WebSphere MQ
EAL4
Security Target

An Employee-Owned Company

Common Criteria Testing Laboratory
7125 Columbia Gateway Drive, Suite 300
Columbia, MD 21046

Date:	25 July 2006
Issue:	1.0
Reference:	LFF/MQ/EAL4/ST/28

© Copyright IBM 2006

This Page Intentionally Left Blank.

Table of Contents

© Copyright IBM 2006

Glossary and Terminology

ACL	Access Control List
Administrator	A user with membership to the MQM administrator group within the operating system
AuthInfo Object	An Authentication Information (AuthInfo) object contains authentication information used in Transport Layer Security (TLS) or the Secure Sockets Layer (SSL) encrypted transport of information.
Authorised User	A user who may, in accordance with the TSP, perform an operation.
CC	Common Criteria
Channel	Channels are objects that provide a communication path from one queue manager to another.
CipherSpec	On a TLS/SSL connection, the CipherSpec identifies the combination of the encryption algorithm used to encipher data and the Message Authentication Code algorithm used to generate the message digest. The CipherSpec forms part of the CipherSuite.
CipherSuite	A CipherSuite is a suite of cryptographic algorithms used by an TLS/SSL connection. It comprises of the key exchange and authentication algorithm used during the TLS/SSL handshake, the encryption algorithm to encipher data and the Message Authentication Code algorithm used to generate the message digest.
DAP	Data Abstraction and Persistence
EAL	Evaluation Assurance Level
FIFO	First In First Out
FIPS	Federal Information Processing Standard
GSKit	Global Security Kit
IT	Information Technology
ITSEC	IT Security Evaluation Criteria
JMS	Java Message Service
JSSE	Java Secure Sockets Extension
MCA	Message Channel Agent. A program that transmits prepared messages from a transmission queue to a communication link, or from a communication link to a destination queue.
Message	A *message* is a string of bytes that is meaningful to the applications that use it. Messages are used to transfer information from one application program to another (or between different parts of the same application).

MQ	Message Queue
MQI	Message Queue Interface
MQSC	Message Queue Script Commands
MDV	Maintenance Delivery Vehicle. This is a fix pack or a refresh pack. A fix pack contains only fixes, a refresh pack contains fixes and new function. MDVs are cumulative e.g. all fixes within MDV5 are contained within MDV6.
Namelist	A namelist is a WebSphere MQ object that contains a list of cluster names, queue names or authentication information object names. In a cluster, it can be used to identify a list of clusters for which the queue manager holds the repositories.
OAM	Object Authority Manager
Object	Objects are queue managers, queues, process definitions, namelists, authentication information objects, channel objects, clntconn channel objects, service objects and listener objects.
OS	Operating System
OSP	Organisational Security Policy
PCF	Programmable Command Format
Process Definitions	A Process Definition object defines the attributes of an application for the queue manager so that the queue manager can start the application without operator intervention. e.g. at the occurrence of a trigger.
Queue	A *queue* is a data structure used to store messages. Each queue is owned by a *queue manager*.
Queue Manager	A Queue Manager supplies an application with WebSphere MQ services. The Queue Manager is responsible for maintaining the queues it owns, and for storing all the messages it receives onto the appropriate queues.
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
SOF	Strength of Function
SSL	Secure Sockets layer
ST	Security Target
TLS	Transport Layer Security
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 Function. 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.

1 Introduction

This document is the Security Target (ST) for the Common Criteria (CC) evaluation of WebSphere MQ version 6.

This document is the WebSphere MQ EAL4 Security Target, version 1.0 and dated 25 July 2006.

1.1 Overview

IBM WebSphere® MQ is message queuing middleware. It connects all business software together to form one enterprise by providing an open, scalable, industrial-strength messaging backbone.

WebSphere MQ (WMQ) is divided into the operating system specific editions and the specific version for this evaluation is WebSphere MQ for AIX...

Each of the operating system specific Editions can support the following components:

- WMQ server (which includes the queue manager);
- WMQ C Client and
- JMS/Java clients

In addition, to the above components, there are tools and utilities to enable third party development of applications. These applications are often referred to as ‘MQ Applications’. However, these are not within the scope of the evaluation.

The AIX v5.2 operating system (OS) is supported within this evaluation

It is assumed that all hardware used within the operating environment is secured such that no potential vulnerabilities could be introduced that would circumvent the functionality described within this ST.

The WebSphere MQ Server and WMQ C Client use the GSKit software to enable support for TLS/SSL. GSKit is a set of tools and C/C++ programming interfaces that can be used to add secure channels using the SSLv3 and TLSv1 protocols to TCP/IP applications (products). It provides the cryptographic functions, the protocol implementation and key generation and management functionality for this purpose.

The GSKit software must be configured such that

- only SSLv3 and TLSv1 are allowed, other versions of SSL are disabled,
- only SSL/TLS CipherSuites whose CipherSpec parts consist of cryptographic algorithms that are FIPS 140-2 [FIPS140-2] approved and
- the FIPS 140-2 [FIPS140-2] approved random number generator

are used. This configuration is called FIPS mode. Please note that the cryptographic algorithms used for key exchange are not affected by these restrictions.

WebSphere MQ must initialize and use the GSKit software in FIPS mode.

The environment must provide reliable timestamps for certificate and CRL verification.

The processing resources of the TOE must be located within controlled access facilities.

The *GSKit* digital certificates and keys necessary for the TLS/SSL channel are maintained in a key database file. This key database file must be protected. This includes appropriate protection of backup copies. In addition the passwords to access the key database file are kept in a password stash file so that the digital keys can be accessed programmatically and for configurations that do not have a human operator present for normal operations. This password stash file must be protected by an ACL while it resides on the system and by encryption when backed up.

The use of certificate revocation lists (CRLs) for certificate validation is optional (but recommended in [SSLv3] and [TLSv1]). The TOE can retrieve a CRL using LDAP, check that the signatures are valid, and, if they are, will use this CRL for certificate validation. If CRLs are to be used, an LDAP client and an LDAP server that provides a current CRL, which contains all revoked certificates, must be available in the TOE environment.

The GSKit in the evaluated configuration must not use

- non-FIPS approved cryptographic functions of ICC V1.2.1 for the CipherSpec part of SSL/TLS CipherSuites,
- SSL versions prior to 3 (due to known weaknesses),
- “total anonymity mode” for SSL/TLS (i.e. server authentication is mandatory), and
- BSAFE cryptographic library (by RSA).

WebSphere MQ classes for Java (also referred to as WebSphere MQ base Java) allow a Java application to connect to WebSphere MQ as a WebSphere MQ client or connect directly to a WebSphere MQ server. WebSphere MQ classes for Java Message Service (also referred to as WebSphere MQ JMS) are a set of Java classes that implement Sun’s Java Message Service (JMS) interfaces to enable JMS programs to access WebSphere MQ systems. To support WebSphere MQ JMS applications, a WebSphere MQ base client must include WebSphere MQ base Java. A WebSphere MQ Java application can use TLS/SSL to obtain a secure connection to a queue manager, with authentication, message integrity, and data encryption. . The IBM® Java JSSE FIPS provider (IBMJSSEFIPS) provides the ability to use TLS/SSL connections with cryptographic modules that have been FIPS 140-2 certified

1.2 Description

WebSphere MQ (WMQ) allows application programs to use *message queuing* to participate in message-driven processing. Application programs can communicate across different platforms by using WMQ. For example, AIX and Sun Solaris applications can communicate through WebSphere MQ. The applications are shielded from the mechanics of the underlying communications.

