IBM WebSphere Portal 6.0
Security Target

Version 1.0
09/23/09

Prepared for:
IBM

Prepared By:
Science Applications International Corporation
Common Criteria Testing Laboratory
7125 Columbia Gateway Drive, Suite 300
Columbia, MD 21046

LIST OF TABLES

1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is WebSphere Portal 6.0 provided by IBM, which is designed to operate in the context IBM Websphere Application Server and provide a platform supporting and controlling access to web-related objects.

The Security Target contains the following additional sections:

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

1.1 Security Target, TOE and CC Identification

ST Title – IBM WebSphere Portal 6.0 Security Target

ST Version – Version 1.0

ST Date – 09/23/09

TOE Identification – IBM WebSphere Portal 6.0.0.0 (with APAR PK67104 and APAR PK79436)

TOE Developer – IBM

Evaluation Sponsor – IBM

CC Identification – Common Criteria for Information Technology Security Evaluation, Version 2.3, August 2005

1.2 Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Requirements, Version 2.3, August 2005.
  - Part 2 Conformant
- Common Criteria for Information Technology Security Evaluation Part 3: Security Assurance Requirements, Version 2.3, August 2005.
  - Part 3 Conformant
  - Assurance Level: EAL 4 augmented with ALC_FLR.2
  - Strength of Function Claim: SOF-medium

1.3 Conventions and Terminology

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

1.3.1 Conventions

The following conventions have been applied in this document:

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

1.3.2 Terminology

API	Application Programmable Interface
Authorised User	A user who may, in accordance with the TSP, perform an operation
CC	Common Criteria
CAI	Collaborative Application Infrastructure
EAL	Evaluation Assurance Level
Entity	Subject
ID	Identity
IT	Information Technology
ITSEC	IT Security Evaluation Criteria
J2EE	Java 2 Enterprise Edition
LDAP	Lightweight Directory Access Protocol
OS	Operating System
PAC	Portal Access Control
PAC Client	A component or portlet of WP that invokes the PAC for access control decisions
Portlet	A portlet is a small portal application, usually depicted as a small box in a web page.
Principal	An entity within the portal that can be authorized, i.e. user or group
Resource	An entity within the portal controlled by PAC (e.g. page, portlet) i.e. an object
SF	Security Function. A part or parts of the TOE that have been relied upon for enforcing a closely related subset of rules from the TSP.

SFR	Security Functional Requirement
SLES	SuSE Linux Enterprise Edition
SOF	Strength Of Function
ST	Security Target
TAI	Templating Application Infrastructure
TOE	(Target Of Evaluation) An IT product or system and its associated administrator and user guidance documentation that is the subject of the evaluation.
TSF	(TOE Security Functions) A set consisting of all hardware, software and firmware of the TOE that must be relied upon for the correct enforcement of the TSP.
TSP	(TOE Security Policy) A set of rules that regulate how assets are managed, protected and distributed within the TOE.
WAR	Web Application aRchive
WAS	WebSphere Application Server
Web Modules	Web modules are portlet WAR files that are installed on WAS.
WMM	WebSphere Member Manager
WP	WebSphere Portal
WSRP	Web Services for Remote Portals (OASIS Standard)

2. TOE Description

The Target of Evaluation (TOE) is WebSphere Portal 6.0.0.0 (also known as WebSphere Portal (WP)) with APAR PK67104 and APAR PK79436.

2.1 TOE Overview

WP is a Java 2 Enterprise Edition (J2EE) application executed in the run-time environment provided by WebSphere Application Server (WAS) version 6.0.2.29 that provides users a consistent view of portal applications and allows users to define specific sets of applications which are presented in a single context. WP allows authorized users to establish protected portal resources like pages and portlets. As an example, authorized users (a team) can develop, share, and store information for projects. This allows for fast access to and transfer of information between members of the team working on the same project.

The Access Control administration can be performed using corresponding portlets within the running portal, the *XmlAccess* interface, or via portal scripting.

WP contains the following components:

- Page Aggregation. This is used for generating the content returned to the client e.g. the objects to display on the browser;
- Deployment. This is used for installing new portlets on a running portal and for including remote portlets from remote portals via Web Services for Remote Portals (WSRP);
- WSRP. Implements the WSRP protocol to allow access to remote portlets from a WSRP Producer Portal or to offer local portlets to be included into other remote portals;
- Portal Access Control (PAC). This controls access to all protected portal resources;
- Application Infrastructure consisting of the Collaborative Application Infrastructure (CAI) and a corresponding templating infrastructure (TAI);

- Policy. A concept to manage sets of configuration settings for the portal;
- Administration UIs: These user interfaces all the administration of the portal. The available user interfaces consist of text based scripting interfaces as well as a set of administration portlets including:
    - Manage Users and Groups,
    - Resource Permissions,
    - User and Group Permissions,
    - Membership,
    - Roles,
    - Manage Applications,
    - Manage Portlets,
    - Manage Pages,
    - URL Mapping Portlet,
    - Global Settings Portlet,
    - Manage Web Modules,
    - Application Template Library, and
    - Web Service Configuration.

2.2 TOE Architecture

WP allows authorized users to establish protected portal resources as defined later in this document. As an example, authorized users (a team) can develop, store, and share information for projects. This then allows for fast access to and transfer of information between members of the team working on the same project.

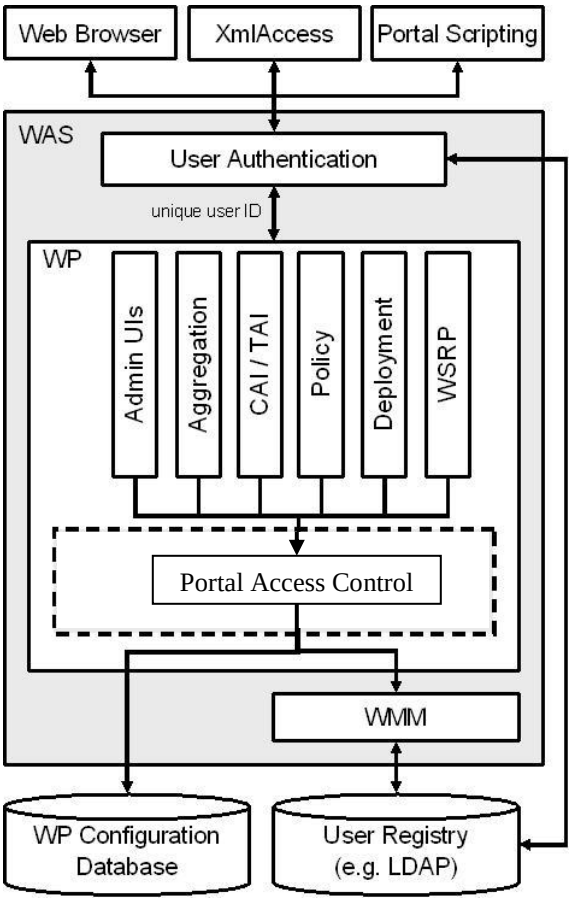

Figure 2.1: WP Dependencies and scope of evaluation.

WP provides access control to protected resources as identified later in this document. Access control checks are implemented specifically by the Portal Access Control (PAC) component within WP. This is shown within Figure 2.1, which illustrates that PAC is the common point of access control. While the PAC makes all the access decisions, the other WP components (as listed in the figure above) are responsible to enforce the decisions of the PAC so that the access control is effective (i.e., not bypassable). As such, all the components of WP cooperate to instantiate the TSF.

When a user requests access to a resource from the web browser, WP relies upon WAS to perform identification and management of users, WebSphere Member Manager (WMM) to provide the group membership and a database for the mapping of users to roles and the actions to resources. The request is passed to PAC to make an access decision to be enforced by the other applicable (depending on the specific service being invoked) WP component. Neither WAS or WMM are within the scope of evaluation and are therefore part of the TOE environment. WP also relies upon WAS and a database for its own proper and secure operation. More specifically, it is expected that the operating environment provided by the underlying WAS will serve to protect the execution environment of WP and the database will serve to protect the WP data so that it is accessible only by WP. In general, WP expects WAS and WMM to protect against attempts, outside the control of the TOE, at tampering with or bypass of the TOE security functions.

Figure 2.1 shows the PAC component as the central point of access control. The main interfaces to TOE, which are exposed through the various WP components, are:

- The Administration (Admin) Portlets[1];
- XmlAccess
- Portal Scripting

Administration portlets are the GUI Administration interface operated through a Web Browser. XmlAccess is a command line tool that allows importing and exporting portal configuration data from and to XML documents. Portal Scripting is an interactive command line tool for WP administration similar to an operating system command line shell. The term *Admin Portlets* includes the administration portlets contained in the WP product for administrative purposes as identified in section 2.1 above.

The following table identified the administration portlets used to create the respective protected resources:

Resource	**Admin Portlets**
Web Module	Manage Web Modules
Portlet Application Definition	Manage Applications
Portlet	Manage Portlets
Page	Manage Pages
User Group	Manage Users and Groups
URL Mapping Context	URL Mapping Portlet
Application	Manage Applications
Template	Application Template Library
Application Role	Roles
Policy	Resource Policies
WSRP Producer	Web Service Configuration

The page aggregation component is used for generating the content returned to the client e.g. the objects to display on the browser. The deployment component is used for installing new portlets on a running portal. The application infrastructure is used to create and manage portal applications and corresponding application templates. The policy component creates and manages individual policy objects which comprise reusable portal configuration settings. The WSRP component allows to consume remote portlets from a WSRP Producer Portal or to offer local portlets to be included into other remote portals via the Web Services for Remote Portals (WSRP) protocol. The resources managed by these components need to be protected from unauthorized access. In order to do so, each request that gets dispatched to one of these components is authorized through a method call to the PAC component. Only if the authorization check succeeds, the requested operation is actually executed. If the authorization check fails, a corresponding error message is presented.

Since these components use the PAC component to get access control decisions, they are also referred to as 'PAC clients'. The administration portlets mentioned earlier are PAC clients as well, since they do pro-active access control filtering. This means, in most cases the portlets call PAC to get an access control decision prior to exposing the UI components for a specific functionality in order to only expose UI facilities for those functions the executing user is actually allowed to execute. This function can be viewed as a usability feature since the final access control check is performed by the executing component (e.g. the deployment component).

2.2.1 Physical Boundaries

The TOE is the WP product. Its boundaries are primarily defined by the interfaces offered to the resources to which it controls access and the other components upon which it relies for supporting services.

[1] Portlets are the heart of a portal. A portlet is a small application window, usually depicted as a small box in a web page. Portlets are re-usable UI components that provide access to backend business logic, web-based content and other resources or services. Portlets can be grouped together in a portlet application. Portlets run inside the portlet container of WP, similar to a servlet running on an application server.

