IBM Tivoli Netcool/OMNIbus 7.3.1 Security Target

Version:	**2.10**
Status:	**Released**
Last Update:	**2012-11-09**
Classification:	**Public**

Trademarks

IBM and the IBM logo are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

- IBM WebSphere Application Server
- IBM Tivoli Netcool/OMNIbus

The following terms are trademarks of Microsoft Corporation in the United States, other countries, or both:

- Microsoft Windows

The following terms are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both:

- Sun Solaris
- Java

Other company, product, and service names may be trademarks or service marks of others.

Legal Notice

This document is provided AS IS with no express or implied warranties. Use the information in this document at your own risk.

This document may be reproduced or distributed in any form without prior permission provided the copyright notice is retained on all copies. Modified versions of this document may be freely distributed provided that they are clearly identified as such, and this copyright is included intact.

Revision History

Version	Date	Author(s)	Changes to Previous Revision	Application Notes
0.1	2008-10-09	David Ochel	First version	Draft for 1st review of chapter 1 (TOE Description)
0.2	2008-10-22	David Ochel	Incorporating review comments from IBM, completing TOE description and SFRs.	
0.3	2008-10-29	David Ochel	Further clarifications from IBM, response to initial evaluator comments, TSS completed.	
0.4	2008-11-11	David Ochel	Integrating feedback from IBM.	Augmented TOE Overview; figure 1: AEN now in JRE; clarified ObjectServer PAM module and transaction journal in 1.4.1.2; removed restriction re. IPv6 and changed SSL requirement in 1.4.1.6; changed FAU_GEN.1 to FAU_GEN_EXT.1, CC Part 2 conformant to extended, and updated 7.1.6 accordingly;

Version	Date	Author(s)	Changes to Previous Revision	Application Notes
				clarified Web GUI caching in FIA_USB.1.3 and 7.1.2; clarified crypto CLIs in 7.1.4; added definition of root user; added explanation of SFR operation abbreviations; fixed typographical errors.
0.5	2008-11-17	David Ochel	Integrating feedback from IBM and evaluator.	Updated definition of Administrators and changed name of platform in Figure 1 to "User" host/OS; 1.4.1.1: clarified that gateways do not generate events, event data/alert wording in probe section, and definition of proxy servers; clarified use of embedded WAS and non-dependency on hardware in 1.4.1.5; updated A.SEL_PRO and OE.Runtime to explicitly include audit records (CCEVS Policy 15); fixed typos.
0.6	2008-11-20	David Ochel	Evaluation feedback and internal review.	Added explanation of arrows in Figure 1; corrected list of runtime environments; explained FIPS mode in 1.4.1.6; modified user terminology in T.UNAUTH; replaced references o O.Replication with O.Failover; added OE.Authentication (mapped to T.UNAUTH); changed various references from FAU_GEN.1 to FAU_GEN_EXT.1 and added corresponding rationale in dependency analysis; replaced/clarified definition of "authorized users"; added unresolved references from SFRs; fixed typos.
0.7	2008-12-04	David Ochel	Evaluator and developer feedback.	Introduced and defined operator term in 1.4.1.1; added section "Property encryption" in 1.4.1.2; changed list of supported OS versions in 1.4.1.5; replaced FAU_GEN_EXT.1 by FAU_GEN.1; no [] around operations on SFRs anymore; added details on restriction filters in 7.1.2; minor clarifications and correction of typos.
0.8	2008-12-09	David Ochel	Evaluator feedback.	Additional correction to arrow in Figure 1; removed redundant/confusing section on encryption utilities in 1.4.1.2/Security function management; added further details regarding the evaluated configuration in 1.4.1.6; adjusted A.SEL_PRO to cover all runtime environments; updated O.SecureComms to clarify that use of SSL/TLS is only an option for administrators to achieve the required network protection; augmented 7.1.6 with auditing of start/stop; added properties editor to 7.1.7; minor clarifications and correction of typos.
0.9	2008-01-23	David Ochel	Evaluator comments and internal review.	Added section on authentication providers in 1.4.1.4 and authentication provider in Figure 1; modified threat descriptions to better align with identification of threat agents; added A.COMM and OE.Communication.
1.0	2009-02-04	David Ochel	Evaluator feedback	Fixed mapping from OE's to SPD.

Version	Date	Author(s)	Changes to Previous Revision	Application Notes
1.1	2011-01-26	David Ochel	Review session with development teams.	Updated CC reference to R3. Added separate Web GUI access control policy to reflect the details of how access is enforced in the GUI (FDP_ACC.1-b and related modifications throughout the ST). Updated FPT_FLS.1 and FRU_FLT.1 to remove ambiguities. Throughout document, reflected the fact that the "Web GUI" is an integral part of the Netcool/OMNIbus product, rather than being the stand-alone "Webtop" component. Other clarifications.
1.2	2011-02-28	David Ochel	Further development review.	Updated the list of probes and gateways included in the evaluated configuration. Removed FPT_TRC.1, as TSF data is not typically synchronized between ObjectServers, and the synchronization mechanism also doesn't provide for absolute consistency (see TSS). Clarified the caching of ObjectServer TSF data in the Web GUI in FIA_USB.1.3 and the TSS. Editorial changes.
1.3	2011-06-30	David Ochel	See application note.	Added WAS auditing framework dependency in 1.5.1.4 and application note to FAU_GEN.1, updated lists of probes and gateways in 1.5.1.5, added AIX to list of platforms in 1.5.1.5, clarified in A.COMM that DES is not considered a security functionality, added AES and HMACs for SNMP v3 throughout document (P.SNMP-SEC, FCS_COP.1-e, etc.), mentioned unavailable Web GUI components in 1.5.1.6.
1.4	2011-07-15	David Ochel	See application note.	Fixed typos. Removed reference to Solaris 9 in 1.5.1.5.
1.5	2011-07-25	David Ochel	Test review.	Changed underlying Windows platform from 2003 to 2008.
1.6	2011-08-24	David Ochel	Clarifications.	More detailed explanations on some aspects; revised Figure 1 to make consistent with updates to evaluated configuration; removed Desktop from evaluated configuration; added WAAPI client; updated added SNMP Probe guide to list of guidance documents; editorial changes.
1.7	2011-09-15	David Ochel	Editorial changes, change in password policiesant.	Added Socket Gateway to TOE Description. Clarified inclusion of uni- and bi-directional ObjectServer Gateway in 1.5.1.5. Clarified requirement for user definition for Web GUI / ObjectServers in 1.5.1.6. More consistency in labeling iterated components and elements. Narrowed choices in FIA_ATD.1 and FIA_SOS.1 to "secure" parameters instead of depicting overall TOE capabilities. Editorial changes.
1.8	2011-10-03	David Ochel	Developer and evaluator reviews.	Updated version numbers; excluded native Event List client from evaluated configuration; fixed arrow to Socket Listener in Figure 1; added details on authentication