Messages are used to transfer information from one application program to another (or between different parts of the same application). The applications can be running on the same platform, or on different platforms.

Each queue is owned by a *queue manager*. The queue manager is responsible for maintaining the queues it owns, and for storing all the messages it receives onto the appropriate queues. The messages might be put on the queue by application programs, or by a queue manager as part of its normal operation.

The TOE is indicated by the thick black line shown in Figure 2.1 of this ST.

Figure 1.1 shows an example of how different servers and clients can communicate with one another. The operating systems on each of these can be independent to one another and the applications can reside on the same machine as the queue manager without the requirement for a client.

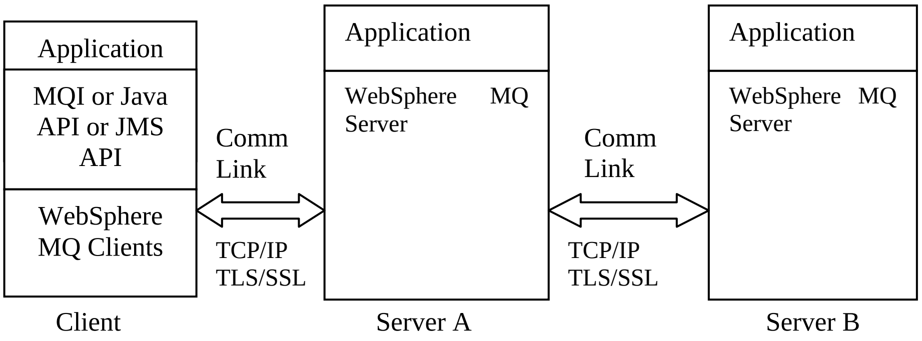

Figure 1.1 Connections between MQ servers and WMQ clients

1.2.1 WMQ Server

The WMQ server contains the queue manager, which is responsible for maintaining the queues that it owns, and for storing all the messages it receives onto the appropriate queues. The server contains components that:

- Interface with the operating system to retrieve information (Common Services),
- Provides a command line interface for administration of the queues; and
- Interface to remote queue managers (Message Channel Agent (MCA)). This component is responsible for sending and receiving of messages to remote queues. Messages are transmitted between queue managers on a *channel*. *Channels* are objects that provide a communication path from one queue manager to another; this can be secured using TLS or SSL protocols.

1.2.2 WMQ Clients

There are three types of WMQ clients. These are the WMQ C Client, the WMQ base Java client and the WMQ JMS Client. The functionality of the WMQ JMS client builds on and extends the functionality contained within the WMQ Java client. Even though we are talking about two distinct clients the channel-level internals are shared by the WMQ Java client and the WMQ JMS client. For the purposes of this document we will refer to these clients as the JMS/Java clients.

1.2.2.1 WMQ C Client

The WMQ C client is part of the WebSphere MQ product that can be installed on its own, on a separate machine from the server. A WMQ user application can be built and run on a WMQ C client system and it can interact with one or more WMQ servers and can connect to their queue managers by transmitting messages on channels that may be secured using TLS or SSL protocols. The servers to which the client connects might or might not be part of a cluster. The IBM® Global Security Kit (GSKit) provides the ability to use TLS/SSL connections with cryptographic modules that have been FIPS 140-2 certified. WMQ C client applications use the MQI.

1.2.2.2 WMQ JMS and Base Java Clients

WebSphere MQ classes for Java (also referred to as WebSphere MQ base Java or WebSphere MQ Java) allow a Java application to connect to WebSphere MQ as a WebSphere MQ client or connect directly to a WebSphere MQ server. WebSphere MQ base Java encapsulates the Message Queue Interface (MQI), the native WebSphere MQ API.

WebSphere MQ classes for Java Message Service (also referred to as WebSphere MQ JMS) are a set of Java classes that implement Sun's Java Message Service (JMS) interfaces to enable JMS programs to access WebSphere MQ systems. As with WMQ classes for Java, WMQ classes for JMS allow a WMQ JMS application to connect to WMQ either as a WMQ client or directly to a WMQ server.

WebSphere MQ JMS and Java clients can be installed as WebSphere MQ clients either on the WebSphere MQ server machine or on a separate machine. A WebSphere MQ Java application can use TLS/SSL to obtain a secure connection to a queue manager, with authentication, message integrity, and data encryption. The IBM® Java JSSE FIPS provider (IBMJSSEFIPS or IBMJSSE2) provides the ability to use TLS/SSL connections with cryptographic modules that have been FIPS 140-2 certified. The Java JSSE FIPS provider is outside the TOE boundary.

Note also that the Java and JMS interfaces can also be configured locally to access a queue manager. This is achieved using a JNI layer which wrappers the C MQI. An application chooses to use this local configuration by selecting a transport type called "bindings". Since the JNI Layer wraps the C MQI, the description of the MQI also applies to the base Java and JMS APIs in the bindings mode. The Java and JMS clients can also use a third transport mode called Direct IP. This 'direct' transport mode is excluded from the TOE.

1.2.3 Message Queue Interface (MQI)

WMQ provides the *Message Queue Interface* (MQI), a common application-programming interface available wherever the applications run. This makes it easier to port application programs from one platform to another and enables the MQI to be running on a separate machine to the queue manager.

1.3 CC Conformance

This ST is [CC] *Part 2 extended with* FAU_GEN_MQ.1 and FMT_MSA_MQ.3 *and Part 3 conformant*, with a claimed Evaluation Assurance Level of EAL4 augmented with ALC_FLR.2.

1.4 Strength of Functions

There is no strength of function claim because the TOE does not identify any security functional requirements for which an explicit Strength Of Function (SOF) is appropriate and does not identify any functions that are of a permutational or probabilistic nature. Therefore, a minimum SOF claim is not included for the TOE.

1.5 References

[CC] Common Criteria for Information Technology Security Evaluation, Version 2.3, August 2005.

1.6 Structure

The structure of this document is as defined by [CC] Part 1, Annex C:

- Section 2 is the TOE description;
- Section 3 provides a statement of the TOE security environment;
- Section 4 provides the statement of IT security objectives;
- Section 5 provides a statement of IT security requirements;
- Section 6 provides the TOE summary specification, which includes the detailed specification of the IT functions; and
- Section 7 provides the rationale for the security objectives, security requirements and TOE summary specification.

2 TOE Description

The diagram below shows the physical scope and boundaries in relation to the components within the TOE. It should be noted that the TOE includes the WMQ C Client and the WMQ JMS/Java clients. The TOE is a subset of the product and the solid line within figure 2.1 illustrates the boundary of the TOE in relation to the components. It should be noted that the applications, application programs and operating systems referred to within this ST are outside the scope of the evaluation.

Administrative commands on the queue manager can be supplied on the command line using Message Queue Script Commands (MQSC), or using the WMQ Explorer GUI or from an application using the Programmable Command Format (PCF). Henceforth, in this document, a reference made to an MQSC command implicitly includes other ways to submit the command. The JMSAdmin tool is available for users of the WebSphere MQ classes for JMS to manage and administer JMS objects in JNDI. Note that the WMQ Explorer is not included within the scope of this evaluation.