The following subsections offer more information about the TOE as well as a summary of the use of key supporting components not included in the TOE.

Note that since WP is an application operating within the context of WAS and its host operating system and also relying on data stored external to itself, WP depends on the security offered by those other products (many of which have been independently evaluated) in order that it is protected from potential attacks unrelated to the interfaces offered directly by WP.

2.2.1.1 Portal Access Control (PAC)

PAC is the single access control decision point within the TOE. It controls access to all sensitive portal resources. Protected resources are resources that can be accessed by a restricted set of users only. In order to be granted access to a protected resource in a specific way, the user needs a corresponding permission on this resource, e.g. a specific portal page can only be viewed by a specific user, if the user has the permission to perform the action 'View' on that page.

Note that this discussion focuses on the PAC only because it is the central access control decision maker utilized by all other components of the TOE.

Protected Resources

The following types of resources are protected within the portal:

- Web Modules: Web modules are portlet archives that are installed on WAS. Web modules can contain multiple portlet applications. If a new Web module is installed, it is automatically a child of the Web Modules virtual resource.
- Portlets (Portlet Definitions): A portlet is an installed portlet having its own portlet configuration. For example, a Mail portlet can be configured to a specific mail server.
- Portlet Application Definitions: Portlet applications provide a logical grouping of individual portlets. If a new Web module is installed, the portlet applications contained within that Web module are automatically child resources of the Portlet Applications virtual resource. Portlets contained within a portlet application appear as child nodes of that portlet application. A two-layer hierarchy consisting of portlet applications and the corresponding portlets exists beneath the Portlet Applications virtual resource.
- Content Nodes (Pages): Pages (also known as content nodes) contain the content that determines the portal navigation hierarchy. A portal page is basically the frame that contains a specific set of individual portlets arranged in a specific layout. If a new top-level page is created, it is automatically a child resource of the Content Nodes virtual resource. If a new page is created beneath an existing page, the new page is automatically child of the existing page.
- Application Template: An application template is the formal description of a portal application represented by an XML document. A template can be instantiated multiple times to create corresponding portal applications.
- Template Category: Individual application templates can be organized into individual categories similar to a folder structure for document organization.
- User Groups: Users can be grouped into user groups (database records). User groups can be nested. Access privileges are propagated with user groups membership. If a new user group is created, it will appear as a corresponding child resource underneath the virtual resource User Groups.
- URL Mapping contexts: URL mapping contexts are user-defined definitions of URL spaces that map to portal content. If a new top-level URL mapping context is created, it is automatically a child resource of the URL Mapping Contexts virtual resource. If a new URL mapping context is created beneath an existing context, the new context is automatically a child the existing context. URL mapping contexts inherit access control configuration from their parent context unless role blocks are used.

- Policy: A policy contains a set of related configuration settings that can be attached to individual users or other portal artifacts in a flexible fashion.
- WSRP Producer: A WSRP Producer identifies a remote portal server the provides individual portlets via the WSRP protocol

Users (database records) are implicitly protected resources, which means that access to specific user profile data can only be obtained via corresponding privileges on a user group that contains the given user as a member i.e. implicitly protected resources are those resources that are not linked into the protected resource hierarchy. Implicitly protected resources behave in the same way as normal protected resources. The Users virtual resource protects sensitive operations that deal with user management. For example, in order to add a user to a user group you must have the Security Administrator role.

PAC directly supports access control configuration of hierarchical resource topologies through the concept of permission inheritance. This concept reduces the administration overhead for an administrator when controlling access to a large number of portal resources. Inherited permissions are automatically assembled into roles that can be assigned to individual users and user groups, granting them access to whole sets of logically related portal resources. Permission inheritance can be prevented using role blocks. Role blocks can be either inheritance or propagation blocks, which either prevent child resources from inheriting permissions from their parent resources, or preventing parent resources from propagating permissions to their child resources respectively.

Each of these resources has a database entry which contains a reference to the parent resource, a list of role blocks tied to the resource, a reference to the owner of the resource and a list of the roles that exist on the resource. The access permissions contained in those roles are dependant upon the type of the roles the resources contained in the sub-tree rooted by this resource and the role blocks that exist on those resources.

In addition to protected resources, portal access control supports the notion of virtual resources that are used to group resources of a specific type and to configure access to abstract concepts within the portal e.g. the virtual node *portal* provides a means to give a user full control over the portal. Permission inheritance on the virtual resources behaves in the same way as non-virtual resources. The portal defines a set of fixed virtual resources, which are created and initialized during portal installation.

Figure 2.2 shows the general layout of the resource topology that is protected by Portal Access Control, Figure 2.3 depicts an example sub-set of this topology that could exist in a real portal setup. Implicitly protected resources (light yellow boxes in Figure 2.2) are protected via their non-implicit parent resources. Thus, they do not show up in the PAC administration user interfaces they are not represented within the protected resource hierarchy.

While it is possible to configure WP to allow the access control functionality to be performed externally, however WP has no control over external applications within the environment and therefore this functionality is outside the scope of the evaluation.

Virtual Resources

The portal defines a fixed set of virtual resources that are created and initialized during portal installation. Virtual resources are resources that are used to group resources of a specific type and to configure access to abstract concepts within the portal that do not directly relate to individual resource instances. Thus, virtual resources have two functions:

- They protect sensitive operations that affect the entire portal or specific concepts in the portal. For example, the XmlAccess virtual resource protects the ability to execute the XmlAccess import/export tool.
- They are well-known parent resources for resource instances. For example, the Web Modules virtual resource is the root node of all Web modules instances within the portal. Thus, role assignments on the Web Modules virtual resource permit access to all Web modules in the portal (as long as no role blocks have been added to those resources).

Figure 2.2 shows the general layout of the resource topology that is protected by PAC.

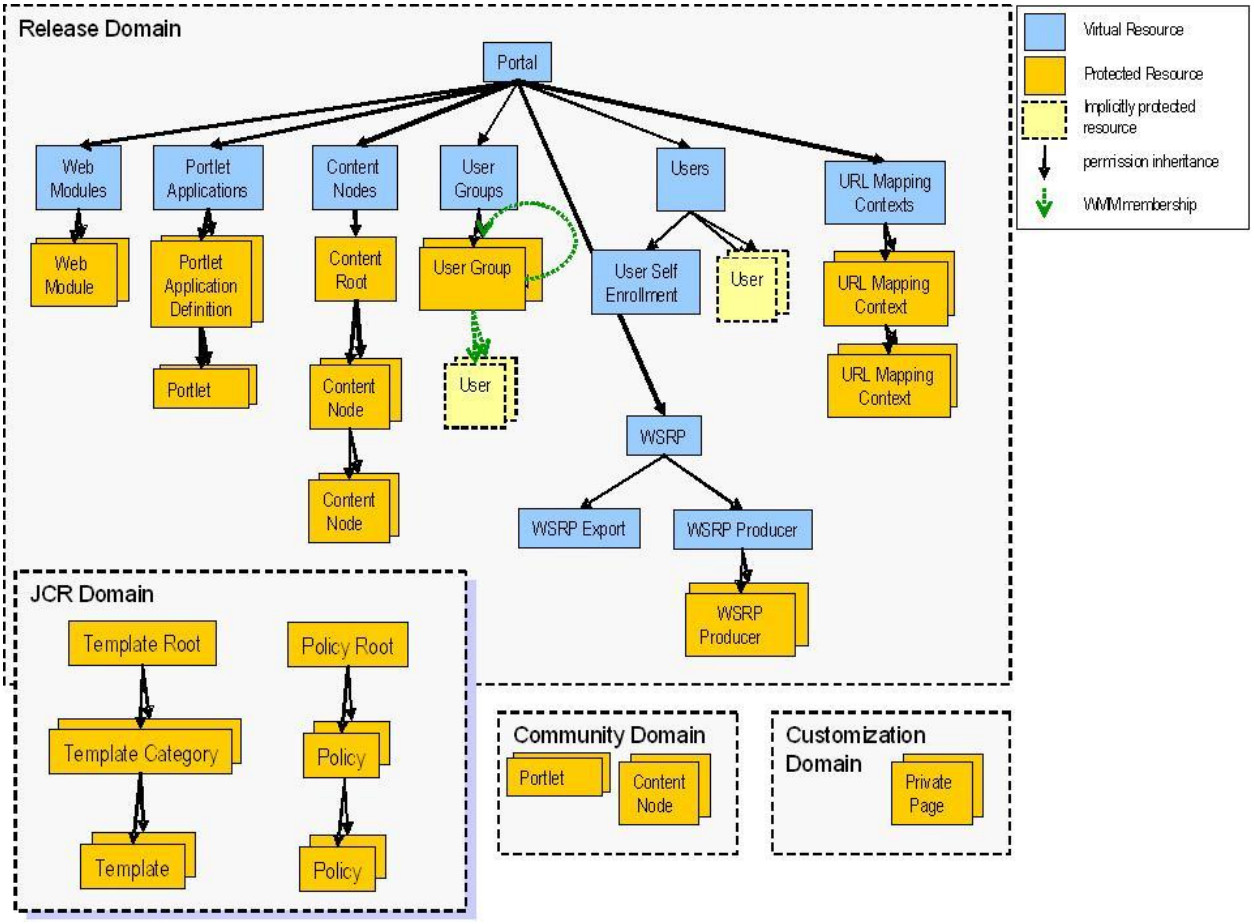

Figure 2.2: General Layout of Resource Topology

Figure 2.3 shows an example subset of the topology shown in Figure 2.2 that could exist in real portal installation.

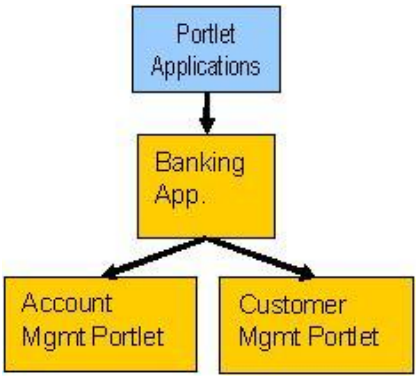

Figure 2.3: Example Sub-set of Resource Topology

The TOE also supports the virtual principals *Anonymous User* and *All-Authenticated-Users*. The *Anonymous User* can be used to grant permissions to users who have not been authenticated by the portal (i.e. WAS). The *All-Authenticated-Users* is the virtual group of users who have successfully authenticated (by WAS) to the portal[2]. These principals are called "virtual" since they do not map to entities within the users subsystem but are concepts within the access control component.

[2]Reference to Authentication is used purely as a description for these principals, and is not intended to imply that authentication functionality is included as part of this evaluation.

