These credentials define the user's identity and group membership. If there is no TAMOS RM user corresponding to the user's native username, then the user is treated as unauthenticated when making authorization decisions. Users with disabled accounts are also treated as unauthenticated.

The TOE associates the following user security attributes with subjects acting on behalf of users:

1. The user identity which is associated with auditable events;
2. The user identity or identities which are used to enforce the Object-Space Policy;
3. The group membership or memberships used to enforce the Object-Space Policy;

Upon successful identification, the TAMOS RM user name is the one specified in the user entry for the user that has authenticated successfully. If a user cannot be identified, the user is considered an "unauthenticated user".

The TOE restricts the use of the underlying operating system's switch user/surrogate capabilities. Users can change their effective user ID or group ID if they have permissions on the Surrogate/User/username or Surrogate/Group/groupname objects or if they have permission on the .../Sudo/sudo-command object and use the pdossudo command. See also sections 6.1.3.5.3 and 6.1.3.5.4.

6.1.2.2 Authentication of Administrators

Base Administrators are authenticated with user ID and password and with use of an external LDAP server. The TOE provides the pdadmin command line interface for administration tasks. To execute a single administration command the administrator uses the following pdadmin command structure:

pdadmin [-a *admin_user*] [-p *password*] *command*

where *admin_user* is replaced by the administrator's userid and *password* is replaced by the password of the administrator. As an alternative the administrator can specify the command without the password, the system will prompt the administrator for the correct password.

To execute a set of commands the administrator can create a file containing all the commands and then issue the command

pdadmin [-a *admin_user*] [-p *password*] *filename*

where *filename* is replaced by the name of the file containing the set of commands the administrator wants to be executed. As above he may omit the password from the command line in which case he is prompted by the system for the correct password.

A third alternative is to start an interactive administrative session and using the **login** command in the form

login –a *admin_user* –p <*password*>

Again the administrator may omit the password in which case he is prompted to enter the correct password. The interactive session is terminated with the **logout** command.

The TOE also has parameters an administrator can use to define the password policy. Those parameters are:

- The minimum length of a password (default: 8)
- The minimum number of alphabetic characters (default: 4)
- The minimum number of non-alphabetic characters (default: 1)
- The maximum number of repeated characters (default: 2)

In addition, the TOE offers to disable administrator accounts for an administrator-defined amount of time after an administrator-defined number of consecutive unsuccessful authentication attempts.

6.1.3 F.Authorization

6.1.3.1 Authorization General Model

The authorization model of the TOE is based on Access Control Lists (ACL), "Protected Object Policies" (POP) and Access Restriction Extended Attributes. The objects that are protected (the protected object space) are defined in a tree structure that maintains several types of objects:

Operating system objects, which represent an operating system resource:

- Files, including programs, directories, soft links, hard links and devices
- Network Connections
- Changing of operating system user ids (surrogate)

Tivoli Access Manager Management Objects, which represent the management objects that can be managed through the pdadmin interface.

Administrators can define Access Control List policies, "Protected Object Policies" and "Access Restriction Extended Attributes" that together build the set of rules the authorization evaluator subsystem checks to decide if a user can be given the requested type of access to an object within the protected object space.

The TOE uses the ACLs also to determine access to management objects within the protected object space. The semantics of those access modes for the different types of management objects are described in the following sections. Please note that other access modes than the ones described with the individual types of management objects have no effect.

TAMOS RM General Authorization Flow

TAMOS RM components operate in the user-level application space and also within the operating system kernel. Applications access system resources through system-provided APIs, which arrive in the TAMOS RM operating system kernel extension via the system call interface.

The primary function of the TAMOS RM operating system kernel extension is to intervene in accesses to resources that are subject to the authorization policy. The kernel extension uses the authorization daemon process, PDOSD, to obtain an authorization decision by applying named authorization policy mechanisms and then enforces that decision. If the policy permits access to the resource, the operation continues and is then subject to the native system's security. Otherwise the resource access is denied.

It should be noted, that TAMOS RM consistently applies this general authorization flow to TAMOS RM management commands. This yields that only authorized TAMOS RM users are able to use these commands.

6.1.3.2 Access Control Lists (ACL)

The protected object space is organized as a tree with a single root, addressed by a forward slash. The next level of hierarchy consists of the Operating Systems Objects (/OSSEAL), the Tivoli Access Manager Management Objects (/Management) and (eventually) the user-defined objects.

The leaves within the tree that defines the protected object space are actually the individual objects. All branches within the tree are called "container objects" since they represent the container for all the leaves within subtree defined by the "container object".

Within the Tivoli Access Manager Management Object space, the following categories exist in the next level of the tree:

- User management objects (/Management/Users)
- Group management objects (/Management/Groups)
- *Global Sign On (GSO) management objects (/Management/GSO)*
- Server management objects (/Management/Server)
- ACL policy objects (/Management/ACL)
- POP objects (/Management/POP)
- Configuration authorization control objects (/Management/Config)
- *Third-party authorization control objects (/Management/Action)*
- Authorization database replication control objects (/Management/Replica)
- *Domain management (/Management/Domain)*
- *Authorization rules management (/Management/Rules)*

(Categories marked in italics are not used in the TOE configuration)

An administrator can create a new object with the **pdadmin object** command by defining the fully qualified location within the protected object space (provided he has the required permission).

An administrator can define and modify Access Control Lists (ACL) for objects within the protected object space. An ACL consists of:

1. A Type, which can be either “user”, “group”, “any-other” or “unauthenticated”.
   The type identifies, if the ACL defines permissions for specific user(s), group(s), any authenticated user or unauthenticated users.
2. An ID, which defines the unique identifier for the user (if of type “user”) or group (if of type “group”). ACLs of the types “any-other” or “unauthenticated” do not have such an ID.
3. A set of permissions, that define the type of access (action) allowed with the ACL. The possible permissions are:

Action	Description
a	Attach
A	Add
b	Browse
B	Bypass POP
c	Control
d	Delete
g	Delegation
l	List Directory
m	Modify
N	Create
r	Read
s	Server Administration
t	Trace
T	Traverse
v	View
W	Password
x	Execute

For user and /OSSEAL objects, the semantics of those permissions is defined by the resource manager that uses the authorization evaluator and the access manager database for access decision. The semantics of those permissions within the TOE subsystem are defined

later in this chapter.

6.1.3.3 Administration of the Object Space

As mentioned all objects (i. e. representation of objects) in the overall protected object space build a tree structure with a single root. The tree itself is structured into different “object spaces”.

Objects within an object space can be created by an administrator that has the “m” (modify) permission for the object container where the object is created. Objects can be deleted by an administrator that has the “d” (delete) permission for the object container of the object.

The following access rights to objects are managed by pdmgrd, since they relate to object management activities that are not controlled by the resource manager (TAMOS RM):

- **b (browse):** Permission to browse objects and object spaces using the following administration commands: *objectspace list, object list, object listandshow*. Note: The command *object listandshow* requires the permission “v” in addition to “b”.
- **d (delete):** Permission to delete objects and object spaces using the following administration commands: *objectspace delete, object delete, object modify set name*. Note: The command *object modify set name* requires the permission “m” in addition to “d”.
- **m (modify):** Permission to create and modify objects and object spaces using the following administration commands: *objectspace create, object create, object modify*. Note: The command *object modify set name* requires the permission “d” in addition to “m”.
- **v (view):** Permission to show object values and attributes using the following administration commands: *object listandshow, object show*. Note: The command *object listandshow* requires the permission “b” in addition to “v”.

6.1.3.4 ACL Semantics for Management Objects

As mentioned above the TOE uses ACLs also to control access to its own management objects. The “container objects” (object spaces) that exist for TOE management objects have been identified in the previous section.

ACLs for management objects can be used to define the commands an administrator is allowed to use with a defined management object. This allows for flexible delegation of specific administrative tasks to specific administrators or administrator groups.

The following semantics for permissions exist for TOE management objects:

/Management/ACL Permissions :

- **d (delete):** Permission to delete the ACL policy with the *acl delete* command. Requires “c” permission also be become effective.
- **m (modify):** Permission to create a new ACL policy using the *acl create* command
- **v (view):** Permission to find, list and show ACLs using the *acl find*, *acl list* and *acl show* command

/Management/Action Permissions

This object defines the administrators allowed to manage custom actions. Permissions are:

- **d (delete):** Permission to delete an existing action or action group using the *action delete* and *action group delete* commands
- **m (modify):** Permission to create a new action or action group using the *action create*

and *action group create* commands

/Management/POP Permissions

The object defines the permissions of administrators to manage protected object policies (POP). Permissions are:

- **d (delete):** Permission to delete a POP using the *pop delete* command
- **m (modify):** Permission to create POPs and modify POP attributes using the *pop create* and *pop modify* commands.
- **v (view):** Permission to find and list POPs and show POP details using the *pop find*, *pop list* and *pop show* commands.
- **B (Bypass TOD):** Permission to overwrite the time-of-day POP attribute on an object using the *acl modify set attribute B* command.

/Management/Server Permissions

The object defines the permissions of administrators to perform server management tasks. Permissions are:

- **s (server):** Permission to replicate the authorization database using the *server replicate* command.
- **v (view):** Permission to list registered servers and display server properties using the *server list* and *server show* commands.
- **t (trace):** Permission to enable dynamic trace or statistics administration using the *server task server_name trace* and *server task server_name stats* command.

/Management/Config Permissions

The object defines the permissions of administrators to perform configuration management tasks. Permissions are:

- **m (modify):** Permission to define and modify the configuration using the *svrsslcfg –config* and *svrslcfg –modify* commands.
- **d (delete):** Permission to delete (deconfigure) the configuration using the *svrsslcfg –unconfig* command.

/Management/Policy Permissions

The object defines the permissions of administrators to perform *policy get* and *policy set* commands to define or retrieve the overall user related policy attributes (like password restrictions, etc). Permissions are:

- **v (view):** Permission to perform the *policy get* command.
- **m (modify):** Permission to perform the *policy set* command.

/Management/Replica Permissions

The object defines the permission of administrators to control the replication of the master authorization database. Permissions are:

- **v (view):** Permission to read the master authorization database

/Management/Users Permissions

The object defines the permissions of administrators to manage user accounts. Permissions are:

- **d (delete):** Permission to delete a user account using the *user delete* command.
- **m (modify):** Permission to modify a user account using the *user modify* command.
- **N (create):** Permission to create a user account using the *user create* and *user import* commands.
- **v (view):** Permission to view a user account and user account details using the *user list, user list-dn, user list-gsouser, user show, user show-dn* and *user show-groups* command.
- **W (password):** Permission to reset and validate a user password using the *user modify password* and *user modify password-valid* command.

/Management/Groups Permissions

The object defines the permissions of administrators to manage groups. Permissions are:

- **d (delete):** Permission to delete a group using the *group delete* command.
- **m (modify):** Permission to modify a group using the *group modify description* and *group modify remove* commands.
- **N (create):** Permission to create a group using the *group create* and *group import* commands.
- **v (view):** Permission to view a group definition using the *group list, group list-dn, user, group show, group show-dn* and *group show-members* command.
- **A (add):** Permission to a member to a group using the *group modify add* command.

Further details on the management of the TOE are defined in the description of the function F.Management.