The WebSphere MQ Server product and the WebSphere MQ C Client use the IBM® Global Security Kit's TLS/SSL API to request TLS/SSL connections. The JMS/Java Clients use the IBM® Java JSSE FIPS (IBMJSSEFIPS) or the IBMJSSE2 providers to request TLS/SSL connections. Only FIPS 140-2 certified TLS/SSL cipher specs are permitted within the TOE. GSKit version 7.0.3.18 is within the TOE boundary and was evaluated as a component TOE evaluation. As such, this evaluation will reuse the GSKit component evaluation and not perform any further evaluation. JSSE is outside the TOE boundary.

A channel secured using SSL or TLS can use differing authentication, encryption and hashing algorithms. This is accomplished by specifying a CipherSpec on the channel definition on the WMQ server or WMQ C client and by specifying a CipherSuite on the JMS or Java client channel. WMQ supports several CipherSuites and CipherSpecs. However, EAL4 certification can only be obtained for those that are FIPS 140-2 certified. Hence WMQ has to run in FIPS mode

The CipherSpec parts of the TLS/SSL CipherSuites provided by the GSKit operating in FIPS mode contain algorithms that are FIPS 140-2 approved (ref: FIPS certificate 384). The following CipherSuites are supported:

- TLS_RSA_WITH_DES_CBC_SHA
- TLS_RSA_WITH_3DES_EDE_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA
- SSL_RSA_FIPS_WITH_DES_CBC_SHA
- SSL_RSA_FIPS_WITH_3DES_EDE_CBC_SHA

The following CipherSuites are supported by the JMS/Java clients. Note that all of these CipherSuites specify TLS authentication.

- SSL_RSA_WITH_AES_128_CBC_SHA

- SSL_RSA_WITH_AES_256_CBC_SHA
- SSL_RSA_WITH_DES_CBC_SHA
- SSL_RSA_WITH_3DES_EDE_CBC_SHA

© Copyright IBM 2006

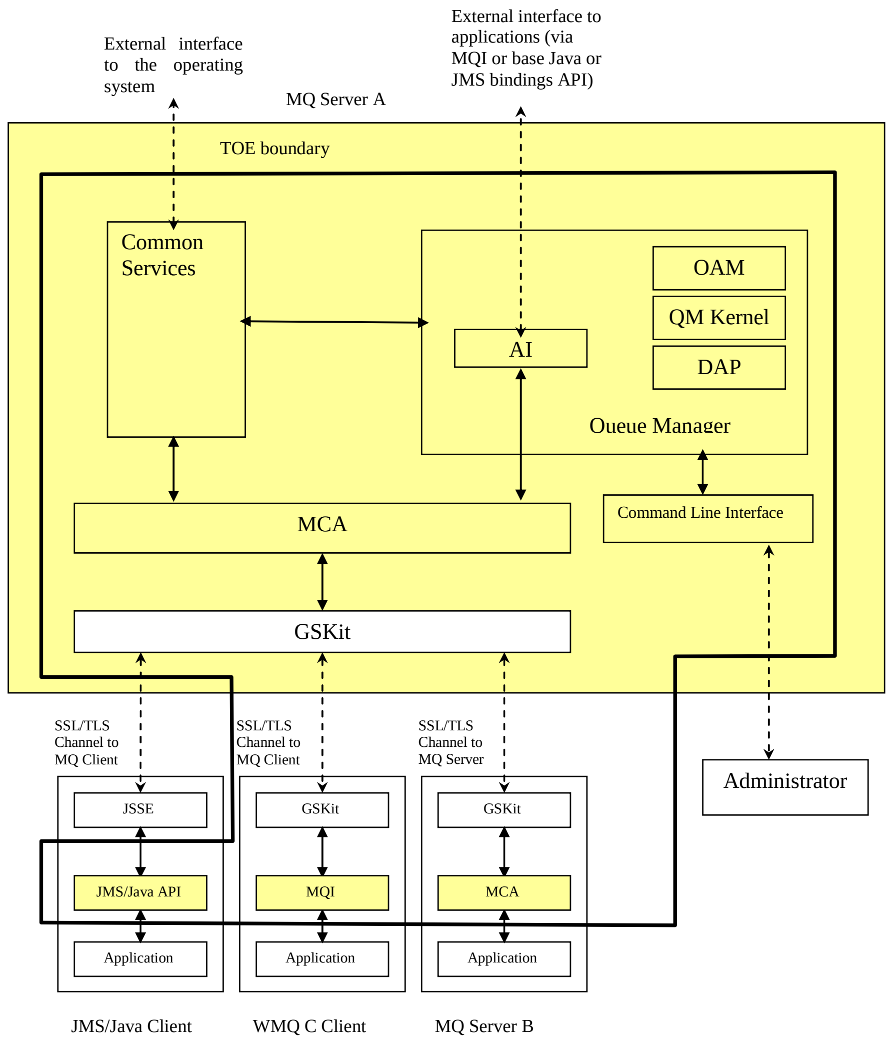

Figure 2.1: TOE boundary

2.1 Queue Manager

A *queue manager* is a system program that provides queuing services to applications. It provides an application-programming interface so that programs can put messages on, and get messages from, queues. A queue manager provides additional functions so that administrators can create new queues, alter the properties of existing queues, and control the operation of the queue manager. The actual creation and deletion of queue managers are controlled by access control functions in the OS and therefore are not within the scope of the evaluation. Management within queues is within the scope of the evaluation.

2.1.1 Object Authority Manager (OAM)

Authorization for using MQI calls, commands, and access to objects, is provided by the Object Authority Manager (OAM), which is enabled by default. Access to WebSphere MQ objects is controlled by the OAM based upon User and Group IDs controlled by the operating system. A command line interface to enable administrators to grant or revoke authorizations is used.

The OAM needs to be able to identify who is requesting access to a particular object. WebSphere MQ uses the term *principal* to refer to the User identifier associated with a user. The principal is established when the application first connects to the queue manager; it is determined by the queue manager from the user ID associated with the connecting application.

WebSphere MQ propagates the user ID received from the system in the message header of each message as identification of the user. This user ID is then checked against those on the Access Control List (ACL) of the object.

2.1.2 QM Kernel

In WebSphere MQ, event monitoring is performed by the QM Kernel. An instrumentation event is a logical combination of conditions that is detected by a queue manager. Such an event causes the queue manager or channel instance to put a special message, called an *event message*, on an event queue. An event queue is like any other MQ queue in the way access is controlled but it stores only event messages.

WebSphere MQ instrumentation events provide information about errors, warnings, and other significant occurrences in a queue manager, and in particular authorisation failures.

2.1.3 Application Interface (AI)

The AI component provides an external interface to the TOE. It is responsible for accepting calls from an application, and performing simple syntax checking on the parameters. The native interface for WMQ is the MQI API.

Applications can access MQ objects by issuing MQI calls. The applications can also use PCF commands to access these objects. PCF encapsulates the MQI API

Base Java applications use the Java API provided by the WebSphere MQ classes for Java. JMS applications use the JMS API and the additional API provided by the WebSphere MQ classes for JMS.