Data Domains

The term 'Data Domain' represents a data sink being used to persist portal configuration data. A specific data domain is identified by a reference to a database instance and a reference to a database schema within the database instance. Portal Access control needs to be able to protect resources that are stored in different data domains and to federate the results of access control decisions that involve resources of multiple data domains. This is required due to allowing consistent database back-up and restore with respect to access control configuration data. Portal version 6.0 supports four such data domains named: *release*, *community*, *customization* and *JCR* (see Figure 2.2).
While access control administration for the resources in the *release* and *JCR* domains is done using the *Resource Permissions* and *Users and Groups Permissions* portlets, the resources contained within the *community* domain are administered using the *Roles* and *Membership* portlets.

Ownership

Every time a protected resource is created within the portal, the user that created that resource becomes the initial owner of that resource. The owner of a resource is always allowed a specific set of actions on the owned resources. Ownership of resources can be changed during the lifetime of a resource.

Actions

Actions model the different ways of accessing a specific resource. PAC supports the following actions:

- Grant Access On;
- Delegate To;
- Add Child;
- Add Private Child;
- Delete;
- Edit;
- Personalize;
- View.

Role Types

A role type is a named set of actions that provides a grouping of individual actions (e.g. *Editor* = {Traverse, View, Edit, Add Child}). The portal provides a set of predefined role types each of which containing a set of actions that is typically needed to fulfill specific tasks within the portal (e.g. adjust and modify the layout of shared resources).

The default role types within the TOE are:

- Admin;
- Security Admin;
- Delegator;
- Manager;
- Editor
- Contributor
- Privileged User; and
- User.

Roles

A Role is a set of permissions identified by a role type and a resource. A permission is combination of an action (like view, edit, etc.) and a resource reference (e.g. AccountMgmtPortlet). Roles are created within the portal by combining a role type with a specific resource within the resource topology. The resulting set of permissions is determined by combining the set of actions contained in the role type on the resource and all child resources (as long as no role blocks are encountered). For example, combining the role type 'Administrator' to the resource named 'Banking App' in Figure 2.3 would result in a role that contains administrator specific permissions on the Banking App resource and the two portlets 'Account Mgmt. Portlet' and 'Customer Mgmt. Portlet'. The set of resources affected by a role are also referred to as the 'role domain' of the role instance with the identifying resource being called the 'domain root resource'.

The portal installation procedure creates a set of initial roles defining the out-of-the-box access control configuration. Furthermore, portal access control supports the notion of a 'Domain Administrator'. A domain administrator for a specific data domain is granted an implicit role assignment to the Administrator role tied to the root resource of the corresponding data domain.

Roles can be assigned to individual principals granting those principals the corresponding permissions. For example, let there be a role called *Editor@AccountMgmtPortlet* containing the permissions (View, *AccountMgmtPortlet*) and (Edit, *AccountMgmtPortlet*). If this role is assigned to the user group *SalesForce*, all members of this group (including nested groups) are allowed to perform the actions *View* and *Edit* on the *AccountMgmtPortlet*, i.e. they are allowed to see *AccountMgmtPortlet* and to modify its preferences.

Application Roles

To simplify organization of permissions, several roles can be aggregated to an application role. An Application role is a named set of roles that can be assigned to users or groups. Since the PAC role concept is defined to be a 'set of permissions', an application role is a 'set of set of permissions'. Assigning a user/group an individual application role, grants the union of all permissions contained in this set of sets of permissions to the assigned user or user group. Note that there is an Application Manager role and an Application Membership Manager role that are defined for every application. These are not considered to be security management roles since they are based on combinations of application access rights and serve only as pre-defined examples of potential roles that could be defined for a given application based on combinations of available access rights.

2.2.1.2 Data Storage

The TOE environment includes WMM for retrieving user group information and several database references for storing and retrieving access control configuration data.

WebSphere Member Manager (WMM)

WMM provides user profile information and group membership information to the TOE. For the scope of this CC evaluation WMM is within the TOE environment and responsible for the group membership of users. WMM relies upon a database and/or other user registries (e.g. directory servers supporting the LDAP protocol) for storage of the group memberships.

Note that user and group management itself is considered outside of the TOE and managed by the WMM component. The User and group management portlets made available by the TOE solely provide a user interface for functionality provided by the WMM component. Also, any constraints on users, groups, and their attributes are implemented outside of the TOE (e.g., within WMM).

WMM is expected to be hosted on a secure server and configured such that it, its functions, and its data are protected from tampering or unauthorized access that might serve detrimental to the TOE.

WebSphere Portal Database Instances

The TOE stores data in various data domains dependant on where the resources under protection are being stored. WP 6.0 uses 4 different data domains all pointing to relational database management systems. The information stored by the TOE in those data domains consists of references to the protected resources and their parent-child relationships (see Figure 2.2), the individual role instances and role blocks, ownership information and individual role assignments to users or user groups. In each case, the TOE creates its own database schemas in the databases containing the portal database tables. Those tables are used exclusively by the TOE.

2.2.1.3 WebSphere Application Server (WAS)

The TOE relies on WAS (in the TOE environment) for the identification and authentication of principals. WAS provides the user's unique ID, which is retrieved from an authentication component (via a WAS Application Programmable Interface (API)). Once the TOE has received a user ID from WAS, then it consults the WMM, which provides the user's group membership(s). Note that WAS establishes user sessions and maintains those session in order to keep user sessions separate as well as to protect WP when instantiated within WAS.

2.2.2 Logical Boundaries

This section summarizes the security functions provided by WebSphere Portal:

- User data protection
- Security management
- Protection of the TSF

2.2.2.1 User data protection

As previously described, the TOE offers a Portal Access Control (PAC) mechanism that is invoked by the other TOE components and makes access decisions for the following resources offered by WP: Web Modules, Portlets (Portlet Definitions), Portlet Application Definitions. Content Nodes (Pages), Application Templates, Template Categories, User Groups, URL Mapping contexts, Policies, and WSRP Producers. Access is controlled based on permissions contained in roles assigned to individual users or user groups. The WP components enforce the AC decision made by PAC.

2.2.2.2 Security management

As previously described, the TOE supports roles defined as sets of resource specific permissions. Roles can be assigned to users, groups, and can also be aggregated into other (application) roles.

Roles can be used to enable security management functions, such as managing roles and assigning those roles to users and user groups. In general, access to resources is restrictive insofar as a given user must have permission before they can access a resource. While it is possible to authorize anonymous access to a resource, such permission must be explicitly established prior to so doing.

The TOE also supports the ability to log the use of the security management functions. While generated by the TOE, the log is stored in ASCII format in a file in the IT environment.

2.2.2.3 Protection of the TSF

The TOE ensures that its own security policies cannot be bypassed by ensuring that appropriate access checks are made and enforced at all interfaces made available by the TOE.

2.3 TOE Documentation

IBM offers a series of documents that describe the installation of WebSphere Portal as well as guidance for subsequent use and administration of the applicable security features (see section 6.2 for details).

3. Security Environment

The TOE security environment describes the security aspects of the intended environment in which the TOE is to be used and the manner in which it is expected to be employed. The statement of the TOE security environment defines the following:

- Organizational Policies that the TOE and the environment of the TOE fulfill
- Threats that the TOE and the environment of the TOE counters
- Assumptions made about the operational environment and the intended method of use for the TOE

Furthermore, the TOE is intended to be used in environments where the relative assurance that its security functions are enforced is commensurate with EAL 4 augmented with ALC_FLR.2 as defined in the CC.

3.1 Organizational Policies

P.ACCESS	The right to access a resource is determined on the basis of: the user groups the user is a member of; the role instances assigned to those user groups or to the user; the role instances contained in application roles assigned to those user groups or to the user; the permissions contained in the individual assigned role instances which are determined on the basis of the actions contained in the corresponding role type and the set of descendant resources in the role domain which is determined in the basis of the topology defining the parent child relation ship among the resources and the role blocks that exist on those resources; and, the set of resources owned by the user or the user groups the user is a member of.

3.2 Threats

T.ACCESS_RES	A user of the TOE gains access to an object without the correct authority to access that object.
T.APP	The applications that the TOE depends upon might be compromised by a potential attacker.

3.3 Assumptions

A.ADMIN	It is assumed that there are one or more competent individuals that are assigned to manage the TOE and the security of the information it contains. Such personnel are assumed not to be careless, willfully negligent or hostile.
A.APP	It is assumed that the applications that the TOE relies upon, have been configured in accordance with the manufacturer's installation guides and where applicable, in its evaluated configuration. It is securely configured such that the applications protect the TOE from any unauthorized users or processes.

4. Security Objectives

This section defines the security objectives for the TOE and its supporting environment. The security objectives are intended to counter identified threats, comply with defined organizational security policies, and address applicable assumptions.

4.1 Security Objectives for the TOE

O.ACCESS The TOE must ensure that only those users with the correct authority are able to access a resource.

O.MANAGE The TOE must allow administrators to effectively manage the TOE and that this is only performed by authorized users.

4.2 Security Objectives for the IT Environment

O.CONFIG The IT environment must ensure that users of the TOE have associated user IDs and where applicable have an associated Group ID, that reliable time information is available to the TOE, and also that the TOE execution environment is appropriately protected from bypass or other penetration attempts.

4.3 Security Objectives for the Environment

O.ADMIN Those responsible for the TOE environment are competent and trustworthy individuals, capable of managing the TOE environment and the security of the information it contains.

O.APP Those responsible for the TOE environment must ensure that the supporting applications are installed and configured in accordance with the manufacturer's instructions, the evaluated configuration where applicable and is secure.

O.RECOVER Those responsible for the TOE environment must ensure that procedures and/or mechanisms are provided to ensure that after system failure or other discontinuity, recovery without a security compromise is obtained.

5. IT Security Requirements

The security requirements for the TOE have all been drawn from Parts 2 and 3 of the Common Criteria. The security functional requirements have been selected to correspond to the actual security functions implemented by the TOE while the assurance requirements have been selected to offer a moderate to high degree of assurance that those security functions are properly realized by users of the TOE.

5.1 TOE Security Functional Requirements

The following table describes the SFRs that are candidates to be satisfied by WebSphere Portal.

Requirement Class	Requirement Component
FDP: User data protection	FDP_ACC.2: Complete access control
	FDP_ACF.1: Security attribute based access control
FMT: Security management	FMT_LOG_EX.1: Logging of security management functions
	FMT_MSA.1: Management of security attributes
	FMT_MSA.3: Static attribute initialization
	FMT_SMF.1: Specification of Management Functions
	FMT_SMR.1: Security roles
FPT: Protection of the TSF	FPT_RVM.1a: Reference Mediation

Table 1 TOE Security Functional Components

5.1.1 User data protection (FDP)

5.1.1.1 Complete access control (FDP_ACC.2)

FDP_ACC.2.1 The TSF shall enforce the [**Access Control Policy**] on [**users; objects (web modules, portlet application definitions, portlets, content nodes (pages), user groups, users (implicitly protected via user groups), URL mapping contexts, templates, template categories, policies, and WSRP producers);**] 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.