6.1.3.5 ACL Semantics for Operating System Objects

The TAMOS RM policy is defined under the /OSSEAL root of the object name space. The next portion of the object name defines the branch name, /OSSEAL/branch-name. Following the branch name are the supported TAMOS RM policy types, /OSSEAL/branch-name/policy-type. The names for the ACL relevant policy types are:

- File
- NetIncoming
- NetOutgoing
- Login
- Surrogate
- Sudo

TAMOS RM permissions are defined by actions within the TOE policy database. An action defines a single-letter mnemonic representing the permission, the name of the permission, the kind of the resource it applies to, and the action group of which it is a part. An action group is a collection of related actions. All TAMOS RM actions are defined as members of the OSSEAL action group ([OSSEAL]). Actions in the OSSEAL action group represent operations that may be performed on the resources that TAMOS RM protects.

Action	Description	Resource Type

Action	Description	Resource Type
C	Connect	NetIncoming and NetOutgoing
D	Change directory	File
G	Surrogate	Surrogate
K	Kill program	File
L	Login	Login
N	Create	File
R	Rename	File
U	Update timestamp	File
d	Delete	File
l	List directory	File
o	Change ownership	File
p	Change permission	File
r	Read	File
w	Write	File
x	Execute	File and Sudo

6.1.3.5.1 Semantics of “NetOutgoing” and “NetIncoming”

TAMOS RM provides the ability to control access to remote network services from a local machine and also to control access to local network services from remote locations. These two types of network access are controlled separately by defined, protected resources of type NetOutgoing and NetIncoming, respectively. These resources are represented in the Tivoli Access Manager namespace as:

/OSSEAL/policy-branch/NetIncoming/protocol[/service[/host]]

/OSSEAL/policy-branch/NetOutgoing[/hostspec[/protocol[/service]]]

6.1.3.5.2 Semantics of “File”

TAMOS RM provides the ability to control access to file system resources. File systems resources consist of:

- Files
- Directories

- Soft links
- Hard links
- Device files

File system resources are represented in the TAMOS RM namespace by defining an object name with resource type File and specifying the name of the file system resource to be protected:

/OSSEAL/policy-branch/File/filespec

6.1.3.5.3 Semantics of “Surrogate”

TAMOS RM provides the ability to control operations that can change the UNIX identity of a process. Such operations are referred to as surrogate operations and are controlled by resources of type Surrogate. Surrogate operations can change the user identity or group identity of a process. Access control of each of these kinds of surrogate operations is established by applying authorization policy to the User and Group sub-types of the Surrogate resource type. The object names identify the potential targets of surrogate operations and control the ability, for example, to surrogate to the root user or the system group. Surrogate resource names follow the form:

/OSSEAL/policy-branch/Surrogate/User/user-name

/OSSEAL/policy-branch/Surrogate/Group/group-name

6.1.3.5.4 Semantics of “Sudo”

Sudo resources describe commands that require more stringent access control than whether or not a particular program can be executed. Sudo commands allow access control based not only on a command but also on the parameters passed to that command. You can use Sudo commands to remove the requirement for a user to become the root user on a system in order to perform administrative tasks. Sudo does this by providing the capability to execute a command as a UNIX user other than that of the invoker. Sudo resources are identified in the Tivoli Access Manager namespace in the following way:

/OSSEAL/policy-branch/Sudo/sudo-command[/sudo-argclass]

6.1.3.6 Protected Object Policies (POP)

Protected Object Policies contain additional conditions on the request that are passed back to the resource manager (in the case of the TOE: TAMOS RM) in the case the evaluation of the ACLs for the request was positive (i. e. according to the ACL policy request is granted). For those conditions of a POP it is the responsibility of the resource manager to enforce the conditions defined by the “Protected Object Policy”.

The following attributes can be set in a “Protected Object Policy”:

- **Warning Mode**. This attribute is used for debugging purpose mainly. Possible values are: “yes” and “no”. If set to “yes”, all decisions result in a “permit” being returned (no policies are enforced), and audit records are generated that capture the result of all ACL authorization decisions that would have been made if the warning mode would have been set to “no”.
- **Audit Level**. This attribute defines the level of audit for the object. Possible values are: “permit”, “deny” and “error”. (The audit level “error” is not supported by TAMOS RM.) In the case of “permit”, all requests on a protected object that result in successful access

are audited. In the case of “deny”, all requests on a protected object that result in denial of access are audited. In the case of “error”, all internally generated error messages resulting from the denial of access to the protected object are audited.

- **Time-of-Day**. This attribute defines the day and time conditions on the access to a protected object. This attribute is overwritten by the “B” (Bypass POP) ACL policy permission.
- **Authentication Strength**. (Not supported by TAMOS RM) This attribute can be used to define restrictions on the authentication method required to gain access to the protected object. This is useful, if access to the object requires a high grade of confidence in the correct authentication of the user. It is the task of the resource manager to ensure that the user has authenticated with required authentication method before granting access to the object.
- **Network-based Authentication**. (Not supported by TAMOS RM) This attribute allows to control access based on the IP address of the user. This can be used to prevent access to protected objects from specific IP addresses or range of IP addresses.
- **Quality of Protection**. (Not supported by TAMOS RM) This attribute allows to define the required level of protection for an object. Possible values are: “Privacy” and “Integrity”. In the case of “Privacy” the resource manager has to ensure that the object is transferred over an SSL encrypted communication link. In the case of “Integrity” the resource manager has to ensure that a mechanism for the protection of the integrity of the object is used when transferred.

6.1.3.7 Access Restriction Extended Attributes

TAMOS RM defines an extended attribute on an ACL which enables control over what programs users can use to perform particular actions. The name of the attribute is Access-Restrictions. Access to resources is controlled based on the identity of the user, the action that the user is performing, and the current program being used to perform the action. This restriction is in addition to the access control enforced by the base ACL that the attribute is associated with. Before an access restriction is applied the user must first have been granted access by the base ACL entries.

The format for an extended attribute Access-Restrictions entry is:

```
rule : accessor : permission-set : program-set
```

6.1.3.8 ACL Evaluation

ACLs may be either explicit or inherited. Any object without an explicit ACL inherits the ACL of the container object above in the object space tree. Note that this container object may also just have an inherited ACL. The root object must always have an ACL. A default ACL for this object is set at the TOE installation and initial configuration.

The TOE uses the following rule to determine if an authenticated user has the permission for the action requested for a defined object within the protected object space

(Note: When checking for the existence of an ACL for an object, it always means checking for an explicit or inherited ACL):

1. Check that the user has the traverse permission for all container objects on the path from the root container object down to the actual object. To check this, use the steps 2 to 4 of this algorithm for all container objects on the path and the “Traverse” (T) permission.
2. Check if an ACL entry of type “user” exists for the user and the object. If this is the case,

permission is granted if the requested action is defined in the ACL entry. The ACL evaluation algorithm stops if the permission is granted.

3. Check if ACL entries of type “group” exist for the groups the user belongs to and the object. If they exist, check if the requested permission is contained in at least one of those entries. If yes, the permission is granted and the evaluation algorithm stops.
4. Check if an ACL entry of type “any-other” exists for the object. If yes, check if the permission is granted within this ACL entry. If yes, permission is granted. Permission is denied, if it is not granted by the “any-other” ACL entry or if the ACL entry of type “any-other” does not exist for the object.

The TOE uses the following rule to determine if an unauthenticated user has the permission for the action requested for a defined object within the protected object space.

1. Check that unauthenticated users have the traverse permission for all container objects on the path from the root container object down to the actual object. To check this, use the steps 2 to 4 of this algorithm for all container objects on the path and the “Traverse” (T) permission.
2. Check if an ACL entry of type “unauthenticated” exists for the object. If no such ACL entry exists, access is denied and the evaluation algorithm stops.
3. Check if an ACL entry of type “any-other” exists for the object. If no such ACL entry exists, access is denied and the evaluation algorithm stops.
4. Check if the requested access is granted in both the ACL entries of type “unauthenticated” and in the ACL entries of type “any-other” for the object. Access is granted if the requested type of access is granted in both ACL entries. Otherwise access is denied.

As a result, a user has the requested access to an object if the two following conditions are satisfied:

1. The user has traverse permission for all container objects on the path from the root down to the object
2. The user has the requested permission being explicitly granted by the object’s ACL, which may be an explicit ACL or an inherited ACL.

6.1.3.9 Access Restriction Evaluation

If the user is authenticated, entries with accessor values of user, group, and any-other apply. Only the attribute entries whose permission set contains all of the permissions associated with the actions being performed against the protected resource are evaluated to see if they apply to the current access decision. For example, if read and write access is requested, then only entries with both the r and w permissions (and possibly others) are considered. Entries with a permission set value of * match all access requests regardless of the action being performed.

Entries with the accessor type user have the highest precedence during evaluation:

- if a deny rule entry is found with a user accessor type and the specified name matches the accessing user’s name and the user’s program or * is included in its program set, access is denied;
- if a permit rule entry is found with a user accessor type and the specified name matches the accessing user’s name and the user’s program or * is included in the entry’s program set, access is granted;
- if access has not been granted by other user accessor type entries and a permit rule en-

try is found with a user accessor type and the specified name matches the accessing user's name and the user's program or * is not included in the entry's program set; access is denied;

If no matching user accessor entries are found, entries with the group accessor type are evaluated:

- if a deny rule entry is found with a group accessor type and the specified name matches a group the accessing user is a member of and the user's program or * is included in its program set, access is denied;
- if a permit rule entry is found with a group accessor type and the specified name matches a group the accessing user is a member of and the user's program or * is included in its program set, access is granted;
- if access has not been granted by other group accessor type entries and a permit rule entry is found with a group accessor type and the specified name matches a group the accessor is a member of and the user's program or * is not included in the entry's program set; access is denied;

If no matching user or group accessor entries are found, entries with the any-other accessor type are evaluated:

- if a deny rule entry is found with an any-other accessor type and the user's program or * is included in its program set, access is denied;
- if a permit rule entry is found with an any-other accessor type and the user's program or * is included in its program set, access is granted;
- if access has not been granted by other any-other accessor type entries and if a permit rule entry is found with an any-other accessor type and the user's program or * is not included in the entry's program set, access is denied.

If the user is unauthenticated, only entries with accessor value unauthenticated apply. As with authenticated users, only the attribute entries whose permission set contains all of the permissions associated with the actions being performed against the protected resource are evaluated to see if they apply to the current access decision:

- if a deny rule entry is found with an unauthenticated accessor type and the user's program or * is included in its program set, access is denied;
- if a permit rule entry is found with an unauthenticated accessor type and the user's program or * is included in the entry's program set, access is granted;
- if access has not been granted by other unauthenticated accessor type entries and if a permit rule entry is found with an unauthenticated accessor type and the user's program or * is not included in the entry's program set; access is denied.

If no Access-Restrictions attribute entries apply that deny access, the access granted by the base ACL still applies and access is granted.

6.1.3.10 Dropping access credentials

Users can "drop" their access credentials and spawn an operating system shell that is therefore treated as "unauthenticated" by TAMOS by using the pdosunauth command.