Version	Date	Author(s)	Changes to Previous Revision	Application Notes
				providers in A.COMM, OE.Administrators, OE.Communication; add User Preferences to FDP_ACF.1-b; added PAM attribute to FIA_ATD.1; clarifications and editorial changes.
1.9	2011-10-12	David Ochel	Developer and evaluator reviews.	Clarifications on socket gateway authentication; version number updates; editorial changes.
2.0	2011-10-20	David Ochel	Developer feedback.	Removed SSLv3 support; further clarification on nco_sql use; WebGUI FP version number update.
2.1	2011-11-03	David Ochel	Developer and evaluator feedback.	Restricted SNMP to encrypted v3; included "execute" operation in FDP_ACC.1-a; in FDP_ACF.1.2-a and 7.1.2.1, clarified distinction between object and system permissions; updated guidance document versions; added iscadmins role to FMT_SMR.1.
2.2	2011-11-08	David Ochel	Developer and evaluator feedback.	Added DB2 to the operational environment; added support of TLSv1.1; editorial changes.
DRAFT	2012-02-01	David Ochel	tVOR results, evaluator, and developer comments.	Clarifications on phyiscal TOE boundary; correction to Figure 1; clarified version numbers of guidance docs; added details to Web GUI access control policy; editorial changes.
DRAFT	2012-02-02	David Ochel	further refinements	Refinements of previous changes; color-coded authentication mechanisms in Figure 1.
2.3	2012-02-08	David Ochel	Developer feedback.	Minor clarification in FDP_ACF.1-b.
2.4	2012-04-20	David Ochel	Developer feedback.	Removed interface from SQL Interactive Interface to Socket Gateway.
2.5	2012-04-20	David Ochel	Developer feedback.	Clarified that previous change also applies to ObjectServer gateway.
2.6	2012-06-13	David Ochel	Evaluator feedback.	Updated fix pack for WebGUI, added CC Guide as guidance document, and added exclusion of nco_confpack and nco_osreport utilities.
2.7	2012-06-13	David Ochel	Typo fix	
2.8	2012-09-07	David Ochel	Developer feedback.	Added ifixes in section 1.5.1.5.
2.9	2012-09-18	David Ochel	Evaluator feedback	Added iFix for WAS in 1.5.1.5.
2.10	2012-11-09	Scott Chapman	Evaluator feedback.	

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Security Target Identification

Title:	IBM Tivoli Netcool/OMNIbus 7.3.1 Security Target
Version:	2.10
Status:	Released
Date:	2012-11-09
Sponsor:	IBM Corporation
Developer:	IBM Corporation
Keywords:	Netcool/OMNIbus

1.2 TOE Identification

The TOE is IBM Tivoli Netcool/OMNIbus Version 7.3.1 with Core Fix Pack 3 and WebGUI Fix Pack 3.

1.3 TOE Type

The TOE type is Event Management.

1.4 TOE Overview

Netcool/OMNIbus is an enterprise network and service level management (NMS-SLM) system that uses software components called "probes" to collect enterprise-wide event information from many different network data sources and presents a simplified view of this information to administrators so that they can monitor events in their environment. Netcool/OMNIbus tracks alert information in a high-performance, in-memory database (the ObjectServer) and presents information of interest to specifically identified and authenticated users through individually configurable filters and views. User activity can be accounted for and audited using the administration facilities provided by Netcool/OMNIbus. Users can access the event information assigned to them from a client application or via a Java-enabled browser connecting to the Netcool Web GUI. The TOE includes a probe to collect SNMP data.

The ObjectServer can authenticate users against an internal database, or use external providers to derive authentication decisions. It is then able to enforce an access control policy that mandates which users can perform which sort of operations on the collected data. Security-relevant actions can be audited, and the TOE provides TLS encryption of network traffic between its distributed components. Also, the TOE offers various features that provide fault tolerance, namely replication of redundant ObjectServers and the possibility for users/probes to access backup servers if their primary ObjectServer is not available.

The evaluated configuration of the TOE supports installations on AIX, Solaris, and Windows, and requires a WebSphere Application Server for the Web GUI server application.