5.1.1.2 Security attribute based access control (FDP_ACF.1)

FDP_ACF.1.1 The TSF shall enforce the [**Access Control Policy**] to objects based on the following: [

- **user attributes - user IDS, group IDs, and roles and**
- **object (as specified in FDP_ACC.2.1) attributes – resource identity, topology information, inheritance blocks, and ownership**].

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [**the user is permitted an attempted operation on an identified resource when the corresponding access permissions are a subset of the permissions granted to the user which are a union of the following:**

- **permissions resulting from the user ID or any of the user's group IDs being the owner of identified resources and**
- **permissions assigned to roles assigned to the user ID or any of the user's group IDs derived from an ancestor of the identified resource in the resource topology and that has not been blocked from inheritance].**

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [**no additional rules**].

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

5.1.2 Security management (FMT)

5.1.2.1 Logging of security management functions (FMT_LOG_EX.1)

FMT_LOG_EX.1.1 The TSF shall be capable of generating a record of the following events:

Assigning and revoking roles,

Modification of role blocks, and

Changing resource ownership.

FMT_LOG_EX.1.2 The TSF shall record the following information within each record:

User ID,

Time and date,

Event type, and

Resource identity (when applicable).

5.1.2.2 Management of security attributes (FMT_MSA.1)

FMT_MSA.1.1 The TSF shall enforce the [**Access Control Policy**] to restrict the ability to [***[see table below]***] the security attributes [**see table below**] to [**see table below**].

Role	Operation	Conditions
Administrator@Portal or Security Administrator@Portal	Any below	no conditions
Other than (Administrator@Portal or Security Administrator@Portal)	Assign/Revoke Role	A user can assign a role to another user if that user has 'Grant Access On' the applicable resource; that user has permissions equivalent to those being assigned to the applicable resource; and that user has 'Delegate To' permissions on the other user.
Other than (Administrator@Portal or Security Administrator@Portal)	Create/Delete Role Block	A user can block or unblock a role type from propagating/inheriting at an applicable resource if that user has 'Grant Access On' permissions on the applicable resource and that user has permissions equivalent to the role type to the applicable resource.
Other than (Administrator@Portal or Security Administrator@Portal)	Change Ownership	A user can change ownership of a resource from one user to another if that user has permissions equivalent to those of the resource owner; that user has 'Grant Access On' permission on the applicable resource; and that user has 'Delegate To' permissions on both users.

5.1.2.3 Static attribute initialization (FMT_MSA.3)

FMT_MSA.3.1 The TSF shall enforce the [**Access Control Policy**] to provide [***restrictive***] default values for security attributes that are used to enforce the SFP.

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

5.1.2.4 Specification of Management Functions (FMT_SMF.1)

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

- **assign and revoke roles to/from users and user groups;**

- **modify inheritance and propagation role blocks; and**
- **change ownership**].

5.1.2.5 Security roles (FMT_SMR.1)

FMT_SMR.1.1 The TSF shall maintain the role **types**[3] [

- **Administrator,**
- **Security Administrator,**
- **Delegator,**
- **Manager,**
- **Editor,**
- **Contributor,**
- **Privileged User, and**
- **User**]

and corresponding role instances that emerge from associating those role types to individual resources.

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

5.1.3 Protection of the TSF (FPT)

5.1.3.1 Reference Mediation (FPT_RVM.1a)

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

5.2 IT Environment Security Functional Requirements

The following table describes the SFRs that are candidates to be satisfied by the IT environment of WebSphere Portal.

Requirement Class	Requirement Component
FIA: Identification and authentication	FIA_ATD.1: User attribute definition
	FIA_UAU.1: Timing of Authentication
	FIA_UID.1: Timing of identification
FPT: Protection of the TSF	FPT_RVM.1b: Reference Mediation
	FPT_SEP.1: Domain Separation
	FPT_STM.1: Reliable time stamps

Table 2 IT Environment Security Functional Components

5.2.1 Identification and authentication (FIA)

5.2.1.1 User attribute definition (FIA_ATD.1)

FIA_ATD.1.1 The ~~TSF~~ **IT Environment** shall maintain the following list of security attributes belonging to individual users: [**user ID and group IDs**].

5.2.1.2 Timing of Authentication (FIA_UAU.1)

FIA_UAU.1.1 The ~~TSF~~ **IT Environment** shall allow [**assumption of the anonymous user ID**] on behalf of the user to be performed before the user is authenticated.

[3] Note that the roles are identified as types since they are each defined relative to a specific domain (e.g., @portal) defined within the TOE and as a result there can be multiple instances of each role since there can be multiple hierarchically organized domains.

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

5.2.1.3 Timing of identification (FIA_UID.1)

FIA_UID.1.1 The ~~TSF~~ **IT Environment** shall allow [**assumption of the anonymous user ID**] on behalf of the user to be performed before the user is identified.

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

5.2.2 Protection of the TSF (FPT)

5.2.2.1 Reference Mediation (FPT_RVM.1b)

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

5.2.2.2 Domain Separation (FPT_SEP.1)

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

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

5.2.2.3 Reliable time stamps (FPT_STM.1)

FPT_STM.1.1 The ~~TSF~~ **IT environment** shall be able to provide reliable time stamps for its own use **and that of the TOE**.

5.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are the EAL 4 augmented with ALC_FLR.2 components as specified in Part 3 of the Common Criteria. No operations are applied to the assurance components.

Requirement Class	Requirement Component
ACM: Configuration management	ACM_AUT.1: Partial CM automation
	ACM_CAP.4: Generation support and acceptance procedures
	ACM_SCP.2: Problem tracking CM coverage
ADO: Delivery and operation	ADO_DEL.2: Detection of modification
	ADO_IGS.1: Installation, generation, and start-up procedures
ADV: Development	ADV_FSP.2: Fully defined external interfaces
	ADV_HLD.2: Security enforcing high-level design
	ADV_IMP.1: Subset of the implementation of the TSF
	ADV_LLD.1: Descriptive low-level design
	ADV_RCR.1: Informal correspondence demonstration
	ADV_SPM.1: Informal TOE security policy model
AGD: Guidance documents	AGD_ADM.1: Administrator guidance
	AGD_USR.1: User guidance
ALC: Life cycle support	ALC_DVS.1: Identification of security measures
	ALC_FLR.2: Flaw reporting procedures
	ALC_LCD.1: Developer defined life-cycle model
	ALC_TAT.1: Well-defined development tools
ATE: Tests	ATE_COV.2: Analysis of coverage
	ATE_DPT.1: Testing: high-level design

	ATE_FUN.1: Functional testing
	ATE_IND.2: Independent testing - sample
AVA: Vulnerability assessment	AVA_MSU.2: Validation of analysis
	AVA_SOF.1: Strength of TOE security function evaluation
	AVA_VLA.2: Independent vulnerability analysis

Table 3 EAL 4 augmented with ALC_FLR.2 Assurance Components

5.3.1 Configuration management (ACM)

5.3.1.1 Partial CM automation (ACM_AUT.1)

ACM_AUT.1.1d The developer shall use a CM system.
ACM_AUT.1.2d The developer shall provide a CM plan.
ACM_AUT.1.1c The CM system shall provide an automated means by which only authorised changes are made to the TOE implementation representation.
ACM_AUT.1.2c The CM system shall provide an automated means to support the generation of the TOE.
ACM_AUT.1.3c The CM plan shall describe the automated tools used in the CM system.
ACM_AUT.1.4c The CM plan shall describe how the automated tools are used in the CM system.
ACM_AUT.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.1.2 Generation support and acceptance procedures (ACM_CAP.4)

ACM_CAP.4.1d The developer shall provide a reference for the TOE.
ACM_CAP.4.2d The developer shall use a CM system.
ACM_CAP.4.3d The developer shall provide CM documentation.
ACM_CAP.4.1c The reference for the TOE shall be unique to each version of the TOE.
ACM_CAP.4.2c The TOE shall be labelled with its reference.
ACM_CAP.4.3c The CM documentation shall include a configuration list, a CM plan, and an acceptance plan.
ACM_CAP.4.4c The configuration list shall uniquely identify all configuration items that comprise the TOE.
ACM_CAP.4.5c The configuration list shall describe the configuration items that comprise the TOE.
ACM_CAP.4.6c The CM documentation shall describe the method used to uniquely identify the configuration items.
ACM_CAP.4.7c The CM system shall uniquely identify all configuration items.
ACM_CAP.4.8c The CM plan shall describe how the CM system is used.
ACM_CAP.4.9c The evidence shall demonstrate that the CM system is operating in accordance with the CM plan.
ACM_CAP.4.10c The CM documentation shall provide evidence that all configuration items have been and are being effectively maintained under the CM system.
ACM_CAP.4.11c The CM system shall provide measures such that only authorised changes are made to the configuration items.
ACM_CAP.4.12c The CM system shall support the generation of the TOE.
ACM_CAP.4.13c The acceptance plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.
ACM_CAP.4.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.1.3 Problem tracking CM coverage (ACM_SCP.2)

ACM_SCP.2.1d The developer shall provide a list of configuration items for the TOE.
ACM_SCP.2.1c The list of configuration items shall include the following: implementation representation; security flaws; and the evaluation evidence required by the assurance components in the ST.
ACM_SCP.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.2 Delivery and operation (ADO)

5.3.2.1 Detection of modification (ADO_DEL.2)

ADO_DEL.2.1d The developer shall document procedures for delivery of the TOE or parts of it to the user.
ADO_DEL.2.2d The developer shall use the delivery procedures.
ADO_DEL.2.1c The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.
ADO_DEL.2.2c The delivery documentation shall describe how the various procedures and technical measures provide for the detection of modifications, or any discrepancy between the developer's master copy and the version received at the user site.
ADO_DEL.2.3c The delivery documentation shall describe how the various procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.
ADO_DEL.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

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

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

5.3.3 Development (ADV)

5.3.3.1 Fully defined external interfaces (ADV_FSP.2)

ADV_FSP.2.1d The developer shall provide a functional specification.
ADV_FSP.2.1c The functional specification shall describe the TSF and its external interfaces using an informal style.
ADV_FSP.2.2c The functional specification shall be internally consistent.
ADV_FSP.2.3c The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing complete details of all effects, exceptions and error messages.
ADV_FSP.2.4c The functional specification shall completely represent the TSF.
ADV_FSP.2.5c The functional specification shall include rationale that the TSF is completely represented.
ADV_FSP.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_FSP.2.2e The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

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

ADV_HLD.2.1d The developer shall provide the high-level design of the TSF.
ADV_HLD.2.1c The presentation of the high-level design shall be informal.
ADV_HLD.2.2c The high-level design shall be internally consistent.
ADV_HLD.2.3c The high-level design shall describe the structure of the TSF in terms of subsystems.
ADV_HLD.2.4c The high-level design shall describe the security functionality provided by each subsystem of the TSF.
ADV_HLD.2.5c The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.
ADV_HLD.2.6c The high-level design shall identify all interfaces to the subsystems of the TSF.