2.1.4 Data Abstraction and Persistence (DAP)

The DAP component of the Queue Manager holds the attributes of objects such as process definitions and queues, and the messages on the queues. The DAP component is responsible for the local queue attributes. None of these are security attributes as defined within this ST.

2.2 Command line Interface

The command line interface is used to enable administrators to provide management of the queue manager. Only administrators (i.e. members of the *mqm* group on UNIX and

Windows and local system administrators on Windows) are authorised to issue control commands. The *mqm* group is automatically created by the install process.

Administrators can use control commands to administer WMQ. One of these control commands is *setmqaut*, which is used to grant authorities to users to enable them to access WMQ resources.

Administrators can use the control command *runmqsc* to enable the use of MQSC. These are used to manage the message queues and other WMQ objects.

2.3 Common Services

The Common Services layer provides an Operating System (OS) independent, external interface to those services that other components wish to use that contains platform specific code. It performs no security checks and is simply an WMQ developer's interface.

2.4 Message Channel Agent (MCA)

A message channel is a one-way link. It connects two queue managers via message channel agents (MCAs). Its purpose is to transfer messages from one queue manager to another. Message channels are not required by the client/server environment since the clients use MQI Channels to transfer messages to and from the servers..

The Message Channel Agent is a program that transmits prepared messages from a transmission queue to a communication link or from a communication link to a destination queue. The transmitted messages can be secured through the use of TLS/SSL over TCP/IP. All external data from and to the MCA is protected by the TLS/SSL function provided by the GSKit software.

2.5 TOE Environment

WMQ relies upon those responsible for the administration of the TOE and the TOE environment to be competent and trustworthy individuals, capable of managing the TOE and its environment and the security of the information it contains.

WMQ relies upon the OS to provide the security environment to protect both the client and server. The OS provides WMQ identification and authorization for TOE users and groups. It also provides reliable time and date information for certificate and CRL verification. The OS also ensures that all accesses to certificate and key stores are limited to authorized users.

CRLs can be retrieved using an LDAP client and server provided by the TOE environment. The connection between the LDAP client and the LDAP server must be an internal communication link within a trusted network. Additionally, the JMSAdmin tool requires an LDAP server or file system to store the JNDI objects.

The TOE environment includes an application to read audit records produced by the TOE.

The GSKit key database file is protected by a password. In order to allow unattended access to the key database file GSKit provides a stash file to store the password. This stash file must be protected by an ACL or permission bits while it resides on the system and by encrypting the stash file when it is backed up.

The JMS/Java clients don’t access passwords for the stash file directly – this is handled by the JSSE and how the user gets the password to the JSSE is beyond the scope of this TOE.

3 TOE Security Environment

3.1 Introduction

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

The statement of TOE security environment therefore identifies the assumptions made on the operational environment and the intended method for the product, defines the threats that the product is designed to counter and the organisational security policies which the product is designed to comply.

3.2 Threats

The assumed security threats are listed below:

3.2.1 Threats countered by the TOE

[T.ACCESS_RES] An authorised user of the TOE gains access to an object without the correct authority to access that object

[T.ACCOUNT] Unauthorised attempts to access objects for which the user has no authority go undetected.

[T.CHANNEL] Data transferred between platforms is disclosed to, or modified by unauthenticated users or processes, either directly or indirectly.

3.2.2 Threats countered by the TOE Environment

[T.ACCESS_TOE] An unidentified user gains access to the TOE and its objects.

[T.ROLE] A non-privileged user gains administrative privileges.

[T.OS] The operating system on which the TOE is installed becomes compromised.

3.3 Organisational Security Policies (OSPs)

The TOE complies with the following OSP:

[P.ACCESS] The right to access a specific object is determined on the basis of:

- The identity of the subject attempting to access the object; or

- Membership of a group that has access rights to the object.

3.4 Assumptions

This section provides the minimum physical and procedural measures required to maintain security of the WebSphere MQ product.

3.4.1 Physical aspects

[A.OS] It is assumed that the operating system has been configured in accordance with the manufacturer's installation guides and where applicable, in its evaluated configuration. It is securely configured such that the operating system protects the TOE from any unauthorised users or processes.

[A.PROTECT] It is assumed that all software and hardware, including peripheral devices, have been approved for the transmittal of protected data. Such items are to be physically protected against threats to the confidentiality and integrity of the data.

3.4.2 Personnel Aspects

[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, wilfully negligent or hostile.

4 Security Objectives

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 an object.

[O.ACCOUNT] The TOE must provide a means of recording any unsuccessful access attempts to the objects.

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

[O. PROTECT] The TOE must ensure that data transferred between platforms is secured from disclosure to or tampering by unauthenticated users .[1]

4.2 Security Objectives for the TOE Environment

[O.AUDIT] The operating system must ensure an audit tool is available to only administrators to review the audit trail.

[O.IDENTIFY] The operating system must ensure that all users are identified.

[O.ROLE] The operating system must be able to associate users with roles and maintain an *administrator* role.

[O.TIME] The operating system must ensure that the clock is accurate and reliable.

[O.TOE_PROTECTION] The operating system will provide protection to the TOE and its assets from external interference, tampering, and disclosure.

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

[O.CONFIG] Those responsible for the TOE must ensure that each user on the supporting operating system has a unique user ID and that the operating system is configured to ensure that only approved groups of users may access the system.

[O.OS] Those responsible for the TOE must ensure that the supporting operating system is installed and configured in accordance with the

[1] The TOE does not log the detection of disclosure or tampering of the transferred data

manufacturer’s instructions, the evaluated configuration where applicable and is secure.

[O.RECOVER] Those responsible for the TOE’s 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 Security Requirements

This section specifies the Security Functional Requirements (SFRs) for the TOE and organises the SFRs by class.

Within the text of each SFR, the selection and assignment operations (as defined within [CC]) are *italicised*.

Note: FAU_GEN_MQ.1 and FMT_MSA_MQ.3 are explicitly stated IT security requirements, and have not been specified using CC Part 2 functional components.

The International Interpretations that have been applied for the Security Requirements are 058, 064, 065, and 103.

5.1 TOE Security Functional Requirements

The following table summarises the SFRs:

CLASS	FAMILY	COMPONENT	ELEMENT
FAU	FAU_GEN	FAU_GEN_MQ.1	FAU_GEN_MQ.1.1
			FAU_GEN_MQ.1.2
		FAU_GEN.2	FAU_GEN.2.1
	FAU_STG	FAU_STG.1	FAU_STG.1.1
			FAU_STG.1.2
FCS	FCS_COP	FCS_COP.1(a-g)	FCS_COP.1.1(a-g)
FDP	FDP_ACC	FDP_ACC.1	FDP_ACC.1.1
	FDP_ACF	FDP_ACF.1	FDP_ACF.1.1
			FDP_ACF.1.2
			FDP_ACF.1.3
			FDP_ACF.1.4
FMT	FMT_MSA	FMT_MSA.1	FMT_MSA.1.1
		FMT_MSA_MQ.3	FMT_MSA_MQ.3.1

	FMT_MTD	FMT_MTD.1	FMT_MTD.1.1
	FMT_SMF	FMT_SMF.1	FMT_SMF.1.1
FPT	FPT_ITT	FPT_ITT.1	FPT_ITT.1.1

5.1.1 Security Audit (FAU)

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

- Authorization failures.