1.5 TOE Description

1.5.1 TOE Architecture

1.5.1.1 Overview

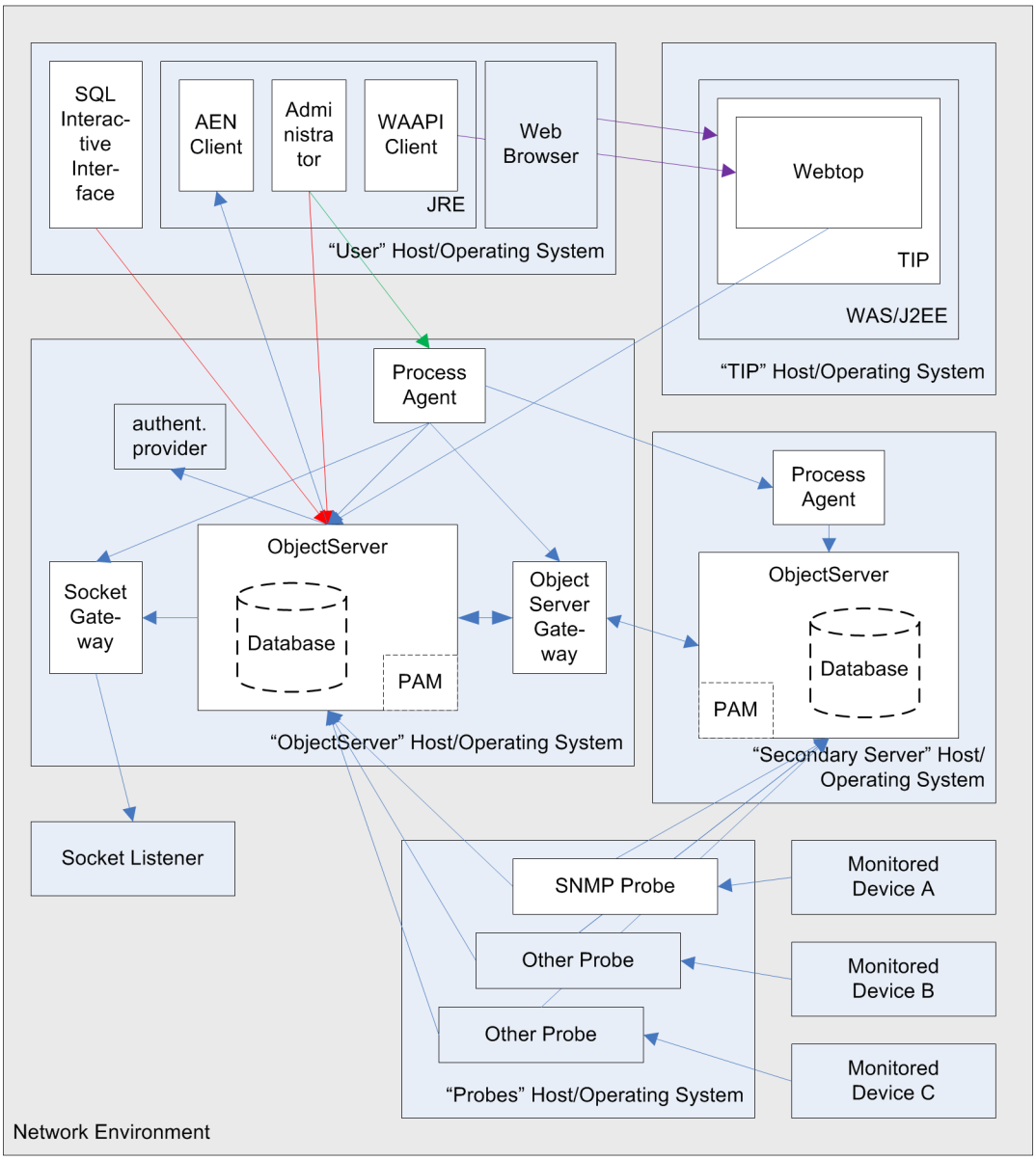

Figure 1: TOE architecture and boundary (white components are part of the TOE)

Netcool/OMNIbus is an enterprise network and service level management (NMS-SLM) system that collects enterprise-wide event information from many different networked data sources and presents a simplified view of this information to administrators. Netcool/OMNIbus tracks alert information in

a high-performance, in-memory database and presents information of interest to specifically identified and authenticated users (in the guidance documentation referred to as operators) through individually configurable filters and views. User activity can be accounted for and audited using the administration facilities provided by Netcool/OMNIbus. Users can access the event information assigned to them from a client application or via a Java-enabled browser connecting to the Web GUI.

The Web GUI is a web server application that processes network alert information and presents the data output to users so that they can monitor events in their Netcool/OMNIbus environment.

Figure 1 presents an exemplary view of a Netcool/OMNIbus deployment, illustrating the various components that the TOE (white boxes) and its operational environment (blue shaded boxes) are comprised of. Arrows generally indicate information flows/connectivity between components - in particular, an arrow pointing from component A to component B would indicate that component A initiates communication with component B. In addition, different colors for arrows have been used to illustrate the mechanisms in use for authenticating external entities (users) as follows:

- red: authentication enforced by the ObjectServer mechanism implemented by the TOE (either based on authentication decisions derived from the ObjectServer database, PAM, or an LDAP server)
- green: authentication implemented by the underlying system (native operating system authentication or PAM)
- purple: authentication implemented by WebSphere's Virtual Member Manager

Connections between TOE parts are generally protected (and authenticated) by means of TLS, as well as the Web GUI (implemented by TIP and Webtop) authenticating against the ObjectServer as "root" user.

The following sections will provide further explanation.

ObjectServer

The ObjectServer is a proprietary main memory database server at the core of Netcool/OMNIbus. Alert information is forwarded to the ObjectServer from external programs (probes) and stored and managed in database tables. Outside the evaluated configuration, a special configuration of an ObjectServer as a Proxy Server (not displayed in Figure 1) is available to multiplex connections from probes before sending them to another ObjectServer.

The SQL interface of the ObjectServer, apart from allowing probes to insert forwarded events into the database, provides a facility for administrators to define procedures, triggers and flow control constructs. Netcool/OMNIbus comes with a set of pre-defined triggers for deduplication of events, state changes for alert updates, etc.

The ObjectServer implements persistence of data using disk-based checkpoints and logs. Checkpoints write all data to disk at system-defined intervals to enable data recovery if the server stops unexpectedly. Between checkpoints, additional modifications to the database are logged to a journal file.

Probes

Probes detect and acquire event data (sometimes referred to as alerts), and forward the data to the ObjectServer. Probes use the logic specified in a rules file to manipulate the event elements before converting them into SQL insert statements for the ObjectServer's database.