ADV_HLD.2.7c The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.
ADV_HLD.2.8c The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing details of effects, exceptions and error messages, as appropriate.
ADV_HLD.2.9c The high-level design shall describe the separation of the TOE into TSP-enforcing and other subsystems.
ADV_HLD.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_HLD.2.2e The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

5.3.3.3 Subset of the implementation of the TSF (ADV_IMP.1)

ADV_IMP.1.1d The developer shall provide the implementation representation for a selected subset of the TSF.
ADV_IMP.1.1c The implementation representation shall unambiguously define the TSF to a level of detail such that the TSF can be generated without further design decisions.
ADV_IMP.1.2c The implementation representation shall be internally consistent.
ADV_IMP.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_IMP.1.2e The evaluator shall determine that the least abstract TSF representation provided is an accurate and complete instantiation of the TOE security functional requirements.

5.3.3.4 Descriptive low-level design (ADV_LLD.1)

ADV_LLD.1.1d The developer shall provide the low-level design of the TSF.
ADV_LLD.1.1c The presentation of the low-level design shall be informal.
ADV_LLD.1.2c The low-level design shall be internally consistent.
ADV_LLD.1.3c The low-level design shall describe the TSF in terms of modules.
ADV_LLD.1.4c The low-level design shall describe the purpose of each module.
ADV_LLD.1.5c The low-level design shall define the interrelationships between the modules in terms of provided security functionality and dependencies on other modules.
ADV_LLD.1.6c The low-level design shall describe how each TSP-enforcing function is provided.
ADV_LLD.1.7c The low-level design shall identify all interfaces to the modules of the TSF.
ADV_LLD.1.8c The low-level design shall identify which of the interfaces to the modules of the TSF are externally visible.
ADV_LLD.1.9c The low-level design shall describe the purpose and method of use of all interfaces to the modules of the TSF, providing details of effects, exceptions and error messages, as appropriate.
ADV_LLD.1.10c The low-level design shall describe the separation of the TOE into TSP-enforcing and other modules.
ADV_LLD.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_LLD.1.2e The evaluator shall determine that the low-level design is an accurate and complete instantiation of the TOE security functional requirements.

5.3.3.5 Informal correspondence demonstration (ADV_RCR.1)

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

5.3.3.6 Informal TOE security policy model (ADV_SPM.1)

ADV_SPM.1.1d The developer shall provide a TSP model.

ADV_SPM.1.2d The developer shall demonstrate correspondence between the functional specification and the TSP model.
ADV_SPM.1.1c The TSP model shall be informal.
ADV_SPM.1.2c The TSP model shall describe the rules and characteristics of all policies of the TSP that can be modeled.
ADV_SPM.1.3c The TSP model shall include a rationale that demonstrates that it is consistent and complete with respect to all policies of the TSP that can be modeled.
ADV_SPM.1.4c The demonstration of correspondence between the TSP model and the functional specification shall show that all of the security functions in the functional specification are consistent and complete with respect to the TSP model.
ADV_SPM.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.4 Guidance documents (AGD)

5.3.4.1 Administrator guidance (AGD_ADM.1)

AGD_ADM.1.1d The developer shall provide administrator guidance addressed to system administrative personnel.
AGD_ADM.1.1c The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.
AGD_ADM.1.2c The administrator guidance shall describe how to administer the TOE in a secure manner.
AGD_ADM.1.3c The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.
AGD_ADM.1.4c The administrator guidance shall describe all assumptions regarding user behaviour that are relevant to secure operation of the TOE.
AGD_ADM.1.5c The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.
AGD_ADM.1.6c The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.
AGD_ADM.1.7c The administrator guidance shall be consistent with all other documentation supplied for evaluation.
AGD_ADM.1.8c The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.
AGD_ADM.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.4.2 User guidance (AGD_USR.1)

AGD_USR.1.1d The developer shall provide user guidance.
AGD_USR.1.1c The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.
AGD_USR.1.2c The user guidance shall describe the use of user-accessible security functions provided by the TOE.
AGD_USR.1.3c The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.
AGD_USR.1.4c The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behaviour found in the statement of TOE security environment.
AGD_USR.1.5c The user guidance shall be consistent with all other documentation supplied for evaluation.
AGD_USR.1.6c The user guidance shall describe all security requirements for the IT environment that are relevant to the user.
AGD_USR.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.5 Life cycle support (ALC)

5.3.5.1 Identification of security measures (ALC_DVS.1)

ALC_DVS.1.1d The developer shall produce development security documentation.
ALC_DVS.1.1c The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.
ALC_DVS.1.2c The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.
ALC_DVS.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ALC_DVS.1.2e The evaluator shall confirm that the security measures are being applied.

5.3.5.2 Flaw reporting procedures (ALC_FLR.2)

ALC_ FLR.2.1d The developer shall provide flaw remediation procedures addressed to TOE developers.
ALC_ FLR.2.2d The developer shall establish a procedure for accepting and acting upon all reports of security flaws and requests for corrections to those flaws.
ALC_ FLR.2.3d The developer shall provide flaw remediation guidance addressed to TOE users.
ALC_ FLR.2.1c The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.
ALC_ FLR.2.2c The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.
ALC_ FLR.2.3c The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.
ALC_ FLR.2.4c The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.
ALC_ FLR.2.5c The flaw remediation procedures shall describe a means by which the developer receives from TOE users reports and enquiries of suspected security flaws in the TOE.
ALC_ FLR.2.6c The procedures for processing reported security flaws shall ensure that any reported flaws are corrected and the correction issued to TOE users.
ALC_ FLR.2.7c The procedures for processing reported security flaws shall provide safeguards that any corrections to these security flaws do not introduce any new flaws.
ALC_ FLR.2.8c The flaw remediation guidance shall describe a means by which TOE users report to the developer any suspected security flaws in the TOE.
ALC_ FLR.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.5.3 Developer defined life-cycle model (ALC_LCD.1)

ALC_LCD.1.1d The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.
ALC_LCD.1.2d The developer shall provide life-cycle definition documentation.
ALC_LCD.1.1c The life-cycle definition documentation shall describe the model used to develop and maintain the TOE.
ALC_LCD.1.2c The life-cycle model shall provide for the necessary control over the development and maintenance of the TOE.
ALC_LCD.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.5.4 Well-defined development tools (ALC_TAT.1)

ALC_TAT.1.1d The developer shall identify the development tools being used for the TOE.
ALC_TAT.1.2d The developer shall document the selected implementation-dependent options of the development tools.
ALC_TAT.1.1c All development tools used for implementation shall be well-defined.

ALC_TAT.1.2c The documentation of the development tools shall unambiguously define the meaning of all statements used in the implementation.
ALC_TAT.1.3c The documentation of the development tools shall unambiguously define the meaning of all implementation-dependent options.
ALC_TAT.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.6 Tests (ATE)

5.3.6.1 Analysis of coverage (ATE_COV.2)

ATE_COV.2.1d The developer shall provide an analysis of the test coverage.
ATE_COV.2.1c The analysis of the test coverage shall demonstrate the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.
ATE_COV.2.2c The analysis of the test coverage shall demonstrate that the correspondence between the TSF as described in the functional specification and the tests identified in the test documentation is complete.
ATE_COV.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

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

ATE_DPT.1.1d The developer shall provide the analysis of the depth of testing.
ATE_DPT.1.1c The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design.
ATE_DPT.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.6.3 Functional testing (ATE_FUN.1)

ATE_FUN.1.1d The developer shall test the TSF and document the results.
ATE_FUN.1.2d The developer shall provide test documentation.
ATE_FUN.1.1c The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.
ATE_FUN.1.2c The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.
ATE_FUN.1.3c The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.
ATE_FUN.1.4c The expected test results shall show the anticipated outputs from a successful execution of the tests.
ATE_FUN.1.5c The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.
ATE_FUN.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.6.4 Independent testing - sample (ATE_IND.2)

ATE_IND.2.1d The developer shall provide the TOE for testing.
ATE_IND.2.1c The TOE shall be suitable for testing.
ATE_IND.2.2c The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.
ATE_IND.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ATE_IND.2.2e The evaluator shall test a subset of the TSF as appropriate to confirm that the TOE operates as specified.
ATE_IND.2.3e The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

5.3.7 Vulnerability assessment (AVA)

5.3.7.1 Validation of analysis (AVA_MSU.2)

AVA_MSU.2.1d The developer shall provide guidance documentation.
AVA_MSU.2.2d The developer shall document an analysis of the guidance documentation.
AVA_MSU.2.1c The guidance documentation shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.
AVA_MSU.2.2c The guidance documentation shall be complete, clear, consistent and reasonable.
AVA_MSU.2.3c The guidance documentation shall list all assumptions about the intended environment.
AVA_MSU.2.4c The guidance documentation shall list all requirements for external security measures (including external procedural, physical and personnel controls).
AVA_MSU.2.5c The analysis documentation shall demonstrate that the guidance documentation is complete.
AVA_MSU.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AVA_MSU.2.2e The evaluator shall repeat all configuration and installation procedures, and other procedures selectively, to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.
AVA_MSU.2.3e The evaluator shall determine that the use of the guidance documentation allows all insecure states to be detected.
AVA_MSU.2.4e The evaluator shall confirm that the analysis documentation shows that guidance is provided for secure operation in all modes of operation of the TOE.

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

AVA_SOF.1.1d The developer shall perform a strength of TOE security function analysis for each mechanism identified in the ST as having a strength of TOE security function claim.
AVA_SOF.1.1c For each mechanism with a strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the minimum strength level defined in the PP/ST.
AVA_SOF.1.2c For each mechanism with a specific strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the specific strength of function metric defined in the PP/ST.
AVA_SOF.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AVA_SOF.1.2e The evaluator shall confirm that the strength claims are correct.

5.3.7.3 Independent vulnerability analysis (AVA_VLA.2)

AVA_VLA.2.1d The developer shall perform a vulnerability analysis.
AVA_VLA.2.2d The developer shall provide vulnerability analysis documentation.
AVA_VLA.2.1c The vulnerability analysis documentation shall describe the analysis of the TOE deliverables performed to search for ways in which a user can violate the TSP.
AVA_VLA.2.2c The vulnerability analysis documentation shall describe the disposition of identified vulnerabilities.
AVA_VLA.2.3c The vulnerability analysis documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.
AVA_VLA.2.4c The vulnerability analysis documentation shall justify that the TOE, with the identified vulnerabilities, is resistant to obvious penetration attacks.
AVA_VLA.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AVA_VLA.2.2e The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure the identified vulnerabilities have been addressed.
AVA_VLA.2.3e The evaluator shall perform an independent vulnerability analysis.