6.1.4 F.Management

6.1.4.1 Tivoli Access Manager Administrators

At installation the TOE the group “iv-admin” is created with an initial administrator “sec_master” as its member. In addition a default ACL is defined for the “root” object in the protected object space. This default ACL for this object is:

Group iv-admin	TcmdbvaBR
Any-other	T
Unauthenticated	T

This default-root ACL allows everyone to traverse the object space (T) in order to pass through the object and gain access to protected resource objects further down in the hierarchy, while only members of the group iv-admin are allowed to perform the following actions: control as owners (c), modify (m), delete (d), browse (b), view (v), attach (a), override the POP policy (B) and override the authorization rule policy (R).

There are also default values for the different management object spaces, which are defined in the Base Administrator’s Guide.

The mechanisms described in F.Authorization allow the initial administrator to define other administrators and/or administration groups and assign them the right to perform only specific administration tasks. This is achieved by assigning them the appropriate permissions for the individual management object spaces and objects within those object spaces as well as the appropriate permissions to individual objects or object spaces within the overall user object space.

6.1.4.2 User and Group Management

Users are managed on the host operating system in combination with an external LDAP server holding the login policy and user credentials. An administrator with the appropriate permission in the */Management/User* object space can perform user management operations like creating users, deleting users or changing the user's login policy. The commands to create and manage user accounts are defined in the Command Reference. The required access rights to perform the commands are defined in section 6.1.3.4 under “/Management/Users”.

Groups are managed using the *pdadmin group* set of commands. The required access rights to perform the commands are defined in section 6.1.3.4 under “/Management/Groups”.

Users can be assigned to more than one group. Section 6.1.3.8 describes, how access rights to objects are evaluated which includes the evaluation of access rights in the case a user belongs to more than one group.

6.1.4.3 ACL, POP and Access Restriction Extended Attributes Management

ACLs are managed using the *pdadmin acl* set of commands defined in the Command Reference. The required access rights to perform the commands are defined in section 6.1.3.4 under “/Management/ACL”.

Protected Object Policies are managed using the *pdadmin pop* set of commands. The required access rights to perform the commands are defined in section 6.1.3.4 under “/Management/POP”.

Access Restriction Extended Attributes are managed using the *pdadmin acl* set of commands defined in the Command Reference. The required access rights to perform the commands are defined in section 6.1.3.4 under "/Management/ACL".

6.1.4.4 TOE Certificate Management

The TOE maintains its own Certification Authority and certificate management functions for the certificates it needs for authentication and key exchange between different servers that are part of the TOE (i. e. TOE internal communication). The single pdmgrd instance within the TOE also acts as the Certification Authority for the TOE internal public key infrastructure.

During installation of the TOE management subsystem (pdmgrd instance), two RSA key pairs (key size 1024 bit) need to be generated. One of those key pairs is used as the CA key to sign certificates (PDCA key), the other one is used for authentication when setting up a SSL v3 connection to another server within the TOE (pdmgrd-SSL key). The TOE management subsystem will then create a self-signed certificate for the PDCA public key and also sign the pdmgrd-SSL key using the PDCA private key.

Certificates and private keys of TOE components are held in special files ("keyring files"). Protection of those files against unauthorized access and modification is essential for the security of the TOE.

If a server's private key is compromised (or needs to be revoked for other reasons) the administrator can do this using the chgcert command. This will require creating a new key pair, generate a certificate for the public key and invalidate the old certificate (by removing it from the keyring files of all servers that are part of the TOE).

6.1.4.5 Login Policy Management

TAMOS RM lets the administrator control when and from where a user can log in to a system. The basic mechanisms for controlling user access are:

- Defining time-of-day login restrictions for users independent of where they log in from
- Defining access controls on local and remote terminals.

TAMOS RM also provides the ability to enforce login-activity-related policy such as password expiry, automatically disabling accounts after a number of failed logins, and automatically disabling inactive accounts.

Time-of-day login restrictions are defined by specific policy attributes in the user registry. They can be specified globally, on a per-user basis, or specifically for unauthenticated users. Time-of-day restrictions define hours of the day and days of the week during which users are permitted to log in. For users defined in the user registry, any user-specific policy overrides any global policy.

A time-of-day restriction is defined by a string of the following format:

day-range:time-range[:utc|local]

Holiday login restrictions specify additional time-of-day restrictions by defining Holidays. Holidays are protected resources that define exceptions to the regular time-of-day restrictions defined in the user registry. Holiday policy is applied when a user logs in.

The ability of a user to log in on the holiday is controlled by the ACL attached to the same resource. The Login (L) permission must be granted to those users allowed to log in. The format of the value of the Holiday-Dates extended attribute is a start time followed by an op-

tional space and an end time. The specified time format follows:

YYYY-MM-DD[-hh[:mm[:ss]]][Z]

The names of the holiday objects are are:

/OSSEAL/policy-branch/Login/Holiday/holidayname

Login location restrictions specify where users can log in. Protected resources are defined under the Terminal branch of the Login resource hierarchy to specify where users can log in. Login locations are referred to as terminals.

Local and remote terminals: Terminals are either local or remote. Terminals are local when used for logins to a system from serial devices and graphical consoles. Terminals are remote when used across a TCP/IP network. You can group both kinds of terminals together and use inheritance to define access controls. The names of terminal objects follow the format:

/OSSEAL/policy-branch/Login/Terminal/Local/termgroup/device

/OSSEAL/policy-branch/Login/Terminal/Remote/termgroup/hostspec

Login activity policy TAMOS RM provides the ability to define and enforce policy related to login activity. The policy is defined centrally by using extended attributes of the

/OSSEAL/policy-branch/Login

object and controls the following aspects of login activity:

- Password expiry
- Account suspensions due to failed login attempts
- Account lockouts due to account inactivity

The status of each user account is recorded on a per-machine basis. Accounts become locked or suspended only on the machine on which they have been active or on which failed login attempts have occurred. Password expiry times are maintained on a per-machine basis. TAMOS RM login activity policy is applied in addition to any such policy provided natively by the operating system. The more restrictive between the TAMOS RM policy and the operating system policy will apply.

User exception policy allows defining exceptions to the default login activity policy. This capability is provided strictly as a mechanism to define exceptions to the default policy and should not be used to define login activity policy for a large number of users. The user exception policy is defined by setting the login activity extended attributes on the

/OSSEAL/policy-branch/Login/UserExceptions/user-name

object. Only attributes that are explicitly set for this object apply to the user. Any login activity extended attribute not explicitly set is given a value of zero. These unspecified attributes do not inherit the value from the default login activity extended attributes.

6.1.4.6 Password Management

TAMOS RM provides the ability to define and enforce policy related to password management. Password management prevents users from specifying weak passwords that are vulnerable to compromise by methods such as a dictionary attack. The policy is defined centrally by using extended attributes of the:

/OSSEAL/policy-branch/Password

object and controls the following aspects of password management activity:

- Password strength
- Password aging

TAMOS RM password management policy is applied in addition to any such policy provided natively by the operating system. The more restrictive between the TAMOS RM and the operating system policy will apply. It is possible that the password will be modified by the native operating system prior to the TAMOS RM password enforcement modules seeing the password. If this happens, the password management policy will be applied to the modified password. For example, if the operating system truncates the password to the number of characters it considers significant, the password management policy is applied to the truncated password.

6.1.4.7 Audit management

The Policy Server provides a configuration file allowing defining parameters such as audit trail location and roll over functionality for those files.

For the configuration of auditable events related to transactions controlled by POPs, refer to section 6.1.3.6.

6.1.4.8 TAMOS RM Management

TAMOS RM provides management commands for the following aspects:

- pdoslpadm – unlocking (and locking) of locked users. The login policy's lockout aspect is immediately changed either allowing access (unlock) or denying it (lock).
- pdosrefresh – refresh the credentials for a user with information obtained from the user registry. The user's credentials are immediately replaced by information obtained from the user registry.
- pdosdestroy – destroys the cached credentials for a user, resulting in a refresh of the credentials at next access.
- pdosctl – manages several operational aspects, such as the generation of audit records and the start/stop of daemons
- pdoscfg - configures Tivoli Access Manager for Operating Systems
- pdoshla - manages the IP address to Host Name Lookaside Database
- pdosrgyimp - imports UNIX users and groups into the Tivoli Access Manager user registry
- pdosshowuser - shows various attributes of a specific user
- pdoswhoami - displays the Tivoli Access Manager for Operating Systems accessor ID information
- pdoswhois - displays Tivoli Access Manager for Operating Systems accessor ID information associated with the specified process Ids

It should be noted, that the execution of these commands is controlled by the centrally managed access control policy and access is therefore possible only for authorized users.

6.1.5 F.Communication

Communication between the TOE and the LDAP server are protected by SSL v3. The TOE acts as a client system for the communication with the LDAP server.

The TOE also uses SSL v3 to protect the communication between different servers that are part of the TOE. This communication always requires client and server authentication using digital certificates. The management of those certificates is part of the security function F.Management.

The TOE uses the GSKit component to implement the SSL Version 3 protocol and the certificate management functions required. The certificate management functions are addressed under F.Management.

The SSL protocol itself is defined in [SSLv3] and [RFC2246]. The TOE supports the following cipher suites defined in those standards:

- CipherSuite SSL_RSA_WITH_NUL_MD5
- CipherSuite SSL_RSA_WITH_NUL_SHA
- CipherSuite SSL_RSA_EXPORT_WITH_RC4_40_MD5
- CipherSuite SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5
- CipherSuite SSL_RSA_WITH_DES_CBC_SHA
- CipherSuite SSL_RSA_WITH_RC4_128_MD5
- CipherSuite SSL_RSA_WITH_RC4_128_SHA
- CipherSuite SSL_RSA_WITH_3DES_EDE_CBC_SHA

Only SSL_RSA_WITH_RC4_128_SHA, SSL_RSA_WITH_RC4_128_MD5 and SSL_RSA_WITH_3DES_EDE_CBC_SHA for SSL v3 are considered part of the evaluation. Those cipher suites are configured as the only cipher suites employed in the evaluated configuration, i.e. the TOE will choose them if they are supported by the client and not establish a connection if the client does not use one of those cipher suites.

6.2 TSF that are subject to a Strength of Function Analysis

All TSF that are based on probabilistic or permutational algorithms are subject to a strength of function analysis except those that use cryptographic algorithms. The function subject to an SOF rating is:

- F.Authentication (SOF-medium)

Within F.Authentication only the password-based mechanism for the authentication of Base Administrators is within the scope of the strength of function analysis.

The Common Criteria exclude the rating of cryptographic mechanisms including key generation. This Security Target does not make any claim on the strength of such mechanisms.