Each probe is uniquely designed to acquire event data from specific sources. Probes can acquire data from any stable data source, including devices, databases, and log files. The probes can be installed on the same host as the ObjectServer or on a remote host.

Probes can be configured to send event data to a secondary server in cases where the primary ObjectServer cannot be reached. They can also be configured to store data in a "store and forward" file, in case no ObjectServer can be reached.

Some probes have a command port to accept commands from a Process Agent. However, the probe included in the evaluated configuration do not have this feature.

Integration Gateways

More than one ObjectServer may be installed in one deployment. ObjectServer Gateways can be used for communication between ObjectServers, in particular to replicate changes to event data (and optionally, configuration data) between different ObjectServers. Gateways can be used as bi-directional (fail-over) gateways between a primary and a backup ObjectServer, or as uni-directional connection between hierarchical servers that only forwards correlated and reduced event data to higher layers.

Gateways may be located on the same machine as one of the ObjectServers, or may be running on a third system. Several gateways may be used to facilitate replication between different servers.

Besides ObjectServer Gateways, the Netcool/OMNIbus family provides a number of other integration gateways that allow ObjectServers to exchange information with complementary third-party applications, such as helpdesk or Customer Relationship Management systems. (Not shown separately in the figure above.)

Socket Gateway

The Socket Gateway, also known as Socket Writer Gateway, forwards alerts using a TCP connection. Any program that listens to that socket (referred to as Socket Listener) receives the alerts.

Process Agent

Process Agents provide process control functionality. They can be used to start, stop, and monitor processes in a Netcool/OMNIbus installation in order to automatically restart failing processes, so-called "managed processes". For example, administrators can configure a Process Agent to start an ObjectServer, rather than starting the ObjectServer directly. In this case, the ObjectServer process becomes a managed process, and the Process Agent will provide a watch dog functionality that will restart the ObjectServer if it fails, and will report to clients about the status of the ObjectServer. Consequently, managed processes always run on the same host as the Process Agent itself. Process Agents can also be used to execute a variety of commands on behalf of other Netcool/OMNIbus components (see below). Process Agents accept connections from ObjectServers, Administrators, a number of command-line utilities, and other Process Agent-aware processes that connect to the Process Agent that started them to report their status (such as certain probes).

Process Agents can connect to each other to form a network so that a Process Agent running on one machine can be asked to run a process on another machine. The local Process Agent will then send a message to the Process Agent running on the target machine, which will run the process. If this is a "managed process", the remote Process Agent will be updated with its status so that it can return the status of both local and remote processes when interrogated by a client.

Besides managing processes, Process Agents can execute arbitrary commands requested by OMNIbus components. An example included in the ObjectServer's default configuration is the sending of email: an ObjectServer can send a message to a Process Agent that causes the Process Agent to invoke a mail transfer agent and send an email on behalf of the ObjectServer.

TIP and Web GUI

TIP stands for Tivoli Integrated Portal, a consolidated web-interface for various Tivoli products, such as Netcool/OMNIbus. TIP is part of the TOE and provides a common framework to be used by Tivoli products to present administrative and user interfaces. TIP itself runs within a WebSphere Application Server (WAS), which is part of the TOE environment.

Netcool/OMNIbus implements its Web GUI within TIP by inserting a product-specific "Webtop" component into it. Administrators and users use the Web GUI to display and handle events collected by one or multiple ObjectServers. Various interfaces can be made available by administrators to users, such as:

- The Java-based active events list (AEL) allows clients to execute actions such as acknowledging alerts, viewing alert journals, taking ownership of alerts, running tools, and so forth.
- The dynamic HTML lightweight event list (LEL) provides clients with the data filtering, data sorting, and information drill-down capabilities of the AEL.
- The HTML tableview component provides clients with a static event list in the form of a table showing a defined set of alerts. The non-interactive tableview provides an immediate snapshot of alert status within a monitored system.

WAAPI Client

The WAAPI client is a Java-based utility that can be used to remotely administer the Web GUI server. Configuration instructions are written in XML and are stored in a text command file. The WAAPI client sends the command file directly to the Web GUI server, which, after authenticating the client, will update its configuration accordingly.

Administrator and the SQL interactive interface

The Administrator is a piece of software that appears as “Administrator Config” after installation.

The Administrator is a Java application that provides a GUI for administrators to configure and manage both ObjectServers and Process Agents.

In addition to the Administrator GUI, a command line interface called "SQL interactive interface" is provided that allows to connect to ObjectServers and issue SQL-based queries and configuration commands.

AEN Client (Accelerated Event Notification)

Netcool/OMNIbus allows administrators to define events for accelerated event notification. Probes can be configured to flag certain events for acceleration, and/or triggers in ObjectServers can be defined that will cause the ObjectServer to identify events for accelerations. It is then possible to define "channels" defining which pieces of such events are being forwarded during accelerated event notification, and to which recipients they will be broadcasted. A dedicated AEN client is provided as part of the TOE for display of these events on an administrator's or user's desktop, in addition to the Web GUI receiving the event during periodic polls of the ObjectServer.

1.5.1.2 TOE Security Function (TSF) Summary

The TOE implements the security functionality described in this section.

Identification and authentication

The **ObjectServer** performs authentication of users connecting to it using either an internal repository of users and their passwords, or an external authentication service (shown as "authentication provider" in Figure 1) in the environment. The ObjectServer can be set to use PAM (on platforms supporting PAM) or LDAP for external authentication, offering a variety of authentication methods depending on the PAM modules available. In addition, the ObjectServer implements direct support for authentication against LDAP servers, including ActiveDirectory LDAP servers on Windows. If the ObjectServer is configured to use an external authentication provider, individual user accounts are still being maintained in the ObjectServer's user database, and must be individually configured to be authenticated against the external service.

Users in this authentication context can be both human users or other TOE components. Other TOE components that the ObjectServer authenticates in this way (when running in secure mode) are gateways, probes, Web GUI, and proxies (the latter not being part of the evaluated configuration).