AVA_VLA.2.4e The evaluator shall perform independent penetration testing, based on the independent vulnerability analysis, to determine the exploitability of additional identified vulnerabilities in the intended environment.

AVA_VLA.2.5e The evaluator shall determine that the TOE is resistant to penetration attacks performed by an attacker possessing a low attack potential.

6. TOE Summary Specification

This chapter describes the security functions and associated assurance measures.

6.1 TOE Security Functions

6.1.1 User data protection

The TSF shall ensure that all access to the following resources is forbidden (i.e., the TOE access rules do not allow access by default) for all users except explicitly allowed via corresponding role assignments or ownership:

- Web Modules,
- Portlet Application Definitions,
- Portlets,
- Content Nodes (Pages),
- User Groups and Users (implicitly protected via user groups),
- URL Mapping contexts,
- Templates,
- Template Categories,
- Policies, and
- WSRP producers.

The resources can be accessed by users using a Web browser, the XmlAccess configuration tool or portal scripting. In all three cases, the TOE expects the hosting WebSphere Application Server (WAS) to identify and authenticate the user and the associated WebSphere Member Manager (WMM) to provide user profile and group information.

When the TOE is invoked, it is provided the applicable user ID, profile, and group information that is used by its individual components or portlets to mediate access through corresponding calls to the TOE's (Portal Access Control) PAC component (see Figure 2.1). In other words, once users are identified/authenticated by WAS and their profiles and groups are determined by WMM, the user-targeted TOE component or portlet is invoked with that information. The PAC component is, in turn, invoked by that component or portlet (i.e., PAC client) to make access decisions based on that information and then the component or portlet holding (or serving as an interface to) the applicable resource enforces the PAC decision.

The assignment and revocation of roles to individual users and groups is managed within the TOE by users allowed to do so by corresponding role assignments (see Section 5.1.2.1) via the administration portlets *User and Group Permissions* and *Resource Permissions*, the XmlAccess configuration tool and portal scripting.

Roles are identified by a role type (representing a set of allowed actions) and a resource. For example a role can be identified by the role type *Editor* and the resource *Banking App*. This role would be named *Editor@Banking_App* (per convention) and contain permissions required to operate in the job responsibility of an editor on the *Banking App* resource itself, and via permission inheritance on the descendant resources of that resource in the hierarchy of protected resources. The resource hierarchy information, roles and role assignments are stored in 4 data domains in the WP database. Permission inheritance can be blocked on individual resources by role type specific role blocks. Role blocks can be either of type *propagation* or *inheritance*, which blocks propagating permissions from resource to its child resources or prevents the resource itself from inheriting permissions from its parent resource. In other words, when determining access for a resource access can be derived from its parents based on the hierarchical topology of resources, except that when a role block gets in the way – propagation blocks prevent permissions from being shared with a child and inheritance blocks prevent a child from seeking permissions from its parent. The access control decision is returned to the appropriate PAC client as a boolean yes/no. If successful then the PAC

client performs the operation requested and the results returned to the user interface. If access is denied then the PAC client either does not display the resource or an error is given, dependent upon the operation requested.

See section 6.1.2 for additional details - initial (default) access permissions, access operations, and security management related access controls.

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

- FDP_ACC.2: As indicates above, WP offers access to web modules, portlet application definitions, portlets, content nodes (pages), user groups, URL mapping contexts, templates, template categories, and policies all subject to the PAC.
- FDP_ACF.1: Access decisions for WP resources are based on permissions associated with user identities via their associated groups and roles. Users can access objects only if the access check is successful; meaning that they have permissions either due to being the owner of the resource or by inheriting permissions that have not been blocked from an ancestor of the resource in the resource topology.

6.1.2 Security management

Within the TOE, the notion of roles and role separation is based on the access control rules and enforcement. The TOE utilizes its PAC component where decisions are made with regard to accessing resources and enforced by the TOE resource containers, and thereby access to TOE configuration objects can be restricted. Note that while users are assigned to roles (and groups) via the WMM, once a user is identified and authenticated by WAS and groups and roles are determined by WMM, the TOE enforces all role restrictions for that user.

The Administrator role (i.e., Administrator@Portal) and Security Administrator role (i.e., Security Administrator@Portal) contain the (Grant Access On, (the virtual resource) portal) permission, which is not available to any other role. This permission is sufficient for being allowed to make arbitrary changes to the access control configuration of all resources that are internally managed by the portal, i.e., it is sufficient for viewing, creating and deleting arbitrary roles, role assignments, and inheritance blocks, and to change ownership of arbitrary resources.

The Access Control administration can be performed using corresponding portlets within the running portal, via the XmlAccess configuration tool or portal scripting. Accessing the portal with the XmlAccess tool requires the permissions (Grant Access On, (the virtual resource) portal) and (Grant Access On, (the virtual resource) XmlAccess).

In addition to the Administrator@Portal and Security Administrator@Portal roles, WebSphere Portal supports fine grained delegated access control administration. An administrator is a user who is authorized to modify the access control configuration for certain resources by changing role assignments, modifying role blocks and changing ownership. Administrators can delegate specific subsets of their administrative privileges to other users or groups. These users or groups can in turn delegate subsets of their privileges to additional users and groups. The delegated administration policy determines how users are permitted to delegate their privileges.

The general policy for creating, or deleting role assignments is as follows: A user P1 can create or delete a role assignment for a specific user or group P2 to a role identified by Role Type RT and resource RE if the following criteria is met: P1 has permission (Grant Access On, RE) and P1 has permissions equivalent to (RT, RE) and P1 has permission (Delegate To, P2)

For example, in order to assign Bob the Editor role on the Banking App resource, you must have been assigned at least Security Administrator role and Editor role on the Banking App resource (either directly or via inheritance) and Delegator role on one of the user groups containing Bob as a member.

The general policy for modifying role blocks is as follows: A user P can change the owner of RE from P1 to P2 if the following criteria is met: P has permissions on RE equivalent to those that come with being owner of RE and P has permission (Grant Access On, RE), and P has permission (Delegate To, P1) and P has permission (Delegate To, P2).

6.1.2.1 Initial and default access permissions

The table below describes the initial access control settings when the TOE is installed.

User or Group	Role
The administrative user identified during the installation	• Administrator@Portal
The administrative user group identified during the installation	• Administrator@Portal
All Identified Users	User@ the following portlet applications: • portletWiring Web Application • Edit page content and layout • Concrete Properties Web App • appearance Web Application • com.ibm.wps.portlets.palette Privileged User@ the following portlet applications: • Welcome • Information Portlet Application • wp.ap.selfcare • Bookmarks • My To Dos Portlet • ReminderPortlet • Newsgroups • Banner Ad • LotusNotesPortlet • Microsoft Exchange2000 • Microsoft Exchange2003 • World Clock • RSS Portlet • QuickLinks • ServletInvoker • JSPServer • FileServer • Document_Viewer_Portlet • CSV File Viewer • IBM Common Mail Portlet • IBM Common Calendar Portlet

User or Group	Role
	• MarketWatch Company Tracker • MarketWatch Retirement Planner • MarketWatch Currency Calculator • MarketWatch Portlets • peopleFinderJSR168.1 • com.ibm.wps.dm.6.1 • Web Content Management - Content Viewer • Search Admin Application • wp.ap.sitemap • ParamConfig Application • Properties Portlet Application • Roles Portlet Application • Community Portlet Application • Application Catalog Manager • SpellCheck application • SearchCenter portlet application User@ the following pages: • Open Tasks • Application Root • Application Membership • Application Roles • Directory Search • Organize Favorites • Page Customizer • Page Properties • People Finder • Quick Links • Template and Application Layout • Template and Application Properties Privileged User@ the following pages: • About • Edit My Profile • Search • Search Seedlist • Site Map

User or Group	Role
	• Personalization • Document Manager • Content Preview • People Palette • Search Center • Portlet Palette • Document Picker • Documents • Domino Integration • Home • Messaging • News • Spell Check Handler • Web Content
Anonymous User	User@ the following portlet applications: • wp.ap.login • wp.ap.selfcare • wp.ap.sitemap • Newsgroups • Banner Ad • World Clock • RSS Portlet • QuickLinks User@ the following pages: • Login • Search Seedlist • Edit My Profile • Welcome • Site Map

In addition, each user is implicitly allowed the actions contained in the role types User, Editor, and Privileged User on itself (e.g. a user is allowed to view and update her own phone number). There is no explicit role assignment for these actions. They are a part of the administration policy.

On creation of a resource, the TSF shall define default security attributes for access to that resource. Every time a protected resource is created within the portal, the user that created the resource becomes the owner of that resource (see Section 2.2.1.1.4). The owner of a resource is allowed to perform the following actions on the resource:

- Add Child (if the resource is a shared resource);

- Add Private Child (if the resource is a private resource);
- Delete;
- Edit (if the resource is a shared resource);
- Personalize (if the resource is a private resource);
- View.

In addition, the resource inherits the permissions assigned to the parent resource, unless propagation on the parent resource block is in place.

Private resources are those resources that can only be accessed by the owner of that resource. Therefore, add private child enables a private child resource to be created that only allows access to the owner of that resource. The action named personalize is the equivalent action as the action named edit (see Section 2.2.1.1.5), but for a private resource. Shared resources are resources that can be accessed by more than one user.

By default, no role blocks are set on a newly created resource.

6.1.2.2 Access operations and role types

Access operations (or actions) are provided as part of a set. The following actions are available:

- The *Grant Access On* action represents the activity of granting or revoking other principals access permissions to the access control configuration on a specific resource;
- The *Delegate To* action supports the activity of delegating a permission to a specific principal; For the complete set of actions necessary to allow a user to delegate a role assignment to a specific principal for a resource see the description of the Delegated Administrative Policy in section 6.1.2;
- The *Add Child* action represents the creation of a new, shared resource underneath an existing resource;
- The *Add Private Child* action represents the creation of a new private resource underneath an existing resource that can be accessed by a single user only;
- The *Delete* action represents the deletion of a resource or the removal of a resource from its parent resource (e.g. when a resource is moved from one place in the topology to an other);
- The *Edit* action represents all modifications to a resource (e.g. changing the meta-information of a resource) that are visible not only to the owner of a resource;
- The *Personalize* action represents all modifications to a resource that are only visible to the owner of a resource (this may imply the creation of an implicitly derived resource);
- The *View* action represents the presentation of the content or meta-information of a resource.
- The *Traverse* action represents being aware of the existence of a resource without being allowed to see the actual content of the resource. E.g. if a user has only traverse permission on a specific portal page, but no view permission on that page, the user will only be allowed to navigate to/through this page by seeing the page title as a label. The user will not see any contents of this page.