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

- Date and time of the event, type of event and subject identity; and
- The type of the application causing the event.

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

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

FAU_STG.1.2 The TSF shall be able to *prevent* unauthorised modifications to the stored audit records in the audit trail..

5.1.2 Cryptography (FCS)[2]

FCS_COP.1.1(a) The TSF shall perform *symmetric encryption and symmetric decryption* in accordance with a specified cryptographic algorithm

a) DES (CBC mode),

b) TDEA with three independent keys (CBC mode) ,

c) *AES (CBC mode)*

and cryptographic key sizes

a) 56 Bits (DES),

[2] All Cryptography SFRs are taken from the GSKit Security Target.

b) 168 Bits (TDEA),

c) 128, 256 Bits (AES)

that meet the following:

a) conformant to FIPS 46-3 [FIPS46-3] (DES), conformant to FIPS 81 [FIPS81] (CBC mode),

b) conformant to FIPS 46-3 [FIPS46-3] (TDEA), conformant to FIPS 81 [FIPS81] (CBC mode),

c) conformant to FIPS 197 [FIPS197] (AES, CBC mode),

and FIPS 140-2 [FIPS140-2] approved.

FCS_COP.1.1(b) The TSF shall perform *digest generation and verification* in accordance with a specified cryptographic algorithm

a) SHA-1

b) *M*D5

and cryptographic key sizes none that meet the following:

a) conformant to the Secure Hash Standard (SHS) as defined in FIPS 180-2 [FIPS180-2] and FIPS 140-2 [FIPS140-2] approved (SHA-1),

b) conformant to RFC 1321 [RFC1321] (MD5).

FCS_COP.1.1(c) The TSF shall perform *data authentication* in accordance with a specified cryptographic algorithm *HMAC SHA-1* and cryptographic key sizes *160 Bits (HMAC SHA-1)* that meet the following:

conformant to FIPS 198 [FIPS198] (HMAC) and FIPS 180-2 [FIPS180-2] (SHA-1), and FIPS 140-2 [FIPS140-2] approved.

FCS_COP.1.1(d) The TSF shall perform *encryption and decryption of session key related data* in accordance with a specified cryptographic algorithm *RSA* and cryptographic key sizes *1024 or 2048 Bits* that meet the following:

conformant to RFC 2437 [RFC2437] and RFC 2313 [RFC2313] (RSA) and encryption/decryption of session key related data as defined in The SSL Protocol, Version 3 [SSLv3] and RFC 2246 [TLSv1].

FCS_COP.1.1(e) The TSF shall perform *generation of random numbers* in accordance with a specified cryptographic algorithm *Universal Software Base True Random Number Generator* algorithm and cryptographic key sizes *none* that meet the following:

the requirements in FIPS 186-2 [FIPS186-2], Appendix 3.2 as required in FIPS 140-2 annex C [FIPS140-2] and FIPS 140-2 level 1 [FIPS140-2] approved.

FCS_COP.1.1(f) The TSF shall perform *digital signature generation and verification* in accordance with a specified cryptographic algorithm *RSA with SHA-1 message digest* and cryptographic key sizes *1024 or 2048 Bits (RSA)* that meet the following:

conformant to RFC 2437 [RFC2437] and RFC 2313 [RFC2313] (RSA) and FIPS 180-2 [FIPS180-2] (SHA-1) and FIPS 140-2 [FIPS140-2] approved.

FCS_COP.1.1(g) The TSF shall perform *digital signature verification* in accordance with a specified cryptographic algorithm

a) RSA with MD2 message digest,

b) *RSA* with MD5 message digest

and cryptographic key sizes

a) 1024 or 2048 Bits (RSA),

b) *1024* or 2048 Bits (RSA)

that meet the following:

a) conformant to RFC 2437 [RFC2437] and RFC 2313 [RFC2313] (RSA) and RFC 1319 [RFC1319] (MD2),

b) *conformant to RFC 2437 [RFC2437] and RFC 2313 [RFC2313] (RSA)* and RFC 1321 [RFC1321] (MD5),

and RSA is FIPS 140-2 [FIPS140-2] approved.

5.1.3 Access Control (FDP)

FDP_ACC.1.1 The TSF shall enforce the *access control policy* on *processes acting on behalf of users, objects and all operations among processes acting on behalf of users and the following objects:*

- *Queue Managers;*
- *Queues;*
- *Process definitions;*
- *Namelists;*
- *Authorization Information;*
- *Channels.*
- *CLNTCONN Channel Objects,*
- *Service Objects*
- *Listener Objects**.***

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

Subject	*Security Attributes*
Process acting on behalf of a user	*User/Group IDs*
Object	*Security Attributes*
Queues	*ACL*
Process definitions	*ACL*
Namelists	*ACL*
Authorization Information	*ACL*
Channel	*ACL*
Clntconn	*ACL*
Service	*ACL*
Listener object	*ACL*

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: *if the subject's user or group ID is present within the object's ACL for the requested access then access is permitted*

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*.

5.1.4 Security Management (FMT)

FMT_MSA.1.1 The TSF shall enforce the *access control policy* to restrict the ability to *modify* the security attributes *ACL* to *the administrator*.

FMT_MSA_MQ.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_MTD.1.1 The TSF shall restrict the ability to *delete* the *event messages* to the *administrator*.

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

5.1.5 Protection of the TSF (FPT)

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

5.2 Strength Of Function (SOF)

There is no strength of function claim because the TOE does not identify any security functional requirements for which an explicit Strength of Function (SOF) is appropriate and does not identify any functions that are of a permutational or probabilistic nature. Therefore, a minimum SOF claim is not included for the TOE.

5.3 TOE Security Assurance Requirements

The target evaluation assurance level for this product is EAL4 augmented with ALC_FLR.2.

5.4 Security Requirements for the IT Environment

This section specifies the Security Requirements for the IT environment and organises the requirements by class.

Within the text of each SFR, the selection and assignment operations (as defined within [CC]) are *italicised*.

CLASS	FAMILY	COMPONENT	ELEMENT
FAU	FAU_SAR	FAU_SAR.2	FAU_SAR.2.1
FIA	FIA_ATD	FIA_ATD.1	FIA_ATD.1.1
	FIA_UAU	FIA_UAU.2	FIA_UAU.2.1
	FIA_UID	FIA_UID.2	FIA_UID.2.1
FMT	FMT_SMR	FMT_SMR.1	FMT_SMR.1.1
			FMT_SMR.1.2
FPT	FPT_RVM	FPT_RVM.1	FPT_RVM.1.1

	FPT_SEP	FPT_SEP.1	FPT_SEP.1.1
			FPT_SEP.1.2
	FPT_STM	FPT_STM.1	FPT_STM.1.1

5.4.1 Security Audit (FAU)

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

5.4.2 Identification and Authentication (FIA)

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users: *User and Group IDs.*

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

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

5.4.3 Security Management (FMT)

FMT_SMR.1.1 The TSF shall maintain the role *administrator.*

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

5.4.4 Protection of the TSF (FPT)

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

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

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

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

© Copyright IBM 2006

6 TOE Summary Specification

6.1 IT Security Functions (SF)

6.1.1 Access Control

AC.1 The TSF shall ensure that access to an object is only given to a process acting on behalf of a user, if the associated user and group IDs associated with the user, has been granted permission to access to that object. The user and group IDs are gained from the operating system and cached in memory for any subsequent access requests. Each process contains the user ID within the message descriptor part of the process, which is used to confirm the group permissions. Permission is confirmed by checking that either the UID or GID is contained within the object's Access Control List (ACL).