Netcool/OMNIbus, when running on UNIX, also implements an ObjectServer PAM module that other components can use to authenticate against. This allows applications using PAM to authenticate against the user database maintained by the ObjectServer, and includes functionality to change user passwords via the PAM module.

In the evaluated configuration, the ObjectServer supports both PAM (on UNIX) and Active Directory (on Windows) for the authentication of users, as well as its internal authentication mechanism.

Configuration and management utilities that do not connect to the ObjectServer are not subject to the ObjectServer-enforced authentication.

ObjectServer Gateways and **Process Agents** running on UNIX authenticate users that connect to their command port, either via PAM, or against the local UNIX password database (users must be in the ncoadmin group). Process Agents (but not ObjectServer Gateways) running on Windows use the operating system to authenticate the requesting user. Every local or domain user account having access to the Windows host can therefore connect.

Managed processes that report their status back to a Process Agent are not authenticated.

The **TIP** server hosting the Web GUI does not implement authentication, but relies instead on its runtime environment for authentication (see below).

When TLS is being used (see Communications security below), certificate-based server authentication is employed in the following communication scenarios:

- Gateways, probes, administrative utilities, and the Web GUI authenticate ObjectServers they connect to.
- Clients (e.g., ObjectServers, Administrators) authenticate the Process Agents they connect to.

As a result, connecting clients will verify that the server certificate a) has been issued by a trusted CA and b) contains the Common Name of the server that the client is trying to reach.

The use of TLS is mandatory in the evaluated configuration.

To support rapid deployment in test and secure environments, the ObjectServer, proxy server, and Process Agent will allow certain types of client application to connect without authentication. In the evaluated configuration, this feature must be disabled by running the ObjectServer and Process

Agents in secure mode (use of the "SecureMode: TRUE" property on the ObjectServer). In this mode, probe, Socket Gateway, and integration gateway connections to an ObjectServer are authenticated with a user name and password. When the Process Agent is run in secure mode, connections are authenticated before external procedures are run. (Proxy servers are excluded from the evaluated configuration.)

Discretionary access control

The ObjectServer implements a fined-grained authorization system based on users, groups, permissions, and roles. As a result of the ObjectServer being first and foremost a database server, this authorization system primarily targets operations on the databases hosted by the server. Permissions for individual actions (such as create, insert, alter) on individual objects (tables, rows, triggers, etc.) can be granted to users directly or combined into roles.

Administrators can further define groups that have several users as members, and associate roles defining specific permissions with individual users or groups.

In addition, so-called restriction filters can be defined and assigned to users or groups. Restriction filters are SQL conditions that are being applied when a user accesses data in the database.

Permissions and restriction filters are also applied to queries that users send via the Web GUI, if the requesting user name matches a user name associated with such permissions and restrictions in the ObjectServer. If the user is unknown to the ObjectServer, only read-only access will be granted. (In this case, Web GUI users can read all data that is available to them on the GUI pages that the Virtual Member Manager in the environment grants them access to, as further explained below.)

Auditing

Actions taken by users generate audit records. These records contain the date, time, event type, identity of the user, and outcome of the action. Only administrators have the ability to review and clear these records.

Communications security

Almost all network connections between remote components of the TOE can be (and, in the evaluated configuration, must be) encrypted using TLS.

In addition, remote entities authenticate their communication partners as described in the section on authentication above.

IDUC (insert, delete, update or control) communication is the only unsecured (not encrypted) communication channel in the evaluated configuration of the TOE. IDUC connections are used by ObjectServers to inform the gateways about event updates being available from the server, and is used for data sent through AEN channels.

Fault tolerance

The TOE implements several mechanisms that contribute to the availability and correctness of event data:

1. The TOE can replicate data between multiple ObjectServers to ensure consistency of data, even if one of the servers is temporarily unavailable, using bi-directional configurations of ObjectServer Gateways.
2. The ObjectServer maintains a transaction journal file to avoid loss of event data that has not been written to permanent storage yet.

3. Probes can be configured to use a backup ObjectServer if the primary ObjectServer is not available, and store event data in local files for later forwarding if no ObjectServer is available.

Property encryption

The TOE implements functionality that can, in addition to the protection measures already in place in the operational environment, be used to further prevent unauthorized access to configuration and TSF data by encrypting it. This functionality is available for:

- property values stored in configuration files for ObjectServers, Probes, and Gateways
- passwords stored on the Web GUI servers
- passwords stored in ObjectServer databases

Security function management

The TOE offers various methods to administrate and configure security functions, including GUI-based standalone and web-based clients, and command line interfaces, as described above.

1.5.1.3 Cryptographic support within the TOE

While not a TOE security function in itself, the TOE employs cryptographic mechanisms to support various of the security functions described above. The Netcool/OMNIbus TOE includes the Global Security Kit (GSKit), consisting of a cryptographic library (ICC) as well as an TLS wrapper. In particular, the following functionality is provided:

- TLS connectivity between Object Servers, gateways, probes and client browsers.
- Optional encryption of user passwords on the ObjectServer using AES CBC.
- Property value/authentication credentials encryption in configuration files using AES CBC.

The GSKit with ICC component in the version used by the TOE is both FIPS 140-2 certified and has undergone Common Criteria evaluation.

1.5.1.4 Security architecture and support from the operational environment

The TOE, being an application running on top of various runtime environments, relies on security functionality provided by these environments for a) its own protection and b) support for certain security functionality implemented by the TOE.

Unix/Windows-provided security functionality

The distributed parts of the TOE (other than the Web GUI) are applications running on top of UNIX or Windows operating systems. As such, they rely on the operating system in a number of ways in order to prevent the circumvention of TSF:

- Protection of the TOE and TSF data. The underlying operating system is responsible for limiting access to files that represent the TOE (binaries, etc.) or store TSF data (configuration files, database files, log files, etc.) to authorized subjects.
- Domain separation. The operating system is responsible for providing an execution domain for the TSF that protects the TSF from interference and tampering by unauthorized subjects.