Actions are part of ‘role types’. A role type is a named set of actions that characterizes a specific way of interacting with resources and defines a corresponding type of roles. The TSF maintains the following role types to ensure secure operation of the TOE.

Role Type / Action	Administrator	Security Administrator	Delegator	Manager	Editor	Contributor	Privileged User	User
Grant Access On	All Actions	x						
Delegate To		x	x					
Add Child				x	x	x		
Add Private Child							x	
Delete				x				
Edit				x	x			
Personalize							x	
View				x	x	x	x	x
Traverse				x	x	x	x	x

6.1.2.3 Logging of security management functions

The TOE includes the ability to log security management functions. This feature is disabled by default and can be enabled as a installation/configuration option for the product.

When enabled, the TOE will generate records of security management functions, including assigning and revoking roles, modifying role blocks, and changing resource ownership. Each record will be created with at least the following content: user ID, applicable (i.e., when a resource is involved) resource IDs, the time and date (collected from the TOE environment), and identification of the function performed.

Note that in addition to the audit records indicated above, the TOE can generate additional events that are not related to the other security requirements, and as a result are not included in the logging security requirement. They are related to use of TOE interfaces regardless of specific security relevance, including interfaces that serve to pass requests on to the environment components supporting the TOE (e.g., WMM). The additional events include: creating, modifying and deleting users and groups; creating, modifying and deleting portlet applications by using the portal user interface; creating, modifying and deleting protected resources; installing and uninstalling web modules; creating and deleting application roles; assigning and revoking application roles to and from users; adding and deleting roles to application roles; and initializing a database domain.

Each generated record is written into a file designated during installation/configuration. The resulting file can be accessed via any suitable means available in the IT environment and is created as a list of ASCII-formatted events.

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

- FMT_LOG_EX.1: The TOE generates the required events with the required contents.
- FMT_MSA.1: WP controls the ability to perform security management operations based on the role of the user. While users being granted Administrator@Portal or Security Administrators@Portal roles can change any security attributes, other users can perform security management functions based on their corresponding permissions.
- FMT_MSA.3: WP comes with an initial set of objects and access permissions and subsequently objects are assigned restrictive default values based on propagation, inheritance and ownership. Note that the access on new objects is restricted based on access available from the parent or via ownership and the TOE provides no means to specify alternate access permissions when an object is being created.

- FMT_SMF.1: WP provides operations to assign and revoke roles to user and groups, modifying role blocks, and change ownership.
- FMT_SMR.1: WP includes a number of role types (Administrator, Security Administrator, Delegator, Manager, Editor, Contributor, Privileged User, and User) representing different combinations of actions that can be used to define role instances that can be granted to users and groups.

6.1.2.4 Protection of the TSF

All of the TOE components are WAS applications that serve to host resources and make those resources available via their interfaces accessible indirectly through WAS. As such, the TOE relies upon WAS to provide a secure execution domain for its components and the resources they hold. The TOE components are defined within WAS such that they expose, to TOE users, only specific interfaces while other interfaces are limited to access only by other TOE components.

As explained in the previous sections, the TOE implements a central access control decision functions within its PAC component. This component is invoked (using interfaces made available to other TOE components by virtue of its definition within WAS) by all other WP components (see list in Section 2.1) when any attempt is made to access a WP resource (see list in Section 2.2) using an available interface to the applicable TOE component. The resulting access decisions are enforced by the component that is servicing the access attempt by a TOE user. As such, WP ensures that its own security policies cannot be bypassed.

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

- FPT_RVM.1a: WP provides a central access decision function and decentralized enforcement of those decisions to ensure that its policies cannot be bypassed.

6.2 TOE Security Assurance Measures

6.2.1 Configuration management

The configuration management measures applied by IBM ensure that configuration items are uniquely identified, and that documented procedures are used to control and track changes that are made to the TOE. IBM ensures changes to the implementation representation are controlled with the support of automated tools and that TOE associated configuration item modifications are properly controlled. IBM performs configuration management on the TOE implementation representation, design documentation, tests and test documentation, user and administrator guidance, delivery and operation documentation, life-cycle documentation, vulnerability analysis documentation, configuration management documentation, and security flaws.

These activities are documented in:

- IBM WebSphere Portal Configuration Management Plan

The Configuration management assurance measure satisfies the following EAL 4 augmented with ALC_FLR.2 assurance requirements:

- ACM_AUT.1
- ACM_CAP.4
- ACM_SCP.2

6.2.2 Delivery and operation

IBM provides delivery documentation and procedures to identify the TOE, allow detection of unauthorized modifications of the TOE and installation and generation instructions at start-up. IBM's delivery procedures describe all applicable procedures to be used to detect modification to the TOE. IBM also provides documentation that describes the steps necessary to install WebSphere Portal in accordance with the evaluated configuration.

These activities are documented in:

- IBM WebSphere Portal Installation and Delivery Guide

The Delivery and operation assurance measure satisfies the following EAL 4 augmented with ALC_FLR.2 assurance requirements:

- ADO_DEL.2
- ADO_IGS.1

6.2.3 Development

IBM has numerous documents describing all facets of the design of the TOE. In particular, they have a functional specification that describes the accessible TOE interfaces; a high-level design that decomposes the TOE architecture into subsystems and describes each subsystem and their interfaces; a low-level design that further decomposes the TOE architecture into modules and describes each module and their interfaces; and, correspondence documentation that explains how each of the design abstractions correspond from the TOE summary specification in the Security Target to the actual implementation of the TOE. Furthermore, IBM has a security model that describes each of the security policies implemented by WebSphere Portal. Of course, the implementation of the TOE itself is also available as necessary.

These activities are documented in:

- IBM WebSphere Portal Functional Specification
- IBM WebSphere Portal High-level Design
- IBM WebSphere Portal Low-level Design
- IBM WebSphere Portal source code
- IBM WebSphere Portal Security Policy Model

The Development assurance measure satisfies the following EAL 4 augmented with ALC_FLR.2 assurance requirements:

- ADV_FSP.2
- ADV_HLD.2
- ADV_IMP.1
- ADV_LLD.1
- ADV_RCR.1
- ADV_SPM.1

6.2.4 Guidance documents

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

These activities are documented in:

- IBM WebSphere Portal Administration Guide

The Guidance documents assurance measure satisfies the following EAL 4 augmented with ALC_FLR.2 assurance requirements:

- AGD_ADM.1
- AGD_USR.1

6.2.5 Life cycle support

IBM ensures the adequacy of the procedures used during the development and maintenance of the TOE through the use of a comprehensive life-cycle management plan. IBM applies security controls on the development environment that are adequate to provide the confidentiality and integrity of the TOE design and implementation that is necessary to ensure the secure development of the TOE. IBM has a documented model of the TOE life cycle that ensures that the TOE is developed and maintained in a well-defined manner. IBM uses well-defined development tools in order to ensure consistent and predictable results while developing the TOE. In addition, IBM identifies and tracks reported flaws, ensuring that they are addressed and corrections and corrective measures are made available as applicable.

These activities are documented in:

- IBM WebSphere Portal Life-cycle Document

The Life cycle support assurance measure satisfies the following EAL 4 augmented with ALC_FLR.2 assurance requirements:

- ALC_DVS.1
- ALC_FLR.2
- ALC_LCD.1
- ALC_TAT.1

6.2.6 Tests

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

These activities are documented in:

- IBM WebSphere Portal Test Plan
- IBM WebSphere Portal Test Procedures
- IBM WebSphere Portal Test Results

The Tests assurance measure satisfies the following EAL 4 augmented with ALC_FLR.2 assurance requirements:

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

6.2.7 Vulnerability assessment

The TOE administrator and user guidance documents describe the operation of WebSphere Portal and how to maintain a secure state. These guides also describe all necessary operating assumptions and security requirements outside the scope of control of the TOE. They have been developed to serve as complete, clear, consistent, and reasonable administrator and user references. Furthermore, IBM has conducted a misuse analysis demonstrating that the provided guidance is complete.

IBM has conducted a strength of function analysis wherein it was concluded that the TOE includes no permutational or probabilistic security mechanisms. As such the TOE automatically fulfills the minimum strength of function claim, SOF-medium, that is technically not applicable (see also section 8.4).

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

These activities are documented in:

- IBM WebSphere Portal Vulnerability Analysis

The Vulnerability assessment assurance measure satisfies the following EAL 4 augmented with ALC_FLR.2 assurance requirements:

- AVA_MSU.2
- AVA_SOF.1
- AVA_VLA.2

7. Protection Profile Claims

There are no Protection Profile claims in this Security Target.

8. Rationale

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

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

8.1 Security Objectives Rationale

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

8.1.1 Security Objectives Rationale for the TOE and Environment

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

	P.ACCESS	T.ACCESS_RES	T.APP	A.ADMIN	A.APP
O.ACCESS	X	X			
O.MANAGE	X	X			
O.CONFIG	X		X		X
O.ADMIN	X		X	X	X
O.APP	X		X		X
O.RECOVER		X	X		

Table 4 Environment to Objective Correspondence

8.1.1.1 P.ACCESS

The right to access a resource is determined on the basis of: the user groups the user is a member of; the role instances assigned to those user groups or to the user; the role instances contained in application roles assigned to those user groups or to the user; the permissions contained in the individual assigned role instances which are determined on the basis of the actions contained in the corresponding role type and the set of descendant resources in the role domain which is determined in the basis of the topology defining the

parent child relation ship among the resources and the role blocks that exist on those resources; and, the set of resources owned by the user or the user groups the user is a member of.

This Organizational Policy is satisfied by ensuring that:

- O.ACCESS provides the means of controlling access to objects by users and processes.
- O.MANAGE supports this policy by the administrators ensuring that the policy is maintained.
- O.ADMIN, O.CONFIG and O.APP further support this policy by ensuring that the applications are configured in a secure manner so that no vulnerability may exist that enables an unauthorized user to gain an authorized identity. Further, O.CONFIG ensures that the TOE has appropriate time stamps to use in conjunction with logging security management functions.

8.1.1.2 T.ACCESS_RES

A user of the TOE gains access to an object without the correct authority to access that object.

This Threat is satisfied by ensuring that:

- O.ACCESS counters this directly by ensuring that only those users with the correct authority can access an object.
- O.MANAGE ensures that privileged actions are performed effectively.
- O.PROTECT ensures that no resources can be accessed via the cabling between the workstations on which the TOE is installed.
- O.RECOVER ensures that following a system failure, the TOE is not operating in an insecure state whereby an unauthorized user can gain access to objects they are not authorized to access.