AC.2 The TSF shall ensure that only the administrators are able to modify the ACL or delete event messages. Administrators are users that belong to the *MQM* or administrator groups within the Operating System environment. Identification is performed in the same way as normal users.

AC.3 On creation of an object, the TSF shall set default values for that object such that only the ID associated with the process creating the object and the administrator are able to access that object. This is done by adding the creators and administrators UID and GIDs to the ACL of that object. Once an object has been created, then the administrator can update the ACL to grant or revoke access via the command line interface.

6.1.2 Audit

Audit.1 Provided that the event queue is not full, then the TOE shall generate an event message for Authorisation failures. The Queue Manager will put an event message onto the event queue, which behaves in the same manner as all other queues and like other queues has an ACL list, with access only given to the administrator (i.e. members of the MQM group). If the event queue becomes full, then no auditing will take place.

Audit.2 For each event message, the following information is recorded:

- Date and Time;
- Type of event;
- Type of application that caused the event; and
- User identity.

The date and time information is retrieved from the Operating system each time an event message is created. The User ID is gained from the process message descriptor. The Type of event in this case is authorisation failure. Viewing of the audit records is performed via a third party application.

Audit.3 The event queue will be protected to prevent unauthorised modification and deletion of audit records. This is done in the same way as all other queues (see AC.1) with only the administrator (member of MQM group) being able to access the queue. Queue administration is performed using the MQSC commands, which are initialised by entering the *runmqsc* control command at the administrative interface.

6.1.3 TOE Protection

TP.1 The TSF shall ensure that WMQ channels from WMQ clients to a WMQ server, or between two WMQ servers, are established using a TLS/SSL CipherSpec (or CipherSuite for Java/JMS) listed in section 2.0. The TLS/SSL support provided by WMQ provides authentication, message integrity checking, and data encryption for transmitted data.

6.2 Assurance Measures

Assurance measures will be adopted to address each of the EAL4 assurance requirements, as summarised in table B.1 within [CC]. The following table provides a summary:

Assurance Component	Description of how Requirement will be met
ACM_AUT.1	A description of the automated measures to control access to the TOE implementation representation is included in the CM plan.
ACM_CAP.4	A description of the configuration management used by the developers is provided to the evaluators together with a configuration list, which identifies the items that comprise the TOE, an acceptance plan, and generation support for the TOE. This document uniquely references the TOE stated within Section 1 of this ETR. Confirmation that the TOE is labelled with the correct reference was provided during testing.
ACM_SCP.2	A configuration list is included that identifies items to be tracked by the CM system. This list includes all CC evidence.
ADO_DEL.2	The developers provided the evaluators with the delivery procedures used to ensure that security is maintained when distributing versions of the TOE to the user's site. This is contained within the Configuration Management documentation.
ADO_IGS.1	Procedures for the secure installation, generation and start-up are provided at the following URL: http://www-3.ibm.com/software/integration/mqfamily/library/manualsa/manuals/platspecific.html
ADV_FSP.2	An informal description of the TSF and its external interfaces, describing effects, exceptions and interfaces is provided to the evaluators.

ADV_HLD.2	A high-level design is provided to the evaluators, which informally describes the components of the TSF. The security of each of these components is described. All hardware, software and firmware required by the TOE is identified. A presentation of the functions provided by the supporting protection mechanisms implemented in these, is also included. It also identifies the interfaces between the components and which of these are externally visible.
ADV_IMP.1	Sample source code files that are representative of the subcomponents identified in the low-level design is provided as an implementation representation.
ADV_LLD.1	A low-level design is provided to the evaluators, which informally describes the TSF in terms of modules. Module descriptions include a description of not only module purpose, but the manner in which the module achieves its purpose.
ADV_RCR.1	This correspondence information is contained within the Functional Specification and high-level design. This provides a correspondence analysis between the TOE summary specification, the functional specification and the high level design, the high level design to the low level design, and from the low level design to the implementation.
ADV_SPM.1	An informal security policy describes both implicit and explicit security policies that are included in the ST in terms of controlled entities, potential modifiers, and rules potential modifiers must follow.
AGD_ADM.1	The WebSphere MQ operational documentation that describes to the administrator how to operate the TOE in a secure manner is provided at the following URL: http://publibfp.boulder.ibm.com/epubs/html/amqzag04/amqzag04tfrm.htm This describes the administrative security functions and interfaces available to the administrator. All details of any warnings about functions and privileges and assumptions about user behaviour are included. Secure parameters under the control of the administrator are provided, indicating secure values where applicable.
AGD_USR.1	User guidance is provided that details of any warnings about functions and privileges and assumptions about user behaviour. .
ALC_DVS.1	A description of the security measures used in the development environment that are necessary to protect the confidentiality and integrity of the TOE design and implementation is provided.
ALC_FLR.2	A description of the flaw remediation procedures is provided, including identifying and tracking reported flaws, ensuring that they are addressed and corrections and corrective measures are made available as applicable.
ALC_LCD.1	A description of the procedures, tools and techniques used to develop and maintain the TOE is provided, including describing individuals' responsibilities for life cycle activity procedures.

ALC_TAT.1	A description of tools used in life cycle activities is provided. Descriptions of programming languages are not provided as they are common languages.
ATE_COV.2	Coverage of the TSF by the developers functional testing to the functional specification is provided to the evaluators as part of the testing documentation.
ATE_DPT.1	Coverage of the TSF by the developers functional testing to the high level design is provided to the evaluators as part of the testing documentation
ATE_FUN.1	Testing documentation is provided to the evaluators, which describes the functional tests performed by the developers. This document includes test plans, test procedures, expected and actual test results, It also identifies the security functions to be tested.
ATE_IND.2	Resources were made available to the evaluators such that they are able to perform additional, independent testing.
AVA_MSU.2	A misuse analysis demonstrating the guidance documentation is complete has been provided.
AVA_SOF.1	There are no functions within the TOE that have a strength and therefore no Strength of Functions analysis will be produced.
AVA_VLA.2	A description and analysis of any potential vulnerability identified within the TOE was performed and documented with an explanation of why the vulnerabilities cannot be exploited.

7 Rationale

This chapter presents the evidence used in the ST evaluation and supports the claims that the ST is a complete and cohesive set of requirements.

7.1 Correlation of Threats, Policies, Assumptions and Objectives

The following table provides a correspondence of the threats, policies, assumptions and objectives:

Objectives:	O.ACCESS	O.ACCOUNT	O.MANAGE	O.PROTECT	O.AUDIT	O.IDENTIFY	O.ROLE	O.TOE PROTECTION	O.TIME	O.ADMIN	O.CONFIG	O.OS	O.RECOVER
T.ACCESS_RES	x		x	x									
T.ACCOUNT		x	x		x				x	x	x		
T.CHANNEL				x								x	
T.ACCESS_TOE						x					x		
T.ROLE						x	x				x		
T.OS				x						x	x	x	
P.ACCESS	x		x							x	x	x	
A.OS								x	x	x	x	x	x
A.PROTECT										x		x	
A.ADMIN										x			

7.2 Security Objectives Rationale

This section demonstrates that the security objectives stated in section 4 of this ST are traceable to all of the aspects identified in the TOE security environment and are suitable to cover them.