6.3 Statement of Assurance Measures

The following table provides an overview how the assurance measures of EAL3 and ALC_FLR.1 are satisfied:

Assurance Component	Assurance measures
ACM_CAP.3	IBM uses various configuration management tools, such as CMVC and Lotus Notes data bases, for source code, design documentation, guidance, testing, etc.
ACM_SCP.1	As mentioned above, source code, design documentation, user and administrator documentation as well as test documentation are maintained within CM systems.
ADO_DEL.1	Delivery procedures are described as part of the developer documentation. This includes also the measures taken to ensure the integrity and authenticity of the TOE during the delivery process.
ADO_IGS.1	The guidance documentation provided to the customer includes a detailed description how to install and configure the individual components that define the TOE. Additional guidance for the installation and configuration of exactly the evaluated configuration is provided as part of the guidance documentation.
ADV_FSP.1	The TSFI are identified in a separate document which points to the documents describing the different interfaces.
ADV_HLD.2	High level design documents exist that describes the internal structure of the TOE into subsystems, how the security functions of the TOE are implemented and how the subsystems contribute to the security functions.
ADV_RCR.1	Correspondence between the TSF as defined in the TOE summary specification and the functional specification as well as correspondence between the functional specification and the high level design is provided in form of commented tables that show the correspondence.
AGD_ADM.1	Administrator guidance documents exist for the Policy Server as well as for TAMOS RM as the Resource Manager / Authorization Evaluator used in the TOE. They describe the administrative tasks, the commands to be used and the different management aspects.
AGD_USR.1	There is no user guidance required for the TOE, since the user can be anybody that tries to access protected resources and users will not need to know anything about the security functions of the TOE. Organizations using the TOE within their environment have to educate their users to satisfy the requirements for protecting their passwords or private keys. Assumptions on user behavior in the TOE environment are postulated in the administrator guidance.
ALC_DVS.1	The security measures for the IBM development environment are derived from the IBM Global documents that define the minimum requirements for the physical and organizational security.
ALC_FLR.1	Problems that are reported either from the development process or by a customer will result in a “defect” that is managed with CMVC. Defects are classified with respect to their impact and one of the possible classifications is “security”. Since all defects are tracked and managed by CMVC, it is easily possible to extract all security relevant defects, their status and what has been done to fix them.

Assurance Component	Assurance measures
ATE_COV.1	Testing is performed as functional verification testing using a defined test suite in accordance with defined test procedures as described in the test plan. Coverage of security functions is provided in form of a table showing which test cases test which security functions at which interface. The table shows that all security functions and their parameter are tested at the interfaces defined in the functional specification.
ATE_DPT.1	A mapping is produced that shows the mapping of test cases to details defined in the high level design. The mapping shows that those details are covered by test cases and the test cases themselves show that the TOE operates in accordance with its high level design.
ATE_FUN.1	A test plan is provided that describes the test procedures, test cases, purpose of each test and expected results. Records of actual tests performed and their results are maintained under CM.
ATE_IND.2	Independent testing is performed as part of the evaluation by the evaluation facility. The test plan and test cases as well as the TOE suitable for testing will be provided to the evaluation facility such that all the test cases can be performed by the independent evaluator.
AVA_MSU.1	An analysis of the user provided documentation describing the installation and configuration, the administrator interface and commands and the configuration files is performed by the evaluation team to ensure that those documents are consistent and provide all the required guidance for an administrator to install, configure and administer the TOE in a secure manner.
AVA_SOF.1	A strength of function analysis is provided for the mechanisms based on permutational or probabilistic properties (except for cryptographic mechanism) to demonstrate that those mechanisms have a strength of SOF-medium or better.
AVA_VLA.1	A process is in place and documented to search for vulnerabilities of the TOE using open sources of vulnerabilities on the Internet like CVE or CERT advisories. The results of this process are documented and provide the developer vulnerability analysis as required.

7 PP claims

This Security Target does not claim compliance with any existing Protection Profile.

8 Rationale

This chapter provides the rationale for the selection of security objectives and requirements within APP.

8.1 Security Objectives Rationale

8.1.1 Security Objectives Coverage

The mapping in Table 2 indicates how each security objective for the TOE is traced back to at least one threat or organizational security policy.

Objective	**threat / OSP**
O.AUTHORIZATION	T.BYPASS, T.UAUSER
O.AUTHENT_ADMIN	T.UAUSER
O.ACC_ADM	P.ADM_DELEGATION,
O.AUDITING	T.UAUSER, T.UAACTION P.ACCOUNTABILITY
O.SEC_COM	T.UAUSER, T.COM_ATT
O.SOF	T.UAUSER

Table 2: security objectives traced back to threats and organizational security policies

The mappings in Table 3 and Table 4 indicate how each security objective for the environment is traced back to at least one assumption, threat or organizational security policy.

Objective (IT Environment)	**threat / OSP / assumption**
OE.OS_TIME	T.UAACTION (supportive) P.ACCOUNTABILITY
OE.OS_CFG_PROT	TE.BYPASS P.ACCOUNTABILITY
OE.DS_ACCESS_CTRL	A.DIR_PROT
OE.FRIENDLY_DS	T.UAUSER (supportive) A.DIR_PROT A.FRIENDLY_LDAP
OE.REPLICAS	A.DIR_PROT
OE.OS_AUTH	T.UAUSER (supportive) A.USER_PASSWORD
OE.SEPARATION	TE.BYPASS A.FRIENDLY_OS

Table 3: security objectives for the IT environment traced back to threats, organizational security policies and assumptions

Objective (non-IT Environment)	**threat / OSP / assumption**
OE.INSTALL OE.CREDEN	A.ADMIN A.PWD_SAFE
OE.PHYSICAL	A.ADMIN, A.PHYS_PROT
OE.OS_OPERATE	A.ADMIN, A.BOOT, A.OS_CONF_MGMT
OE.SEC_INTEGRATE	A.ADMIN, A.DIR_PROT, A.FRIENDLY_OS
OE.USER	A.ADMIN, A.USER

Table 4: security objectives for the non-IT environment traced back to threats, organizational security policies and assumptions

8.1.2 Security Objectives Sufficiency

The following arguments provide justification that the security objectives are suitable to counter each single threat and that each security objective tracing back to a threat, when achieved, actually contributes to the removal, diminishing or mitigation of that threat:

The threat **T.BYPASS**, imposing that an attacker uses non-TSF portions of the TOE to bypass the TSF, is removed by *O.AUTHORIZATION* requiring an implementation of the TOE that enforces access control before any transaction is allowed. Note that bypassing the TOE completely is addressed by the threat TE.BYPASS discussed below.

T.UAACTION imposes the threat of an attacker to perform unauthorized, TSP-violating actions without detection of those actions. This threat is removed by *O.AUDITING* requiring the TSF to log security relevant actions, supported by *OE.TIME* providing a reliable time source.

The threat of an attacker impersonating an authorized user, **T.UAUSER**, is efficiently diminished by *O.AUTHENT_ADMIN* requiring authentication for the TOE's administrators, supported by *OE.OS_AUTH in the IT environment* (see below) and further mitigated by *O.AUDITING* implementing audit records for security relevant actions. In addition *O.SEC_COM* prohibits that authentication credentials can be intercepted while transferred via the network connections. Note that this is further supported by OE.FRIENDLY_DS, since authentication decisions are derived from the external LDAP server. O.SOF additionally diminishes this threat by requiring strong passwords by means of a mandatory password policy.

The threat of an attacker intercepting and/or modifying the communication traffic between the TOE and an external entity or between physically distributed parts of the TOE, **T.COM_ATT**, is efficiently diminished by *O.SEC_COM* requiring that all communication between physically separated parts of the TOE is protected to maintain its confidentiality and integrity.

Attackers using functions in the IT environment to circumvent TSF, as outlined in **TE.BYPASS**, are prevented from succeeding by *OE.SEPARATION* demanding support of the underlying operating system in providing a security domain for the resource manager systems and OE.OS_CFG_PROT requiring protection of configuration files for the policy server.

The following arguments provide justification that the security objectives are suitable to

cover each single organizational security policy, that each security objective that traces back to an OSP, when achieved, actually contributes to the implementation of the OSP, and that if all security objectives that trace back to an OSP are achieved, the OSP is implemented.

P.ACCOUNTABILITY requires accounting for actions of administrators and access decision requests made by any user. This is covered by *O.AUDITING* containing the requirement of audit records for those very actions. The creation of audit records is supported by the environment as required in *OE.OS_TIME* by providing an accurate time source to be included in those records. In addition *OE.OS_CFG_PROT* supports this policy and requires that the audit configuration file is protected against unauthorized access.

The TOE shall allow the delegation of administrative tasks to manage only access control policy rules related to a dedicated subset of objects (targets) as in **P.ADM_DELEGATION**. This is implemented by *O.ACC_ADM* providing means to control the (management) access to certain access control policy rules by, again, access control policy rules (in this case, the targets of an access control decision request initiated by an administrator are access control policy rules related to a certain object space in the TOE environment).

The following arguments provide justification that the security objectives for the environment are suitable to cover each single assumption, that each security objective for the environment that traces back to an assumption about the environment of use of the TOE, when achieved, actually contributes to the environment achieving consistency with the assumption, and that if all security objectives for the environment that trace back to an assumption are achieved, the intended usage is supported.

The assumption on physical protection for the underlying machine of the TOE, **A.PHYS_PROT**, is covered by *OE.PHYSICAL* requiring protection of those machine(s) from unauthorized physical access.

The assumption **A.PWD_SAFE** on the protection of authentication credentials by administrators and other users of the TOE is achieved by *OE.CREDEN* requiring that appropriate measures for the protection of access credentials are ensured by the responsible personnel.

A.ADMIN assumes that administrators of the TOE and the underlying systems are trained, trustworthy and follow the guidance. This is covered by *OE.INSTALL* requiring competent and trustworthy administrators that deliver, install, manage and operate the TOE in a manner which maintains the IT security objectives and by *OE.OS_OPERATE* which makes dedicated requirements on the operation and configuration of the underlying machines hosting the TOE application. This is in addition achieved by the objectives OE.PHYSICAL, OE.SEC_INTEGRATE and OE.USER, expecting administrators to implement physical protection, secure integration of the TOE and the provision of guidance to users.

A.OS_CONF_MGMT assumes the reliable configuration and maintenance of the underlying operating system, emphasizing on the prevention of unauthorized (local or remote) access to operating system functions including network daemons. This is achieved by *OE.OS_OPERATE* which includes the demand for an appropriate installation, configuration and maintenance of the underlying operating system.

The assumption **A.DIR_PROT** assumes that protection features for the directory server used by the TOE exist, which prohibit unauthorized access to directory entries. This is achieved by the objective *OE.DS_ACCESS_CTRL* requiring the directory server to control the access to directory entries. Furthermore, OE.REPLICAS requires that a clear policy is implemented on how the LDAP server provides for the consistent replication of TSF data. In addition OE.FRIENDLY_DS requires that the LDAP server's support functions for the TOE behave as specified, and OE.SEC_INTEGRATE stipulates protection of the communication

between TOE and directory server.

The assumption **A.USER** is addressed by the objectives *OE.USER* which requires that the persons responsible for the TOE control the user community that can request access to resources protected by the TOE.

The assumption **A.FRIENDLY_LDAP** assumes the LDAP server in the IT environment to behave as specified. This is of relevance for OE.DS_ACCESS_CTRL and OE.FRIENDLY_DS, since the authentication, authorization and replication mechanisms of the LDAP server support the operation of the TSF.

The assumption **A.FRIENDLY_OS** assumes the operating system in the IT environment to be well behaved and cooperative with regard to supporting the TSF. This is achieved by the objective OE.SEPARATION, demanding support of the underlying operating system in providing a security domain for the resource manager systems. As a supportive means consumers of the TOE are encouraged in OE.SEC_INTEGRATE to gain additional assurance that the underlying operating system works as specified.

8.2 Security Requirements Rationale

This chapter provides the rationale for the selection of security requirements within APP. In addition to this rationale, chapter 5 includes application notes for several security functional requirements to further improve the interpretation of those requirements with respect to an APP-conformant implementation of the TOE.