8.1.1.3 T.APP

The applications that the TOE depends upon might be compromised by a potential attacker.

This Threat is satisfied by ensuring that:

- O.APP, O.CONFIG, O.PROTECT and O.RECOVER together ensure that the applications are managed in a secure manner.
- O.ADMIN supports this threat by ensuring that the administrator is a competent individual who will apply the latest patch information within the environment and therefore ensuring that any vulnerabilities that may compromise the security of the applications that become known, will be countered.

8.1.1.4 A.ADMIN

It is assumed that there are one or more competent individuals that are assigned to manage the TOE and the security of the information it contains. Such personnel are assumed not to be careless, willfully negligent or hostile.

This Assumption is satisfied by ensuring that:

- O.ADMIN ensures that the administrator is a competent and trustworthy person who is capable of managing the TOE in a secure manner.

8.1.1.5 A.APP

It is assumed that the applications that the TOE relies upon, have been configured in accordance with the manufacturer's installation guides and where applicable, in its evaluated configuration. It is securely configured such that the applications protect the TOE from any unauthorized users or processes.

This Assumption is satisfied by ensuring that:

- O.ADMIN and O.CONFIG support this by ensuring that the Administrator is a competent and trustworthy person and that the users have been set up appropriately.
- O.APP is the primary environmental objective that satisfies the assumption. This ensures that the administrator installs and configures the supporting applications in accordance with: the manufacturer instructions and any evaluated configurations were applicable.

8.2 Security Requirements Rationale

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

8.2.1 Security Functional Requirements Rationale

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

	O.ACCESS	O.MANAGE	O.CONFIG
FDP_ACC.2	X		
FDP_ACF.1	X		
FMT_LOG_EX.1		X	
FMT_MSA.1	X	X	
FMT_MSA.3	X	X	
FMT_SMF.1		X	
FMT_SMR.1		X	
FPT_RVM.1a	X	X	
FIA_ATD.1			X
FIA_UAU.1			X
FIA_UID.1			X
FPT_RVM.1b			X
FPT_SEP.1			X
FPT_STM.1			X

Table 5 Objective to Requirement Correspondence

8.2.1.1 O.ACCESS

The TOE must ensure that only those users with the correct authority are able to access a resource.

This TOE Security Objective is satisfied by ensuring that:

- The TOE must ensure that only those users with the correct authority are able to access a resource. The access control mechanism must have a defined scope of control [FDP_ACC.2] with defined rules [FDP_ACF.1].
- Authorized users must be able to control who has access to the objects [FMT_MSA.1].
- Protection of these objects must be continuous, starting from object creation [FMT_MSA.3].
- The access control mechanism must not be bypassable [FPT_RVM.1a]

8.2.1.2 O.MANAGE

The TOE must allow administrators to effectively manage the TOE and that this is only performed by authorized users.

This TOE Security Objective is satisfied by ensuring that:

- The TOE must be able to log security management functions when configured to do so in order to have a record of security management activitiers [FMT_LOG_EX.1].

- The TSF must enable an authorized administrator to manage the TOE in accordance with the access control SFP [FMT_MSA.1].
- On creation of resources default values will be used, which enables ease of management [FMT_MSA.3].
- [FMT_SMF.1] specifies the management functions provided by the TOE.
- [FMT_SMR.1] defines roles in order that the TOE is managed effectively.
- The access control mechanism, used to restrict management functions, must not be bypassable [FPT_RVM.1a]

8.2.1.3 O.CONFIG

The IT environment must ensure that users of the TOE have associated user IDs and where applicable have an associated Group ID, that reliable time information is available to the TOE, and also that the TOE execution environment is appropriately protected from bypass or other penetration attempts.

This IT Environment Security Objective is satisfied by ensuring that:

- [FIA_ATD.1] on the IT environment ensures that the user and group IDs of users are maintained within the environment.
- [FIA_UAU.1] on the IT environment ensures that each identified user shall be successfully authenticated.
- [FIA_UID.1] on the IT environment ensures that each user shall be successfully identified and allows the assumption of the anonymous user ID. Therefore each user on the system would be identified either by the unique ID supplied by WAS, or by the anonymous user ID.
- [FPT_RVM.1b] on the IT environment ensures that the IT environment as well as the TOE are appropriately protected from potential bypass of the security mechanisms.
- [FPT_SEP.1] on the IT environment ensures that the IT environment as well as the TOE are appropriately protected from potential attacks.
- [FPT_STM.1] on the IT environment ensures that the IT environment will provide reliable time stamps for use by the TOE. Note that this is included to support FMT_LOG_EX.1.

8.3 Security Assurance Requirements Rationale

The TOE is intended for an environment requiring a moderate to high level of assurance in the security functionality of conventional commodity TOEs, as presented in the statement of security environment (Section 3). The target assurance level of EAL 4 is appropriate for such an environment. The base EAL 4 assurance target has been augmented with ALC_FLR.2 (flaw remediation) since it is also important to ensure that the product will be revised to address any security flaws that might be identified.

8.4 Strength of Functions Rationale

In accordance with EAL 4 augmented with ALC_FLR.2, a Strength of Functions claim of SOF-medium has been made. EAL 4 augmented with ALC_FLR.2 represents a moderate to high level of security assurance and hence SOF-medium should represent an appropriate strength of function. Note that there are no permutational or probabilistic mechanisms in the TOE. Hence, there are no applicable SFRs.

8.5 Requirement Dependency Rationale

The following table identifies the dependencies of the requirements in this ST, including the requirements explicitly defined in this ST. As indicated in the table, all of the dependencies are satisfied.

ST Requirement	CC Dependencies	ST Dependencies
FDP_ACC.2	FDP_ACF.1	FDP_ACF.1
FDP_ACF.1	FDP_ACC.1 and FMT_MSA.3	FDP_ACC.2 and FMT_MSA.3
FMT_LOG_EX.1	FPT_STM.1	*FPT_STM.1*
FMT_MSA.1	FMT_SMR.1 and FMT_SMF.1 and (FDP_ACC.1 or FDP_IFC.1)	FMT_SMR.1 and FMT_SMF.1 and FDP_ACC.2
FMT_MSA.3	FMT_MSA.1 and FMT_SMR.1	FMT_MSA.1 and FMT_SMR.1

FMT_SMF.1	none	none
FMT_SMR.1	FIA_UID.1	*FIA_UID.1*
FPT_RVM.1a	none	none
FIA_ATD.1	none	none
FIA_UAU.1	FIA_UID.1	*FIA_UID.1*
FIA_UID.1	none	none
FPT_RVM.1b	none	none
FPT_SEP.1	none	none
FPT_STM.1	none	none
ACM_AUT.1	ACM_CAP.3	ACM_CAP.4
ACM_CAP.4	ALC_DVS.1	ALC_DVS.1
ACM_SCP.2	ACM_CAP.3	ACM_CAP.4
ADO_DEL.2	ACM_CAP.3	ACM_CAP.4
ADO_IGS.1	AGD_ADM.1	AGD_ADM.1
ADV_FSP.2	ADV_RCR.1	ADV_RCR.1
ADV_HLD.2	ADV_FSP.1 and ADV_RCR.1	ADV_FSP.2 and ADV_RCR.1
ADV_IMP.1	ADV_LLD.1 and ADV_RCR.1 and ALC_TAT.1	ADV_LLD.1 and ADV_RCR.1 and ALC_TAT.1
ADV_LLD.1	ADV_HLD.2 and ADV_RCR.1	ADV_HLD.2 and ADV_RCR.1
ADV_RCR.1	none	none
ADV_SPM.1	ADV_FSP.1	ADV_FSP.2
AGD_ADM.1	ADV_FSP.1	ADV_FSP.2
AGD_USR.1	ADV_FSP.1	ADV_FSP.2
ALC_DVS.1	none	none
ALC_FLR.2	none	none
ALC_LCD.1	none	none
ALC_TAT.1	ADV_IMP.1	ADV_IMP.1
ATE_COV.2	ADV_FSP.1 and ATE_FUN.1	ADV_FSP.2 and ATE_FUN.1
ATE_DPT.1	ADV_HLD.1 and ATE_FUN.1	ADV_HLD.2 and ATE_FUN.1
ATE_FUN.1	none	none
ATE_IND.2	ADV_FSP.1 and AGD_ADM.1 and AGD_USR.1 and ATE_FUN.1	ADV_FSP.2 and AGD_ADM.1 and AGD_USR.1 and ATE_FUN.1
AVA_MSU.2	ADO_IGS.1 and ADV_FSP.1 and AGD_ADM.1 and AGD_USR.1	ADO_IGS.1 and ADV_FSP.2 and AGD_ADM.1 and AGD_USR.1
AVA_SOF.1	ADV_FSP.1 and ADV_HLD.1	ADV_FSP.2 and ADV_HLD.2
AVA_VLA.2	ADV_FSP.1 and ADV_HLD.2 and ADV_IMP.1 and ADV_LLD.1 and AGD_ADM.1 and AGD_USR.1	ADV_FSP.2 and ADV_HLD.2 and ADV_IMP.1 and ADV_LLD.1 and AGD_ADM.1 and AGD_USR.1

8.6 Explicitly Stated Requirements Rationale

This Security Target includes a single explicitly stated security functional requirement: FMT_LOG_EX.1. This requirement is added to the FMT class since it is strictly for the purpose of requiring audit of security management functions. It is explicitly stated since the TOE does not satisfy the CC FAU_GEN.1 requirement which dictates that enabling and disabling of audit must be audited. Otherwise, this explicit requirement is similar to FAU_GNE.1 and as such shares its dependency – FPT_STM.1 – which has been assigned to the IT environment of the TOE.

8.7 TOE Summary Specification Rationale

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

requirements. Furthermore, all of the security functions are necessary in order for the TSF to provide the required security functionality.

This Section in conjunction with Section 6, the TOE Summary Specification, provides evidence that the security functions are suitable to meet the TOE security requirements. The collection of security functions work together to provide all of the security requirements. The security functions described in the TOE summary specification are all necessary for the required security functionality in the TSF. **Table 6 Security Functions vs. Requirements Mapping** demonstrates the relationship between security requirements and security functions.

	User data protection	**Security management**	**Protection of the TSF**
FDP_ACC.2	X		
FDP_ACF.1	X		
FMT_LOG_EX.1		X	
FMT_MSA.1		X	
FMT_MSA.3		X	
FMT_SMF.1		X	
FMT_SMR.1		X	
FPT_RVM.1a			X

Table 6 Security Functions vs. Requirements Mapping

8.8 PP Claims Rationale

See Section 7, Protection Profile Claims.