7.2.1 Threats

This section provides evidence demonstrating coverage of the threats by both the IT and non-IT security objectives.

[T.ACCESS_RES]

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

The objective O.ACCESS counters this directly by ensuring that only those users with the correct authority can access an object. This is supported by O.MANAGE, which ensures that privileged actions are performed effectively. The O.PROTECT objective supports O.ACCESS in countering the threat by ensuring that data transferred between platforms is secured.

[T.ACCOUNT]

Unauthorised attempts to access objects for which the user has no authority go undetected.

Recording unsuccessful attempts to access objects is performed by O.ACCOUNT. O.MANAGE supports this objective by ensuring that all users are identified before they access the TOE and the identify can be used on making access decisions..

O.ADMIN and O.CONFIG further support these objectives by ensuring that the administrator manages the event messaging security functions effectively. [O.TIME] ensures that the time information recorded for each event is accurate. O.AUDIT ensures the administered has a tool to review the audit trail and the tool is protected.

[T.CHANNEL]

Data transferred between platforms is disclosed to, or modified by unauthenticated users or processes, either directly or indirectly.

The TOE ensures that data transferred between platforms i.e. along network cabling, is suitably protected against physical or other (e.g. Sniffing) attacks that may result in the disclosure, modification or delay of information transmitted between platforms. Objective O.PROTECT ensures that this is achieved. O.OS ensures that the protocols used in the transmission of data have been correctly configured within the operating systems.

[T.ACCESS_TOE]

An unidentified user gains access to the TOE and it's objects.

O.IDENTIFY is the primary objective that counters this threat, by ensuring that all users are identified. The environmental objective O.CONFIG supports O.IDENTIFY in countering the threat by ensuring that all users have a valid and unique identity.

[T.ROLE]

A non-privileged user gains administrative privileges.

Only those users with the correct authority can invoke administrative privileges. O.ROLE ensures that the users are associated with roles so enable the efficient management of administrative users, and maintains an administrative role in order that the TOE can be managed. This relies upon O.IDENTIFY and O.CONFIG, which ensure that each user is identified.

[T.OS]

The operating system on which the TOE is installed becomes compromised.

It is essential that the administrator manage the operating system in a secure manner so that vulnerabilities do not exist, which may lead to compromise of the TOE. The objectives O.OS, O.CONFIG, and O.PROTECT all ensure that the operating system is managed in a secure manner. O.ADMIN further supports this threat by ensuring that the administrator is a competent individual that will apply the latest patch information and therefore ensuring that any vulnerabilities to the TOE that become known are be countered by application of the relevant patch.

7.2.2 Security Policy

This section provides evidence demonstrating coverage of the organisational security policy by both the IT and non-IT security objectives.

[P.ACCESS]

The right to access a specific object is determined on the basis of:

- *The identity of the subject attempting to access the object; or*
- *Membership of a group that has access rights to the object.*

This policy is implemented through the objective O.ACCESS, which 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.

The following environmental objectives further support the policy:

- O.ADMIN, O.CONFIG and O.OS all ensure that the operating system is configured in a secure manner so that no vulnerability may exist that enables an unauthorised user to gain an authorised identity.

7.2.3 Assumptions

This section provides evidence demonstrating coverage of the assumptions by both the IT and non-IT security objectives.

[A.OS]

It is assumed that the operating system has been configured in accordance with the manufacturers instructions and where applicable, the evaluated configuration. It is securely configured such that the operating system protects the TOE from any unauthorised users or processes.

O.OS is the primary environmental objective that satisfies the assumption. This ensures that the administrator installs and configures the supporting operating systems in accordance with:

- The manufacturers instructions; and
- Any evaluated configurations were applicable.

O.ADMIN and O.CONFIG supports this by ensuring that the Administrator is a competent and trustworthy person and that the users have been set up appropriately. O.RECOVER ensures that following a system failure, the TOE is not operating in an insecure state whereby an unauthorised user can gain access to objects they are not authorised to access. O.TIME ensures that the time provided by the OS is accurate and reliable. The O.TOE_PROTECTION objective ensures other operating system protects the TOE from any unauthorised users or processes.

[A.PROTECT]

It is assumed that all software and hardware, including peripheral devices, have been approved for the transmittal of data. Such items are to be physically protected against threats to the confidentiality and integrity of the data.

The objective O.OS ensures that the underlying OS is installed and configured properly. O.ADMIN ensures the administrators are capable of selecting and managing the OS such that is the evaluated OS and can handle the transmittal of data, .

[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, wilfully negligent or hostile.

O.ADMIN is the primary objective that meets this assumption, which ensures that the administrator is a competent and trustworthy person whom is capable of managing the TOE in a secure manner.

7.3 Security Requirements Rationale

7.3.1 Security Functional Requirements Rationale

This section demonstrates that the functional components selected for the TOE provide complete coverage of the defined security objectives. The mapping of components to security objectives is illustrated in the table below.

Security Objective	Functional Component
O.ACCESS	Subset Access Control (FDP_ACC.1) Security Attribute Based Access Control (FDP_ACF.1) Management of Security Attributes (FMT_MSA.1) Static Attribute Initialisation (FMT_MSA_MQ.3)
O.ACCOUNT	Audit Data Generation (FAU_GEN_MQ.1) User Identity Association (FAU_GEN.2) Protected Audit Trail Storage (FAU_STG.1) Management of TSF Data (FMT_MTD.1)
O.MANAGE	Management of Security Attributes (FMT_MSA.1) Static Attribute Initialisation (FMT_MSA_MQ.3) Management of TSF Data (FMT_MTD.1) Specification of Management Functions (FMT_SMF.1)
O.PROTECT	Cryptographic Operations (FCS_COP.1) Basic Internal TSF Data Transfer Protection (FPT_ITT.1)

[O.ACCESS]

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

The access control mechanism must have a defined scope of control [FDP_ACC.1] with defined rules [FDP_ACF.1]. Authorised 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_MQ.3]

[O.ACCOUNT]

The TOE must provide a means of recording any unsuccessful access attempts to the objects.

Security relevant actions must be defined, auditable [FAU_GEN_MQ.1] and capable of being associated with individual users [FAU_GEN.2]. The event queue must be protected so that only authorised users can access it [FAU_STG.1]. An authorised administrator must be able to manage the event queue [FMT_MTD.1].

[O.MANAGE]

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

The TSF must enable an authorised administrator to manage the TOE by the access control policy objects [FMT_MSA.1] with default values [FMT_MSA_MQ.3]. The administrator must be able to manage the event queue [FMT_MTD.1] and the access control list [FMT_MSA.1]. [FMT_SMF.1] specifies the management functions provided by the TOE.

[O. PROTECT]

The TOE must ensure that data transferred between platforms is secured from disclosure to or tampering by unauthenticated users.

The TOE will protect all TSF data while it is in transit among distributed portions of the TOE [FPT_ITT.1] using evaluated cryptography (FCS_COP.1). Two requirements from the environment support this objective by ensuring the operating system cannot be bypassed (FPT_RVM.1) and untrusted processes cannot interfere with the TOE (FPT_SEP.1).