8.2.1 Security Requirements Coverage

The following tables illustrate which security objectives are implemented by which security functional requirements. Table 5 indicates how each TOE security functional requirement can be traced back to at least one security objective for the TOE, Table 6 indicates how each functional security requirement for the IT environment can be traced back to at least one security objective for the environment.

SFR	**Objective**
FAU_GEN.1	O.AUDITING
FAU_GEN.2	O.AUDITING
FAU_GEN.3-TAMOS	O.AUDITING
FAU_SAR.1(1)	O.AUDITING
FAU_SAR.1(2)	O.AUDITING
FAU_SEL.1(1)	O.AUDITING
FAU_SEL.1(2)	O.AUDITING
FAU_STG.1	O.AUDITING
FCS_CKM.1(1)	O.SEC_COM
FCS_CKM.1(2)	O.SEC_COM
FCS_CKM.2(1)	O.SEC_COM

SFR	Objective
FCS_CKM.2(2)	O.SEC_COM
FCS_COP.1(1)	O.SEC_COM
FCS_COP.1(2)	O.SEC_COM
FDP_ACC.2(1)	O.AUTHORIZATION
FDP_ACC.2(2)	O.ACC_ADM O.AUTHORIZATION
FDP_ACF.1 (1)	O.AUTHORIZATION
FDP_ACF.1 (2)	O.ACC_ADM O.AUTHORIZATION
FIA_AFL.1(1)	O.AUTHENT_ADMIN O.SOF
FIA_AFL.1(2)	O.SOF
FIA_ATD.1(1)	O.AUTHORIZATION
FIA_ATD.1(2)	O.ACC_ADM O.AUTHENT_ADMIN O.AUTHORIZATION
FIA_SOS.1	O.SOF
FIA_UAU.2	O.AUTHENT_ADMIN
FIA_UID.1	O.AUDITING O.AUTHORIZATION
FIA_UID.2	O.AUDITING O.AUTHENT_ADMIN O.ACC_ADM O.AUTHORIZATION
FIA_USB.1	O.AUDITING O.AUTHENT_ADMIN O.ACC_ADM O.AUTHORIZATION
FMT_MOF.1	O.ACC_ADM
FMT_MSA.1(1)	O.ACC_ADM
FMT_MSA.1(2)	O.ACC_ADM
FMT_MSA.2	O.ACC_ADM O.SEC_COM
FMT_MSA.3	O.ACC_ADM
FMT_MTD.1	O.ACC_ADM

SFR	Objective
FMT_SMF.1	O.AUDITING O.AUTHENT_ADMIN O.ACC_ADM O.AUTHORIZATION O.SOF
FMT_SMR.1	O.ACC_ADM
FPT_ITT.1	O.SEC_COM
FPT_RVM.1	O.ACC_ADM O.AUTHORIZATION
FPT_TRC.1	O.SEC_COM
FTP_ITC.1	O.SEC_COM

Table 5: SFRs for the TOE traced back to objectives for the TOE

SFR (environment)	Objective (environment)
Operating System for Policy Server	
FIA_UID.1	OE.OS_CFG_PROT
FMT_MTD.1	OE.OS_CFG_PROT
FMT_SMF.1	OE.OS_CFG_PROT
FMT_SMR.1	OE.OS_CFG_PROT
FPT_SEP.1	OE.OS_CFG_PROT
FPT_STM.1	OE.OS_TIME
LDAP Server	
FDP_ACC.1	OE.DS_ACCESS_CTRL
FDP_ACF.1	OE.DS_ACCESS_CTRL
FDP_ETC.1	OE.DS_ACCESS_CTRL
FDP_ITC.1	OE.DS_ACCESS_CTRL
FIA_UAU.2	OE.DS_ACCESS_CTRL, OE.FRIENDLY_DS
FIA_UID.2	OE.DS_ACCESS_CTRL
FMT_MSA.1	OE.DS_ACCESS_CTRL, OE_FRIENDLY_DS
FMT_MSA.3	OE.DS_ACCESS_CTRL
FMT_SMF.1	OE.DS_ACCESS_CTRL
FMT_SMR.1	OE.DS_ACCESS_CTRL
Operating System for TAMOS RM	
FDP_ACC.1	OE.OS_CFG_PROT
FDP_ACF.1	OE.OS_CFG_PROT
FIA_ATD.1	OE.OS_CFG_PROT
FIA_UAU.1	OE.OS_AUTH
FIA_UID.1	OE.OS_AUTH
FIA_USB.1	OE.OS_AUTH
FMT_MSA.1	OE.OS_CFG_PROT
FMT_MSA.3	OE.OS_CFG_PROT
FMT_MTD.1	OE.OS_CFG_PROT
FMT_SMF.1	OE.OS_CFG_PROT
FMT_SMR.1	OE.OS_CFG_PROT
FPT_SEP.1	OE.SEPARATION
FPT_STM.1	OE.OS_TIME

Table 6: SFRs for the environment traced back to objectives for the environment

8.2.2 Security Requirements Sufficiency

The following arguments provide justification for each security objective for the TOE that the TOE security requirements are suitable to meet and achieve that security objective.

O.AUTHORIZATION requires that only authorized administrators and users gain access to the TOE and the resources it protects. For administrators, the TOE implements access control as in FDP_ACC.2(2) and FDP_ACF.1(2), the access control policy for users is modeled in FDP_ACC.2(1) and FDP_ACF.1(1). Access control contributes to the non-bypassability of the TSF (FPT_RVM.1). The implementation of the access control policies is supported by the ability to identify users (FIA_UID.1, FIA_UID.2), user-subject binding (FIA_USB.1), appropriate attributes (FIA_ATD.1(1), FIA_ATD.1(2)), and management functionality for access control (FMT_SMF.1).

O.AUDITING requires the ability to audit security relevant actions of users and administrators. FAU_GEN.1 and FAU_GEN.3-TAMOS specify the types of audit events that can be recorded; FAU_GEN.2 ensures that those records are associated with the originating user identity (as far as possible). FAU_SEL.1(1) and FAU_SEL.1(2) allow administrators to select levels of audits. FAU_SAR.1(1) and FAU_SAR.1(2) implement the requirements for the later analysis of audit records by authorized administrators. FAU_STG.1 protects the audit records against modification. Identification of users (FIA_UID.1, FIA_UID.2) and user-subject binding (FIA_USB.1) allows to record the originator of events as part of the audit records. FMT_SMF.1 enables the management of the audit functionality.

O.AUTHENT_ADMIN requires that the TOE enforces the authentication of administrators (with use of an external LDAP server). This is implemented by FIA_UID.2 requiring identification and FIA_UAU.2 requiring authentication before any action other than authentication can be performed on behalf of an administrator, and FIA_USB.1 providing for proper user-subject binding. Passwords are stored in the initiator security attribute data base for each administrator (FIA_ATD.1(2)). FIA_AFL.1(1) limits the attempts of unsuccessful authentication attempts to prevent password guessing. FMT_SMF.1 enables the management of the authentication function.

O.ACC_ADM requires that administrators must be able to specify which objects may be accessed by which administrators or users (i.e. to manage according access control policy rules). This is implemented by requiring appropriate access control policy rules in FDP_ACF.1(2) and FDP_ACC.2(2), which in turn allow administrators to access information – including access control policy rules – that is maintained by the TOE. In addition, FMT_MTD.1 limits the ability to modify or delete user attributes to authorized administrators. Roles are defined in FMT_SMR.1. FMT_SMF.1 allows management of access control policy rules, which is restricted to be accessible only by administrators by FMT_MOF.1. FMT_MSA.1(1), FMT_MSA.1(2) and FMT_MSA.3 refine the management of those rules. FMT_MSA.2 provides for secure values. Implementation of the administrator access control policy is supported by appropriate attributes as required by FIA_ATD.1(2), authentication of administrators as required by FIA_UID.2 and user-subject binding as required by FIA_USB.1. Access control contributes to the non-bypassability of the TSF (FPT_RVM.1).

O.SEC_COM requires the protection of communication links between separated parts of the TOE. FTP_ITC.1 addresses this by demanding such a protected link between the TOE and external entities (i.e. the LDAP server in the IT environment), while FPT_ITT.1 requires protected communication between different parts of the TOE when transferring TSF data. Internal TSF data consistency (FPT_TRC.1) is maintained via those secure channels. Protection of communications is implemented by use of cryptographic functions, which is expressed by FCS_CKM.1(1) (addressing the generation of session keys), FCS_CKM.1(2) (addressing the generation of RSA key pairs), FCS_CKM.2(1) for exchange of RSA public keys,

FCS_CKM.2(2) for the exchange of session keys, FCS_COP.1(1) for the digital signature generation and verification and FCS_COP.1(2) for the encryption of the data using a symmetric algorithm. In addition FMT_MSA.2 ensures the selection of secure values for keys.

O.SOF required the enforcement of password policies for administrators and users. This is achieved by a password policy defined in FIA_SOS.1 and supported by authentication failure handling in FIA_AFL.1(1) and FIA_AFL.1(2). Management of these mechanisms is provided in FMT_SMF.1.

The following section provides a mapping of security objectives for the TOE environment to security functional requirements for the IT environment.

OE.OS_TIME requires the provision of a reliable time source by the operating system in the IT environment. This is reflected in FPT_STM.1 for the operating systems for both Policy Server and TAMOS RM. requiring them to provide such a time source.

OE.OS_CFG_PROT requires the protection of configuration files within the operating system. For the Policy Server operating system, this is reflected in FMT_MTD.1 requiring to restrict the ability to modify one of those files (the configuration of audit events) to be accessible to authorized administrators only. Protection also requires the definition of roles in the operating system to define those allowed to access and modify configuration files, which is expressed in FMT_SMR.1 (the assignment of roles requires user identification, which is expressed in FIA_UID.1). In addition FMT_SMF.1 is included to address the requirement for user and access management. FPT_SEP.1 for the operating system supports the access protection. Vice versa, the requirements for the TAMOS RM underlying operating system include discretionary access control (since – as opposed to the Policy Server – here it is not assumed that the resource manager is the only user of the system), implemented in FDP_ACC.1 and FDP_ACF.1, and supported by FIA_ATD.1, FMT_MSA.1 and FMT_MSA.3. Analog to the Policy Server operating system, FMT_MTD.1, FMT_SMF.1 and FMT_SMR.1 are required.

OE.DS_ACCESS_CTRL requires authentication and access control mechanisms on the LDAP server in order to prevent unauthorized access to directory entries and support authentication of TOE administrators. This is reflected in the security functional requirements FDP_ACC.1 and FDP_ACF.1 for the directory server requiring an access control system as well as FDP_ETC.1 and FDP_ITC.1 addressing the import and export of user data (i. e. directory information). Identification and authentication are modeled in FIA_UAU.2 and FIA_UID.2. Management of users and the access control function is expressed in FMT_MSA.1, FMT_MSA.3 and FMT_SMF.1. To define a useful access control framework the directory server should have a concept to define user roles, which is expressed with FMT_SMR.1.

OE.FRIENDLY_DS requires a properly functioning LDAP server in order to provide authentication decisions to the TOE for administrators, as expressed in FIA_UID.2 and FIA_UAU.2. Note that the consistency aspects for replicas are not addressed by SFRs for the LDAP server – this Security Target does not impose certain functional mechanisms to provide for consistency between replicated information in the IT environment, but rather expects that whatever mechanism the LDAP server implements is sound and well known to the administrators of the TOE. It is the responsibility of the administrators to ensure this (cf. OE.REPLICAS).