The operating system provides support for the TSF as follows:

- Authentication of users. As described above, several components of the TOE make use or can make use of the underlying operating system, Active Directory, or PAM modules offered on UNIX systems, to deliver authentication mechanisms to them.
- Reliable timestamps. In order for the TOE to generate audit records, it relies on the underlying operating system to provide the current time for inclusion in the records.

WebSphere-provided security functionality

The Web GUI runs in a WebSphere Application Server (WAS) environment. This runtime environment is not part of the TOE, but has been subject to a separate CC-evaluation. The following security functionality of relevance for the TOE is provided by WAS:

- Authentication framework. WAS provides the Virtual Member Manager (VMM) as an authentication and authorization framework for the Web GUI. Users are authenticated either based on VMM's own proprietary (flat file) database of users, against an LDAP server, or against an ObjectServer (via JDBC).
- Role-based access control to user interface. WAS allows to restrict access to the individual views implemented by Web GUI roles that are defined in the VMM during installation of the Web GUI. Management of roles and enforcement of access to GUI pages based on these roles is performed by WAS.
- Protection of the TSF and TSF data, and domain separation. In the same fashion as the operating systems above are responsible for protecting the native applications running on top of them, WAS is responsible for protecting the Web GUI from unauthorized access and interference.
- TLS connectivity. WAS provides the cryptographic and protocol facilities for TLS connections used by the Web GUI, including the IBMJCEFIPS and IBMJSSEFIPS cryptographic modules.
- Auditing framework. The Web GUI uses the auditing facility provided by WAS for recording any Web GUI-generated audit records. This requires the configuration of the correct syslog level in WAS.

DB2

Multiple Web GUI servers can be set up in a load balancing configuration, in which case a DB2 database is required as the repository for configuration information, rather than local configuration files on the servers. In this case, the Web GUI relies on DB2 for the protection of configuration data.

Other authentication providers

Support for the TSF may - depending on the configuration - also be provided by authentication providers other than the operating system, i.e. by LDAP servers in the operational environment. These authentication providers deliver authentication decisions to the TOE which will then be enforced by the TSF.

1.5.1.5 Physical Boundary

The TOE is software only. It is distributed on CD-ROM or can be obtained for download via IBM's Passport Advantage program. Only the latter is included in this evaluation.

The software packages comprising the TOE are:

- IBM Tivoli Netcool/OMNIbus Version 7.3.1, with:
    - Core Fix Pack 3

- WebGUI Fix Pack 3
- iFix PM36620
- iFix PM71389

The following list identifies (optional) product components and whether or not they are part of the TOE's evaluated configuration:

- The only probes that are part of the evaluated configuration are:
  - SNMP Probe (nco_p_mttrapd)
- The only gateways that are part of the evaluated configuration are:
  - ObjectServer Gateway (both uni- and bi-directional)
  - Socket Gateway (nco_g_socket, also known as Socket Writer Gateway)
- The Proxy Server is not part of the evaluated configuration.

The customer assumes the risk of using non-evaluated components in the operational environment.

The runtime environments for the TOE components that are included in the evaluation are:

- for the ObjectServer, gateway, process agent, SQL Interactive Interface, and SNMP probe:
  - AIX 7.1, Solaris 10 on SPARC, or Windows 2008 Server
  - on Solaris: Motif 1.2 or CDE

  Note: *The evaluated configuration expects that all components of an installation of the TOE (except probes) run on the same platform type, i.e., only on one of the platforms listed above.*

- for the Administrator and AEN client:
  - IBM JRE 1.6.0, release 9, including fix PM40694

    Note: *The product may run on other JRE editions as well. However, the only configuration tested as part of the evaluation is the one listed here.*

- for the Web GUI:
  - embedded WebSphere Application Server (eWAS) 7.0.0.17 (shipped with the TOE), with iFix IFPM40694
  - DB2 Workgroup Server Edition 9.7 CPU Option or better for use as common configuration repository, only if load balancing of Web GUI servers is employed (shipped with the TOE)

The TOE does not have any direct dependencies on hardware platforms. All environmental support for TSF is provided by the TOE's runtime environments, which provide an abstraction layer from the physical (and, in the case of eWAS, from the operating system) layer of the systems, making it unnecessary to specify these as part of the evaluated configuration.

Relevant guidance documents for the secure operation of the TOE are:

- IBM Tivoli Netcool/OMNIbus Version 7.3.1 Common Criteria Guide
- IBM Tivoli Netcool/OMNIbus Version 7.3.1 Administration Guide
- IBM Tivoli Netcool/OMNIbus Version 7.3.1 Installation and Deployment Guide
- IBM Tivoli Netcool/OMNIbus Version 7.3.1 Probe and Gateway Guide
- IBM Tivoli Netcool/OMNIbus Version 7.3.1 User's Guide
- IBM Tivoli Netcool/OMNIbus Version 7.3.1 Web GUI Administration and User's Guide

- IBM Tivoli Netcool/OMNIbus Probe for SNMP Version 13.0 Reference Guide
- IBM Tivoli Netcool/OMNIbus Socket Writer Gateway Version 10.0 Reference Guide

The editions applying to version 7, release 3, modification 1 (7.3.1) of IBM Tivoli Netcool/OMNIbus must be used.