7.3.2 Security Environment Requirements Rationale

This section demonstrates that the functional components provided by the environment for the TOE, provide complete coverage of the defined security objectives. The mapping of requirements to security objectives is illustrated in the table below.

Security Objective	Requirement for Environment
O.AUDIT	Audit Review (FAU_SAR.2)
O.IDENTIFY	User Attribute Definition (FIA_ATD.1) User Authentication (FIA_UAU.2) User Identification (FIA_UID.2)
O.ROLE	User Identification (FIA_UID.2) Security Management (FMT_SMR.1)
O.TOE_PROTECTION	Non-bypassability of the TSP (FPT_RVM.1)

Security Objective	Requirement for Environment
	TSF domain separation (FPT_SEP.1)
O.TIME	Protection of the TSF (FPT_STM.1)

[O.AUDIT]

The operating system must ensure an audit tool is available to only administrators to review the audit trail.

In order for the audit records from the TOE to be reviewed, the operating system must provide a tool to review the audit trail and must restrict use of that tool to the administrator (FAU_SAR.2).

[O.IDENTIFY]

The Operating System must ensure that all users are identified.

The TSF must maintain a list of User and Group IDs for each user (FIA_ATD.1) , identify users, and authenticate users before allowing any other actions (FIA_UID.2, FIA_UAU.2)).

[O.ROLE]

The operating system must be able to associate users with roles and maintain an administrator role.

In order to associate a user with a role (FMT_SMR.1), the user needs to be identified FIA_UID.2) and maintain administrative roles (FMT_SMR.1).

[O.TOE_PROTECTION]

The operating system will provide protection to the TOE and its assets from external interference, tampering, and disclosure.

The operating system protects its own data (local and in transit) and executable code as well as that of its hosted applications (FPT_SEP.1, FPT_RVM.1).

[O.TIME]

The operating system must ensure that the clock is accurate and reliable.

In order that the TOE is able to provide accurate time stamps, in this case for audit records, the operating system that the TOE is relying for the time information must ensure that this is reliable and accurate (FPT_STM.1).

7.3.3 Explicitly Stated Security Requirements Rationale

As stated within Section 5 of this ST, FAU_GEN_MQ.1 and FMT_MSA_MQ.3 have been explicitly stated and were not specified using CC Part 2 functional components. The reasons for this are as follows:

FAU_GEN_MQ.1

The TOE does not generate an audit record for the start-up and shutdown of the auditing functions and the success/failure of the events audited is not recorded.

These do not lead to any vulnerability within the system because only the administrator is able to start-up and shutdown the auditing functions and is trusted to operate the system securely. If an unauthorised user were able to start-up and shutdown the auditing function, then that user would have administrative rights and would therefore be capable of performing any action on the TOE.

Auditing of every successful attempt to access an object would create an impractically large audit file with no benefit to the administrator. Therefore only *unsuccessful* attempts to access an object generate an audit record.

FMT_MSA_MQ.3

WMQ does not provide functionality to define alternate initial values that override the default values when an object has been created. This does not reduce security as the default values used are the most restricted that would enable normal operation of the TOE.

7.3.4 Security Assurance Requirements Rationale

This ST contains assurance requirements from the CC EAL4 assurance package.

The EAL chosen is based on the impact that the statements of the security environment and objectives within this ST have on the assurance level. The administrator shall be capable of managing the TOE such that the security is maintained (O.ADMIN) particularly within the operating system that the TOE relies (O.OS). This EAL level also provides a resistance to penetration attackers with a low attack potential.

Given the amount of assurance required to meet the TOE environment and the intent of EAL4, this assurance level was considered most applicable for the TOE described within this ST.

7.4 SFR Dependencies

The below table identifies all of the dependencies of the SFRs included in the ST. Only those SFRs that have a dependency, or are depended upon are shown in the table.

	FAU_GEN.1*	FCS_CKM.1	FCS_CKM.4	FDP_ACC.1	FDP_ACF.1	FIA_UID.1	FMT_MSA.1	FMT_MSA.2	FMT_MSA.3*	FMT_SMF.1	FMT_SMR.1	FPT_STM.1
FAU_GEN_MQ.1												x
FAU_GEN.2	x					x						
FAU_STG.1	x											
FCS_COP.1		x	x					x				
FDP_ACC.1					x							
FDP_ACF.1				x					x			
FMT_MSA.1				o						x	x	
FMT_MSA_MQ.3							x				x	
FMT_MTD.1										x	x	
FPT_ITT.1 (no dependencies)												

The key to the symbols used, are:

x required dependency

o optional dependency

As shown in [CC], all dependencies are satisfied by the TOE, with the exception of the dependencies on FIA_UID.1, FMT_SMR.1 and FPT_STM.1. These dependencies are met by the IT environment of the TOE.

FIA_UID.1 is countered by the IT environment of the TOE because the Operating system provides the TOE with the user IDs (FIA_UID.2). Additionally, this is met by FIA_UID.2 as it is hierarchical to FIA_UID.1.

The dependencies for FCS_COP.1 are not included because they have been addressed in the GSKit evaluation. The WMQ TOE relies on the cryptographic operations provided by the GSKit and depends on the GSKit to have addressed key management appropriately.

The TOE does not 'maintain' an administrator role (FMT_SMR.1) or internal clock (FPT_STM.1), but relies upon the operating system to maintain the role and clock. Identification of the administrator is based on membership of the mqm or administrators group defined within the operating system. [A.OS] assumes that *the operating system has been configured in accordance with the manufacturer's installation guides and where* *applicable, in its evaluated configuration. It is securely configured such that the operating system protects the TOE from any unauthorised users or processes.*

* The reliance on the requirements FAU_GEN.1 and FMT_MSA.3 are countered by the explicitly stated requirements FAU_GEN_MQ.1 and FMT_MSA_MQ.3 respectively, which records auditing details and sets default values on creation of objects. In turn, the explicitly stated requirements assume the dependencies on FAU_GEN.1 and FMT_MSA.3 as shown in the above table.

7.5 TOE Summary Specification Rationale

This section demonstrates that the TOE security functions and assurance measures are suitable to meet the TOE security requirements.

7.5.1 TSF correspondence to SFRs

This section demonstrates that the combination of the specified TSFs work together so that the SFRs are satisfied. The table below shows the TOE security functions, which together satisfy each SFR element. For a detailed description of how the SFR relates to the security function, refer to appropriate section in Section 6.

TOE SFR	TSFs
FAU_GEN_MQ.1.1	Audit.1
FAU_GEN_MQ.1.2	Audit.2
FAU_GEN.2.1	Audit.2
FAU_STG.1.1	Audit.3
FAU_STG.1.2	Audit.3
FDP_ACC.1.1	AC.1
FDP_ACF.1.1	AC.1
FDP_ACF.1.2	AC.1
FDP_ACF.1.3	AC.1
FDP_ACF.1.4	AC.1
FMT_MSA.1.1	AC.2
FMT_MSA_MQ.3.1	AC.3
FMT_MTD.1.1	AC.2
FMT_SMF.1.1	AC.1, AC.2 and AC.3

TOE SFR	TSFs
FTP_ITT.1.1	TP.1

© Copyright IBM 2006