OE.OS_AUTH requires identification and authentication of users of the TAMOS RM underlying operating system as a basis to implement the TOE's access control mechanism for users. This is reflected in the security functional requirements FIA_UAU.1 and FIA_UID.1

and in FIA_USB.1 requiring proper user-subject binding.

OE.SEPARATION requires the provision of a separation mechanism to protect the TAMOS RM application. This is modeled with the requirement FPT_SEP.1 to provide appropriate domain separation.

8.2.3 Security Requirements Dependencies

The following table shows the fulfillment of dependencies imposed on security functional requirements by Part 2 of the Common Criteria (the left column identifies the CC Part 2 component, the middle column identifies the dependencies on that component drawn from CC Part 2, and the right column illustrates how the dependency is fulfilled). Deviations between requirements and fulfillment are explained subsequent to those tables. No additional dependencies exist for the security functional requirements.

Dependencies within the EAL3 “package” selected for the security assurance requirements have been considered by the authors of CC Part 3 and are not analyzed again in this Security Target. The included component on flaw remediation, ALC_FLR.1, has no dependencies on other requirements.

The security functional requirements in this Security Target do not introduce dependencies on any security assurance requirement; neither do the security assurance requirements in this Security Target introduce dependencies on any security functional requirement.

SFR	**Dependencies**	**Fulfillment of dependencies**
FAU_GEN.1	FPT_STM.1	FPT_STM.1 (environment)
FAU_GEN.3-TAMOS	FPT_STM.1	FPT_STM.1 (environment)
FAU_GEN.2	FAU_GEN.1 FIA_UID.1	FAU_GEN.1, FAU_GEN.3-TAMOS FIA_UID.1 FIA_UID.2
FAU_SAR.1	FAU_GEN.1	FAU_GEN.1, FAU_GEN.3-TAMOS
FAU_SEL.1	FAU_GEN.1 FMT_MTD.1	FAU_GEN.1 FMT_MTD.1 (environment)
FAU_STG.1	FAU_GEN.1	FAU_GEN.1
FCS_CKM.1(1)	[FCS_CKM.2 or FCS_COP.1] FCS_CKM.4 FMT_MSA.2	FCS_CKM.2 FCS_COP.1(2) FMT_MSA.2 (see note 1 below)
FCS_CKM.1(2)	[FCS_CKM.2 or FCS_COP.1] FCS_CKM.4 FMT_MSA.2	FCS_CKM.2 FCS_COP.1(1) FMT_MSA.2 (see note 1 below)

SFR	Dependencies	Fulfillment of dependencies
FCS_CKM.2(1)	[FDP_ITC.1 or FCS_CKM.1] FCS_CKM.4 FMT_MSA.2	FDP_ITC.1 (environment) FCS_CKM.1(2) FMT_MSA.2 (see note 1 below)
FCS_CKM.2(2)	[FDP_ITC.1 or FCS_CKM.1] FCS_CKM.4 FMT_MSA.2	FCS_CKM.1(1) FMT_MSA.2 (see note 1 below)
FCS_COP.1(1)	[FDP_ITC.1 or FCS_CKM.1] FCS_CKM.4 FMT_MSA.2	FDP_ITC.1 FCS_CKM.1(2) FMT_MSA.2 (see note 1 below)
FCS_COP.1(2)	[FDP_ITC.1 or FCS_CKM.1] FCS_CKM.4 FMT_MSA.2	FCS_CKM.1(1) FMT_MSA.2 (see note 1 below)
FDP_ACC.2(1)	FDP_ACF.1	FDP_ACF.1(1)
FDP_ACC.2 (2)	FDP_ACF.1	FDP_ACF.1 (2)
FDP_ACF.1 (1)	FDP_ACC.1 FMT_MSA.3	FDP_ACC.2 (1) FMT_MSA.3
FDP_ACF.1 (2)	FDP_ACC.1 FMT_MSA.3	FDP_ACC.2 (2) FMT_MSA.3
FIA_AFL.1	FIA_UAU.1	FIA_UAU.1
FIA_ATD.1	No dependencies	No dependencies
FIA_SOS.1	No dependencies	No dependencies
FIA_UAU.2	FIA_UID.1	FIA_UID.2
FIA_UAU.5	No dependencies	No dependencies
FIA_UAU.6	No dependencies	No dependencies
FIA_UID.1	No dependencies	No dependencies
FIA_UID.2	No dependencies	No dependencies
FIA_USB.1	FIA_ATD.1	FIA_ATD.1
FMT_MOF.1	FMT_SMF.1 FMT_SMR.1	FMT_SMF.1 FMT_SMR.1
FMT_MSA.1(1)	[FDP_ACC.1 or FDP_IFC.1] FMT_SMF.1 FMT_SMR.1	FDP_ACC.2 (1) FMT_SMF.1 FMT_SMR.1

SFR	Dependencies	Fulfillment of dependencies
FMT_MSA.1(2)	[FDP_ACC.1 or FDP_IFC.1] FMT_SMF.1 FMT_SMR.1	FDP_ACC.2 (2) FMT_SMF.1 FMT_SMR.1
FMT_MSA.2	ADV_SPM.1 [FDP_ACC.1 or FDP_IFC.1] FMT_MSA.1 FMT_SMR.1	FDP_ACC.2(1)+(2) FMT_MSA.1(1)+(2) FMT_SMR.1 (see note 2 below)
FMT_MSA.3	FMT_MSA.1 FMT_SMR.1	FMT_MSA.1(1)+(2) FMT_SMR.1
FMT_MTD.1	FMT_SMF.1 FMT_SMR.1	FMT_SMF.1 FMT_SMR.1
FMT_SMF.1	No dependencies	No dependencies
FMT_SMR.1	FIA_UID.1	FIA_UID.1 FIA_UID.2
FPT_ITT.1	No dependencies	No dependencies
FPT_RVM.1	No dependencies	No dependencies
FPT_TRC.1	FPT_ITT.1	FPT_ITT.1
FTP_ITC.1	No dependencies	No dependencies

Table 7: Dependency Analysis for TOE SFRs

Note 1: The dependency FCS_CKM.4 on the secure destruction of cryptographic keys is applicable for the session keys used as well as the private RSA keys. Since those keys are within the TOE and with the physical security of the TOE and the requirement not to have any other application running on a machine of the TOE unless full separation between this application and the TOE can be provided, it does not seem to be suitable to require a secure destruction of session keys.

Note 2: FMT_MSA.2 has been included to address the dependencies from FCS_CKM.1, FCS_CKM.2, FCS_COP.1 on this requirement. This addresses the use of “secure values” for keys used for cryptographic operations. The symmetric keys used for secure communication between the TOE and external entities as well as for communication between different parts of the TOE are generated automatically as defined by the SSL standard. The requirement for an informal security policy model to satisfy the dependency is not resolved, since the generation and distribution of the symmetric keys would anyhow not been addressed by such a model. The other dependencies are formally resolved, but since the management of those session keys does not involve any user or administrator, none of the security functional requirements listed actually addresses the key generation process. It is therefore argued that the dependencies listed in FMT_MSA.2 are not applicable in this TOE.

SFR	Dependencies	Fulfillment of dependencies
LDAP Server		
FDP_ACC.1	FDP_ACF.1	satisfied
FDP_ACF.1	FDP_ACC.1 FMT_MSA.3	satisfied
FDP_ETC.1	[FDP_ACC.1 or FDP_IFC.1]	satisfied
FDP_ITC.1	[FDP_ACC.1 or FDP_IFC.1] FMT_MSA.3	satisfied
FIA_UAU.2	FIA_UID.1	Satisfied by FIA_UID2
FIA_UID.2	No dependencies	No dependencies
FMT_MSA.1	[FDP_ACC.1 or FDP_IFC.1] FMT_SMF.1 FMT_SMR.1	Satisfied
FMT_MSA.3	FMT_MSA.1 FMT_SMR.1	Satisfied
FMT_SMF.1	No dependencies	No dependencies
FMT_SMR.1	FIA_UID.1	Satisfied by FIA_UID.2
Policy Server System		
FIA_UID.1	No dependencies	No dependencies
FPT_STM.1	No dependencies	No dependencies
FMT_MTD.1	FMT_SMR.1 FMT_SMF.1	Satisfied
FMT_SMF.1	No dependencies	No dependencies
FMT_SMR.1	FIA_UID.1	Satisfied
FPT_SEP.1	No dependencies	No dependencies
TAMOS RM System		
FDP_ACC.1	FDP_ACF.1	Satisfied
FDP_ACF.1	FDP_ACC.1 FMT_MSA.3	Satisfied
FIA_ATD.1	No dependencies	No dependencies
FIA_UAU.1	FIA_UID.1	Satisfied
FIA_UID.1	No dependencies	No dependencies
FIA_USB.1	FIA_ATD.1	Satisfied

SFR	Dependencies	Fulfillment of dependencies
FMT_MSA.1	[FDP_ACC.1 or FDP_IFC.1] FMT_SMF.1 FMT_SMR.1	Satisfied
FMT_MSA.3	FMT_MSA.1 FMT_SMR.1	Satisfied
FMT_MTD.1	FMT_SMF.1 FMT_SMR.1	Satisfied
FMT_SMF.1	No dependencies	No dependencies
FMT_SMR.1	FIA_UID.1	Satisfied
FPT_SEP.1	No dependencies	No dependencies
FPT_STM.1	No dependencies	No dependencies

This table shows that all dependencies on security functional requirements are satisfied for all the identified components in the IT environment.

8.2.4 Internal Consistency and Mutual Support

Chapter 8.2.2 has already shown how the security functional requirements work together to implement the single objectives for the TOE and the IT environment. This chapter will elaborate on the internal consistency and mutual support of the security functional requirements. Further information can as well be found in the application notes to the security functional requirements in chapter 5.

Internal Consistency and Mutual Support of security requirements for the TOE

The main goal of the TOE is to perform access control decisions for resources stored in the TOE environment and to allow a sound management of the information that is necessary to do so. The most vital information needed for access control decisions are the access control policy rules, containing basically information on which target is allowed to be accessed by whom. An appropriate access control security function policy, the **Object-Space access control policy** is implemented by *FDP_ACF.1(1)*. The enforcement of the access control decisions made by the TOE is expressed by *FDP_ACC.2(1)*. Another input to this access control decision is the identity of the access request initiators. The TOE derives the identity of resource manager users (FIA_UID.1) from the underlying operating system. Authentication is implemented by maintaining a data base of initiator security attributes, which is reflected by *FIA_ATD.1(2) for users*, and requiring the appropriate user-subject binding in *FIA_USB.1*. For the password based authentication of users and administrators a minimum strength of the passwords is required and defined in *FIA_SOS.1*. In addition the number of consecutive unsuccessful authentication attempts is restricted as expressed in *FIA_AFL.1 (2) for users*. Since the policy database may be replicated, *FPT_TRC.1* ensures the consistency between the master policy database and the replica.