1.5.1.6 Logical Boundary (evaluated configuration)

The logical boundary of the TOE is depicted by the security functions described in section 1.4.1.2. In addition, the following configuration specifics apply to the evaluated configuration of the TOE:

- ObjectServers must be configured to authenticate users against the internal user database, and/or against an LDAP server (UNIX, i.e. AIX and Solaris) or Active Directory (Windows) in the environment.
- ObjectServer Gateways and Process Agents must be configured to derive authentication decisions from their runtime environment, either via PAM (on UNIX) or the native operating system authentication mechanism (on Windows)
- TIP must be configured to authenticate users against an LDAP server. In particular, administrators are responsible for ensuring that the user IDs configured in the authentication provider for TIP (i.e., for the Web GUI) are consistent with the user IDs for the ObjectServer in order for the Web GUI access control policy defined in this ST to be enforced completely.
- ObjectServers and Process Agents must be configured to run in "secure mode" and in FIPS 140-2 mode, causing the TOE to use FIPS-approved cryptographic algorithms only.
- TLS for communication between TOE components must be enabled if no other means to provide for integrity and confidentiality of network communication between TOE parts are implemented in the operational environment.
- The encryption of passwords and other property values with the TOE-provided means is included in the evaluation and recommended as a security layer in addition to the protection provided by the operational environment, but its use is optional.
- The TOE has been tested in environments where all TOE components run either on UNIX or on Windows. Mixed environments have not been tested and are therefore not covered by this evaluation.
- The evaluated configuration supports both uni-directional (i.e., hierarchial) ObjectServer Gateway and bi-directional (i.e., configured for failover) ObjectServer Gateway configurations. An ObjectServer Gateway can either be installed on the same host as an ObjectServer, or on a separate host.
- The evaluation does not make any claims on the re-synchronization of TSF data between primary and backup ObjectServer after a failover situation. Consumers typically operate multiple ObjectServers by populating configuration (TSF) data via external means, rather than using the ObjectServer's gateway mechanism. This is further addressed in the guidance.
- The TOE comes with iKeyman, a graphical utility to generate certificates that can be used for TLS communication. iKeyman is an administrative utility (wrapper) that can be used instead of accessing the GSKit component's key and certificate management command-line-interface. Keystores created with this GUI are not usable in FIPS mode, and hence, iKeyman cannot be used in the evaluated configuration. Consumers are generally expected to use certificates provided within their PKI.
- The native *Desktop* and *Event List* clients component of the TOE are superseded by the functionality made available through the WebGUI and is not available in the evaluated configuration.

- The Web GUI components *Charts* and *Page Manager* are not available in the evaluated configuration.
- The SNMP probe in the evaluated configuration uses AES-encrypted SNMPv3 exclusively.
- Communication from the SQL Interactive Interface to Socket and ObjectServer Gateways is disabled.
- The nco_confpack and nco_osreport utilities are not included in the evaluated configuration.

1.5.2 Security Policy Model

The security policy for the TOE is defined by the security functional requirements. The following is a list of the subjects and objects, and their security attributes, that are participating in the policy.

1.5.2.1 Subjects

Subjects are the users defined in the ObjectServer database or, in case of the Web GUI (and TIP), in external authentication providers (LDAP).

Their security attributes are:

- user name (ID) - a unique ID identifying the user
- password - authentication credential used to login
- restriction filters - while these filters are data objects stored in the ObjectServer, they may be assigned to users or groups, in which case they become a security attribute for the user that is relevant for the enforcement of access control
- ObjectServer group membership - while groups are data objects stored in the ObjectServer, they may be assigned to users (the user assigned to a group becomes a member of the group), in which case they become a security attribute for the user that is relevant for the enforcement of access control
- user filters - data objects stored in the Web GUI that may be assigned to users
- Web GUI group membership - defined in the Web GUI
- Web GUI roles - defined in WAS / VMM
- account status - enabled or locked

1.5.2.2 Objects

Objects are the data elements stored in the ObjectServer's database tables. This includes, but is not limited to, the following examples:

- alert entries
- triggers and trigger groups
- SQL and external procedures
- etc.

Objects are stored in the ObjectServer's database tables, which means that they can be summarized in databases, tables, columns, and rows (i.e., individual table entries).

Their security attributes are:

- permissions - indicating (a) user(s) and/or (a) group(s) that has/have access to an object
- roles - sets of permissions that are assigned to (summarized into) a role

Furthermore, GUI elements in the Web GUI that are governed by the Web GUI access control policy are considered objects in the context of this policy.

1.5.2.3 TSF and user data

TSF data can be identified as:

- the security attributes for subjects defined above
- the security attributes for objects defined above
- event data that serves as audit records
- security-relevant configuration properties for the ObjectServer and other TOE components, such as the ones determining the audit level, use of secure mode, FIPS-compliant operation, the AlertSecurityModel property, the registration of CGI scripts in the Web GUI, etc.

User data is:

- all other event data collected by probes and recorded by the TOE

2 CC Conformance Claim

This ST is CC Part 2 conformant and CC Part 3 conformant, with a claimed Evaluation Assurance Level of EAL4, augmented by ALC_FLR.3.

This ST does not claim conformance to any Protection Profile.

Common Criteria [CC] version 3.1 revision 3 is the basis for this conformance claim.

3 Security Problem Definition

3.1 Threat Environment

This section describes the threat model for the TOE and identifies the individual threats that are assumed to exist in the TOE environment.

The **assets** to be protected by the TOE are the TOE's own TSF data, as well as the event data that is being collected, processed, and stored by the TOE.

The **threat agents** having an interest in obtaining or tampering with these assets can be categorized as either:

- Unauthorized individuals ("attackers") which are unknown to the TOE and its runtime environment.
- Users of the TOE (i.e., who have per the defined ObjectServer access control policy access to at least parts of the TSF and assets) who try to access data that they are not authorized to access.

This evaluation aims to demonstrate that the TOE is able to withstand attackers with an "Enhanced-Basic" attack potential.

TOE administrators, including administrators of the TOE environment, are assumed to be trustworthy, trained and to follow the instructions provided to them with respect to the secure configuration and operation of the systems under their responsibility. Hence, only inadvertent attempts to manipulate the safe operation of the TOE are expected from this community.

3.1.1 Threats countered by the TOE

T.AUD

Security-relevant activities of unauthorized individuals (such as login attempts) and users (such as configuration changes) may go unnoticed.

T.COMPDATA

Event data may be compromised (disclosed to or modified by unauthorized individuals or users) while being transferred between distributed parts of the TOE.