To allow for management of the access control policy rules used by the TOE for its access control decisions, administrators of the TOE need to be able to execute appropriate management functionality. To clearly separate those duties (and the originators of appropriate actions), a separation between user and administrator roles is introduced by *FMT_SMR.1*. To protect the integrity of the TSF, including the TSF data, access control is required for the

administrative access to the TOE. The same mechanisms are employed to implement this access control as they are used to fulfill the access control decisions for the access requests: access control policy rules define which administrator is allowed to access which administrative functionality of the TOE. To prevent confusion, this is expressed in a separate access control policy, the **management access control policy**, defined by *FDP_ACF.1 (2)*. Since it is not the intention of the TOE to leave the enforcement of access control decisions with respect to the management of the TSF up to its environment, the authentication enforcement for administrators (*FIA_UAU.2* and *FIA_UID.2*) as well as the enforcement of the access control decisions (*FDP_ACC.2 (2)*) must be implemented by the TOE. For this reason, the initiator security attribute data base contains as well the authentication secrets for administrative users of the TOE (*FIA_ATD.1(1)*). With respect to the authentication of administrators, authentication failure handling (*FIA_AFL.1(1)*) and a definition of minimum constraints to the choice of authentication secrets (*FIA_SOS.1*) are required.

To detect possible attacks and to allow accounting for administrative actions, requirements for the **generation of audit data** are included. Those comprise the definition of auditable events and the outline of audit records (*FAU_GEN.1, FAU_GEN.3-TAMOS*) as well as the association of auditable events with the identity of their originator, if the originator's identity is known (*FAU_GEN.2*). In order to comprehend the audited events, administrators must be able to inspect the audit trails (*FAU_SAR.1(1) and FAU_SAR.1(2)*) and the audit information has to be protected against unauthorized modifications (*FAU_STG.*1). To include the correct time of an event in the audit records, appropriate information has to be delivered by the IT environment (*FPT_STM.1* for the underlying operating systems). The requirements *FAU_SEL.1(1)* and *FAU_SEL.1(2)* and the requirements *FMT_MTD.1* allow administrators to specify which events are to be audited by the TOE.

Management functions are established by *FMT_SMF.1* to allow the management of authentication, communications security, audit and access control functionality. Such management is restricted by *FMT_MOF.1* to authorized administrators, while *FMT_MSA.1(1)* and *FMT_MSA.1(2)* and *FMT_MSA.3* impose the **management access control policy** on the management actions.

Secure communication between different distributed parts of the TOE is required by *FPT_ITT.1* and also between the TOE and other trusted IT products (LDAP server, resource managers) as required by *FTP_ITC.1*. In both cases this requires the use of cryptographic functions as defined by the SSL protocol. While the generation and import of the RSA public key pair is addressed in *FCS_CKM.1(2)*, the generation of the symmetric session keys as expressed by *FCS_CKM.1(1)*, the distribution of RSA public key certificates as expressed by *FCS_CKM.2(2)*, the distribution of the symmetric session keys as expressed by *FCS_CKM.2(1)* and the associated cryptographic operations for signing and signature verification (*FCS_COP.1(1)*) and symmetric encryption (*FCS_COP.1(2)*) are all addressed as requirements for the TOE as well. The requirement for secure session keys is expressed with *FMT_MSA.2.*

Bypass prevention for the security functions is achieved by introducing a requirement on TSP enforcement in *FPT_RVM.1*.

Internal Consistency and Mutual Support of security requirements for the IT environment

The TOE makes use of the **underlying operating systems** for both Policy Server and resource manager to provide a reliable time stamp (as required by *FPT_STM.1*) and the ability to manage the audit events of the TOE by editing the configuration file (as required by *FMT_MTD.1* and called out by *FMT_SMF.1*). Authorized roles need to be defined that are

allowed to access the audit records as required by *FMT_SMR.1* which depend on the correct identification of users as required by *FIA_UID.1.* This is supported by *FPT_SEP.1* requiring that the security functions of the underlying operating system maintain a domain for their own execution protected from inference or tampering by untrusted subjects.

Additional requirements on the **underlying operating system for the resource manager** exist: In order to protect the TOE resources in a multi-user environment, discretionary access control is required as expressed in FDP_ACC.1, FDP_ACF.1, FIA_ATD.1, FMT_MSA.1 and FMT_MSA.3. Also, the operating system has to authenticate users (FIA_UAU.1, FIA_UID.1, FIA_USB.1) in order to support the access control mechanisms implemented by the TOE.

The TOE uses an **LDAP Server** to store and maintain the user registry. To rely on the security of this LDAP Server it has to be ensured that this server provides functions for user authentication (as required by *FIA_UAU.2* and *FIA_UID.2*) and access control (as required by *FDP_ACC.1* and *FDP_ACF.1*). In addition export and import of user data has to be performed in accordance with the access control policy (as required by *FDP_ETC.1* and *FDP_ITC.1*). Management aspects are addressed by the security functional requirements *FMT_MSA.1, FMT_MSA.3, FMT_SMF.1* and *FMT_SMR.1*

8.2.5 Evaluation Assurance Level and Strength of Function

The evaluation assurance level (EAL) 3 was chosen as a medium level of assurance reflecting the expected assurance requirements of commercial customers using the target of evaluation (TOE) for the protection of data with a low or medium level of sensitivity. The TOE is intended to provide a reasonable level of protection for this data comparable to the protection provided by most commercial-off-the-shelf products. This is reflected as well in the definition of the TOE environment in chapter 2 and the security objectives for the TOE in chapter 4 of this Security Target.

The assurance level EAL3 was augmented with ALC_FLR.1 to address the flaw remediation process employed by IBM. Since the evaluation methodology for ALC_FLR has been harmonized and is also covered by the Mutual Recognition Arrangement, this was considered to be a useful augmentation for the assurance level chosen.

In line with this medium level of assurance the functions provided by the TOE that are subject to probabilistic or permutational analysis (except for cryptographic algorithms and algorithms related to the cryptographic functions) are claimed to have at least a medium strength (SOF-medium). The function that is subject to strength of function analysis is F.Authentication, which uses passwords. The certificate based authentication function and the cryptographic algorithms used within F.Communication as well as the related key generation process are not subject to a strength of function analysis.

8.3 TOE Summary Specification Rationale

8.3.1 Security Functions Justification

The following table shows that the TOE security functions specified in the TOE summary specification meet all security functional requirements for the TOE and work together to satisfy the TOE security functional requirements.

SFR	Security Functions from the TOE Summary Specification

SFR	Security Functions from the TOE Summary Specification
FAU_GEN.1	This requirement is addressed by the security function **F.Audit**, which defines the events that the different parts of the TOE are able to record, and the information contained in audit records.
FAU_GEN.2	This requirement is addressed by the functions **F.Authentication**, which requires users and administrators to be authenticated and **F.Audit** which audits the identity of the user that caused the event together with other relevant information.
FAU_GEN.3-TAMOS	This requirement is addressed by the security function **F.Audit**, which defines the events that the different parts of the TOE are able to record, and the information contained in audit records.
FAU_SAR.1(1) and (2)	This requirement is addressed by the function **F.Audit** which provides tools to review audit records or, for the Policy Server, audit logs that are human readable.
FAU_SEL.1(1) and (2)	This requirement is addressed by **F.Audit** which allows defining the events that are to be audited in a configuration file. Note that in theory the access control function of the TOE could be used to control access to the audit configuration file, but in most cases this would be left to the access control functions of the underlying operating system.
FAU_STG.1	This requirement is addressed by **F.Audit** setting appropriate access control permissions for audit logs that can be enforced by the DAC of the underlying operating system.
FCS_CKM.1(1)	This requirement is addressed by **F.Communication** which defines that the SSL protocol is used to secure the communication between different parts of the TOE as well as between the TOE and external entities. The cryptographic operations themselves are described in the SSL standard.
FCS_CKM.1(2)	This requirement is addressed by **F.Communication** which defines that the administrator can generate RSA key pairs for the use with the SSL v3 protocol. The TOE will only generate key pairs for its own internal use. External entities authenticating to the TOE using a digital certificate are assumed to have this authentication credential generated and protected securely.
FCS_CKM.2(1)	This requirement is addressed by **F.Communication** which defines that the SSL protocol is used to secure the communication between different parts of the TOE as well as between the TOE and external entities. The cryptographic operations themselves are described in the SSL standard.

SFR	Security Functions from the TOE Summary Specification
FCS_CKM.2(2)	This requirement is addressed by **F.Communication** which defines that the SSL protocol is used to secure the communication between different parts of the TOE as well as between the TOE and external entities. The cryptographic operations themselves are described in the SSL standard.
FCS_COP.1(1)	This requirement is addressed by **F.Communication** which defines that the SSL protocol is used to secure the communication between different parts of the TOE as well as between the TOE and external entities. The cryptographic operations themselves are described in the SSL standard.
FCS_COP.1(2)	This requirement is addressed by **F.Communication** which defines that the SSL protocol is used to secure the communication between different parts of the TOE as well as between the TOE and external entities. The cryptographic operations themselves are described in the SSL standard.
FDP_ACC.2(1)	This requirement is addressed by **F.Authorization** where the ACL and POP policies for resource objects are described.
FDP_ACC.2(2)	This requirement is addressed by **F.Authorization** where the access control policy for management objects is described.
FDP_ACF.1(1)	This requirement is addressed by **F.Authorization** where the ACL and POP policies for resource objects are described.
FDP_ACF.1(2)	This requirement is addressed by **F.Authorization** where the access control policy for management objects is described.
FIA_AFL.1(1) and (2)	This requirement is described in **F.Authentication** which describes the limits on the number of successive authentication failures allowed.
FIA_ATD.1(1) and (2)	This requirement is described in **F.Authentication** where the different user attributes are defined.
FIA_SOS.1	This requirement is described in **F.Authentication** where the different possibilities for the password policy are defined, while **F.Management** allows defining the policy.
FIA_UAU.2	This requirement is described in **F.Authentication** where the authentication process for administrators is described.
FIA_UID.1	This requirement relates to **F.Audit** for the identification of event generators and to **F.Authorization** where access control is enforced based on user identities.
FIA_UID.2	This requirement is described in **F.Authentication** where the identification of administrators is described.
FIA_USB.1	This requirement is described in **F.Authentication** where the binding of users / administrators to subjects is described. User-subject binding contributes to **F.Audit** and **F.Authorization**.

SFR	Security Functions from the TOE Summary Specification
FMT_MOF.1	This requirement relates to **F.Management**, which defines the management functionality for security functions.
FMT_MSA.1(1)	This requirement is described in **F.Authorization** where the different management objects and their management functions are described and in **F.Management** where ACL and POP management is described.
FMT_MSA.1(2)	This requirement is described in **F.Authorization** where the different management objects and their management functions are described and in **F.Management** where ACL and POP management is described.
FMT_MSA.2	This requirement relates to **F.Communication**, providing for secure cryptographic keys.
FMT_MSA.3	This requirement relates to **F.Management** where the inheritance policy for ACLs is described.
FMT_MTD.1	This requirement relates to **F.Management** where the management functions of administrators are described.
FMT_SMF.1	This requirement calls out the management functions described in **F.Management**.
FMT_SMR.1	This requirement is described in **F.Authorization** where the different roles are described.
FPT_ITT.1	This requirement is addressed by **F.Communication** which defines that the SSL protocol is used to secure the communication between different parts of the TOE. The cryptographic operations themselves are described in the SSL standard.
FPT_RVM.1	This requirement is addressed by the overall architecture of the TOE. Every request from a client system is passed through the authorization function **F.Authorization**.
FPT_TRC.1	This requirement is described in **F.Authorization** where the replication mechanism for the policy database is described, while **F.Communication** provides integrity of data exchanged via communications links.
FTP_ITC.1	This requirement is addressed by **F.Communication** which defines that the SSL protocol is used to secure the communication between the TOE and external entities. The cryptographic operations themselves are described in the SSL standard.

Table 8: Mapping Security Functional Requirements to Security Functions

8.3.2 Justification that the Security Functions are mutually supportive

The main objective of the TOE is to provide access control for objects hosted on managed resources.

As a result the TOE requires an authentication mechanism for administrators of the TOE, which is defined in **F.Authentication**. Authentication of users is acquired from the underly-

ing operating systems. Access control is divided into access control for user objects and access control for management objects as defined in **F.Authorization**. The rules for administrators how to define and manage access control lists and protected object policies – the two mechanisms the TOE uses for access control – as well as the rules for user and group management are defined in **F.Management**.

To allow for individual accountability the TOE also includes an audit function as described by **F.Audit**. This function can be used to trace the activities of users and administrators and make them accountable for the actions they have performed.

To be able to enforce this policy the TOE needs to establish secure communication links between itself and external entities (i.e. the LDAP server hosting the user registry) as well as between distributed parts of the TOE itself. This is accomplished by **F.Communication** defining that the SSL protocol is used for all those communication links. This protects the TSF data (including the protected objects) from unauthorized disclosure and modification when transmitted over communication links. It also ensures that the different parts of the TOE itself authenticate themselves to each other when establishing a communication link. Management of the certificates used for protected communication is defined in **F.Management**.

As a result one can state:

- Protected resources can only be accessed by users authorized for this access.
- Management of the TOE functions is restricted to defined administrators.
- Bypassing the TOE security functions or attacking the communication between the TOE and external entities or between distributed parts of the TOE is actively prohibited by the TOE.
- Accountability is enforced by an audit trail capable to audit all administrator actions as well as all access of users to protected resources.

This shows that the TOE security functions are mutually supportive.

8.4 PP Claims Rationale

No compliance with any existing Protection Profile is claimed.

9 Abbreviations

ACI	Access Control Information
ACL	Access Control List
ADF	Access Control Decision Function
ADI	Access Control Decision Information
AEF	Access Control Enforcement Function
API	Application Programming Interface
APP	Authorization Protection Profile
CC	Common Criteria, the name used historically for this multi-part standard ISO/IEC 15408 in lieu of its official ISO name of “Evaluation criteria for information technology security”
EAL	Evaluation Assurance Level
PP	Protection Profile
SF	Security Function
SFP	Security Function Policy
SOF	Strength of Function
ST	Security Target
TOE	Target of Evaluation
TSC	TSF Scope of Control
TSF	TOE Security Functions
TSFI	TSF Interface
TSP	TOE Security Policy

10 Glossary

Access Control Decision Function (ADF)	A specialized function that makes access control decisions by applying access control policy rules to an access request, ADI (of initiators, targets, access requests, or that retained from prior decision), and the context in which the access request is made.
Access Control Decision Information (ADI)	The portion (possibly all) of the ACI made available to the ADF in making a particular access control decision.
Access Control Enforcement Function (AEF)	A specialized function that is part of the access path between an initiator and a target on each access request and enforces the decision made by the ADF.
Access Control Information (ACI)	Any information used for access control purposes, including contextual information.
Access Control Policy	The set of rules that define the conditions under which an access may take place.
Access Control Policy Rules	Security policy rules concerning the provision of the access control service.
Access Request	The operations and operands that form part of an attempted access.
Assets	Information or resources to be protected by the countermeasures of a TOE.
Assignment	The specification of an identified parameter in a component.
Assurance	Grounds for confidence that an entity meets its security objectives.
Attack potential	The perceived potential for success of an attack, should an attack be launched, expressed in terms of an attacker's expertise, resources and motivation.
Augmentation	The addition of one or more assurance component(s) from Part 3 to an EAL or assurance package.
Authentication data	Information used to verify the claimed identity of a user.
Authorized user	A user who may, in accordance with the TSP, perform an operation.
Class	A grouping of families that share a common focus.
Component	The smallest selectable set of elements that may be included in a PP, an ST, or a package.
Connectivity	The property of the TOE which allows interaction with IT entities external to the TOE. This includes exchange of data by wire or by wireless means, over any distance in any environment or configuration.

Contextual Information	Information about or derived from the context in which an access request is made (e.g. time of day).
Dependency	A relationship between requirements such that the requirement that is depended upon must normally be satisfied for the other requirements to be able to meet their objectives.
Element	An indivisible security requirement.
Evaluation	Assessment of a PP, an ST or a TOE, against defined criteria.
Evaluation Assurance Level (EAL)	A package consisting of assurance components from Part 3 that represents a point on the CC predefined assurance scale.
Evaluation authority	A body that implements the CC for a specific community by means of an evaluation scheme and thereby sets the standards and monitors the quality of evaluations conducted by bodies within that community.
Evaluation scheme	The administrative and regulatory framework under which the CC is applied by an evaluation authority within a specific community.
Extension	The addition to an ST or PP of functional requirements not contained in Part 2 and/ or assurance requirements not contained in Part 3 of the CC.
External IT entity	Any IT product or system, untrusted or trusted, outside of the TOE that interacts with the TOE.
Family	A grouping of components that share security objectives but may differ in emphasis or rigor.
Formal	Expressed in a restricted syntax language with defined semantics based on well established mathematical concepts.
Human user	Any person who interacts with the TOE.
Identity	A representation (e.g. a string) uniquely identifying an authorized user, which can either be the full or abbreviated name of that user or a pseudonym.
Informal	Expressed in natural language.
Initiator	An entity (e.g. human user or computer-based entity) that attempts to access other entities.
Internal communication channel	A communication channel between separated parts of TOE.
Internal TOE transfer	Communicating data between separated parts of the TOE.
Inter-TSF transfers	Communicating data between the TOE and the security functions of other trusted IT products.

IT environment	See TOE environment.
Iteration	The use of a component more than once with varying operations.
Object	An entity within the TSC that contains or receives information and upon which subjects perform operations.
Organizational security policies	One or more security rules, procedures, practices, or guidelines imposed by an organization upon its operations.
Package	A reusable set of either functional or assurance components (e.g. an EAL), combined together to satisfy a set of identified security objectives.
Product	A package of IT software, firmware and/or hardware, providing functionality designed for use or incorporation within a multiplicity of systems.
Protection Profile (PP)	An implementation-independent set of security requirements for a category of TOEs that meet specific consumer needs.
Reference monitor	The concept of an abstract machine that enforces TOE access control policies.
Reference validation mechanism	An implementation of the reference monitor concept that possesses the following properties: it is tamperproof, always invoked, and simple enough to be subjected to thorough analysis and testing.
Refinement	The addition of details to a component.
Role	A predefined set of rules establishing the allowed interactions between a user and the TOE.
Secret	Information that must be known only to authorised users and/or the TSF in order to enforce a specific SFP.
Security attribute	Information associated with subjects, users and/or objects that is used for the enforcement of the TSP.
Security Function (SF)	A part or parts of the TOE that have to be relied upon for enforcing a closely related subset of the rules from the TSP.
Security Function Policy (SFP)	The security policy enforced by an SF.
Security objective	A statement of intent to counter identified threats and/or to satisfy identified organization security policies and assumptions.
tSecurity Target (ST)	A set of security requirements and specifications to be used as the basis for evaluation of an identified TOE.
Selection	The specification of one or more items from a list in a component.

Semiformal	Expressed in a restricted syntax language with defined semantics.
SOF-basic	A level of the TOE strength of function where analysis shows that the function provides adequate protection against casual breach of TOE security by attackers possessing a low attack potential.
SOF-high	A level of the TOE strength of function where analysis shows that the function provides adequate protection against deliberately planned or organized breach of TOE security by attackers possessing a high attack potential.
SOF-medium	A level of the TOE strength of function where analysis shows that the function provides adequate protection against straightforward or intentional breach of TOE security by attackers possessing a moderate attack potential.
Strength of Function (SOF)	A qualification of a TOE security function expressing the minimum efforts assumed necessary to defeat its expected security behavior by directly attacking its underlying security mechanisms.
Subject	An entity within the TSC that causes operations to be performed.
System	A specific IT installation, with a particular purpose and operational environment.
Target	An entity to which access may be attempted.
Target of Evaluation (TOE)	An IT product or system and its associated administrator and user guidance documentation that is the subject of an evaluation.
TOE environment	The term TOE environment depicts everything outside the actual physical and logical TOE boundary. The TOE may have certain expectations on the TOE environment. Requirements to provide supportive functions for the (technical) IT environment are expressed as security objectives for the environment and security functional requirements in the ST, while objectives on security policy support on the (organizational) non-IT environment are expressed in security objectives for the environment.
TOE resource	Anything useable or consumable in the TOE.
TOE Security Functions (TSF)	A set consisting of all hardware, software, and firmware of the TOE that must be relied upon for the correct enforcement of the TSP.

TOE Security Functions Interface (TSFI)	A set of interfaces, whether interactive (man-machine interface) or programmatic (application programming interface), through which TOE resources are accessed, mediated by the TSF, or information is obtained from the TSF.
TOE Security Policy (TSP)	A set of rules that regulate how assets are managed, protected and distributed within a TOE.
TOE security policy model	A structured representation of the security policy to be enforced by the TOE.
Transfers outside TSF control	Communicating data to entities not under control of the TSF.
Trusted channel	A means by which a TSF and a remote trusted IT product can communicate with necessary confidence to support the TSP.
Trusted path	A means by which a user and a TSF can communicate with necessary confidence to support the TSP.
TSF data	Data created by and for the TOE that might affect the operation of the TOE.
TSF Scope of Control (TSC)	The set of interactions that can occur with or within a TOE and are subject to the rules of the TSP.
User	Any entity (human user or external IT entity) outside the TOE that interacts with the TOE.
User data	Data created by and for the user, that does not affect the operation of the TSF.

11 References

[CAPP] Controlled Access Protection Profile, Version 1.d, National Security Agency

[CC] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; Version 2.1, August 1999

Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Requirements; Version 2.1, August 1999

Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements; Version 2.1, August 1999

[CEM] Common Methodology for Information Technology Security Evaluation CEM-99/045 Part 2: Evaluation Methodology, Version 1.0, August 1999

[SSLv3] Alain O. Freier, Philip Karlton, Paul C. Kocher: The SSL Protocol, Version 3; IETF Memo, Internet Draft, November 1996

[AZNAPI] Open Group Technical Standard: Authorization (AZN) API, The Open Group, January 2000

[ISO10181-3] ISO/IEC 10181-3: Information Technology – Open Systems Interconnection – Security frameworks for open systems: Access control framework, 1996

[AMBADM] Tivoli Access Manager for eBusiness Base Installation Guide, Version 5.1

[X.509] ITU-T RECOMMENDATION X.509 | ISO/IEC 9594-8: INFORMATION TECHNOLOGY - OPEN SYSTEMS INTERCONNECTION - THE DIRECTORY: PUBLIC-KEY AND ATTRIBUTE CERTIFICATE FRAMEWORKS