T.DATABASEFAIL

Database unavailability, either due to system failure or due to intervention of unauthorized individuals, may prevent event information from being recorded.

T.UNAUTH

Unauthorized individuals or users may access configuration and event data without proper authorization.

3.2 Assumptions

3.2.1 Intended usage of the TOE

A.INSTALL

It is assumed that the TOE is configured and operated in its evaluated configuration as defined in this Security Target and the TOE guidance.

3.2.2 Environment of use of the TOE

3.2.2.1 Physical

A.PHYSEC

It is assumed that the machine(s) providing the runtime environment for the TOE are protected against unauthorized physical access and modification.

3.2.2.2 Personnel

A.TRUSTED

The administrators of the TOE, of the TOE's underlying systems, and of the systems in the TOE's operational environment who are involved in safeguarding TSF data or providing functionality that the TOE depends on are assumed not to be careless, willfully negligent, or hostile. They will follow and abide by the instructions provided in the administrator guidance that is part of the TOE. They are well trained to securely and trustworthy administer all aspects of the TOE operation in accordance with this Security Target.

3.2.2.3 Connectivity

A.SEL_PRO

The machines providing the runtime environment for the server components of the TOE (i.e., all components other than those running on user hosts) are assumed to be used solely for this purpose and not to run other application software except as required for the support of the TOE and for the management and maintenance of the underlying system and hardware.

Especially it is assumed that the underlying systems for all TOE components are configured in a way that prevents unauthorized access - either locally or via any network-based connections - to security functions, TSF and user data, including audit records generated by the TSF.

A.COMM

It is assumed that the protection of communication over inherently insecure protocols between the TOE and remote IT entities is protected by environmental means as appropriate for the operational environment. This also includes unencrypted communication between probes and any HTTP server that may be used to serve rules files to probes, communication between TOE components over the unencrypted IDUC protocol, and communication between the TOE and authentication providers.

3.3 Organizational Security Policies

P.SNMP_SEC

The TOE shall offer an option to provide for integrity and authentication of SNMP v3 messages.

4 Security Objectives

4.1 Objectives for the TOE

O.Auditing

The TOE shall offer a mechanism that can be used to hold users of the TOE accountable for specified security-relevant actions.

O.Discretionary_Access

The TOE must control access to resources based on the identity of users. The TSF must allow authorized administrators to specify which resources may be accessed by which users.

O.Password_Encryption

The TOE shall offer a mechanism to encrypt configuration data stored in flat files in the runtime environment, in particular authentication data.

O.Failover

The TSF shall offer fault tolerance towards the unavailability of an ObjectServer in a system of ObjectServers with failover configuration. In a failover configuration, the TSF must ensure that data is replicated consistently between primary and backup database.

O.SecureComms

The TSF must be able to securely transfer data between the servers and clients that comprise the TOE, and between SNMP probes and data sources in the operational environment.

4.2 Objectives for the Operational Environment

OE.Authentication

The runtime environment for the TOE shall implement authentication mechanisms commensurate with the level of protection sought by the TOE, and provide authentication decisions for TOE users to the TSF.

OE.TimeSource

The runtime environment for the Server shall provide a reliable time source for the TOE’s use.

OE.Administrators

Those responsible for the operation of the TOE must ensure that administrators are not careless, willfully negligent, or hostile, and that they are well trained and will follow the provided administrator guidance to configure and operate the TOE and the TOE environment.

This includes ensuring that all access credentials are protected against disclosure by the users of the TOE, and that only trusted authentication providers are used.

OE.Runtime

Those responsible for the operation of the TOE must ensure that the systems hosting the server components are used solely for this purpose and configured in a way that prevents unauthorized access to the TOE and any TSF and user data, including audit records generated by the TSF.

This includes preventive measures to ensure that all systems that are hosting parts of the TOE are protected against unauthorized physical access and network-based attacks.

OE.Communication

Those responsible for the operation of the TOE must assess the risks of communication between TOE probes and systems in the operational environment via protocols that are inherently insecure (i.e., known vulnerabilities exist); between probes and HTTP server in the operational environment that might be used to serve rules files to probes over unencrypted communication channels; between TOE components and authentication providers; between TOE components via the unsecured IDUC protocol; and implement appropriate protection measures.

4.3 Security Objectives Rationale

4.3.1 Coverage

The following table provides a mapping of TOE objectives to threats and policies, showing that each objective counters or enforces at least one threat or policy, respectively.

Objective	Threats / OSPs
O.Auditing	T.AUD
O.Discretionary_Access	T.UNAUTH
O.Password_Encryption	T.UNAUTH
O.Failover	T.COMPDATA T.DATABASEFAIL
O.SecureComms	T.COMPDATA P.SNMP_SEC

Table 1: Mapping of security objectives to threats and policies

The following table provides a mapping of the objectives for the Operational Environment to assumptions, threats and policies, showing that each objective holds, counters or enforces at least one assumption, threat or policy, respectively.

Objective	Assumptions / Threats / OSPs
OE.Authentication	T.UNAUTH
OE.TimeSource	T.AUD
OE.Administrators	A.INSTALL A.TRUSTED

Objective	Assumptions / Threats / OSPs
OE.Runtime	A.PHYSEC A.SEL_PRO
OE.Communication	A.COMM

Table 2: Mapping of security objectives for the Operational Environment to assumptions, threats and policies

4.3.2 Sufficiency

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

Threat	Rationale for security objectives
T.AUD	O.Auditing requires the TOE to implement auditing of security-relevant events. This is supported by OE.TimeSource, which expects the operational environment to provide a reliable time source for the generation of audit record time stamps.
T.COMPDATA	O.SecureComms requires TSF that protect the communication between parts of the TOE to be protected against disclosure and manipulation. This is also applicable to the special case of replication between ObjectServers, as addressed by O.Failover.
T.DATABASEFAIL	ObjectServers can be configured in a redundant fail-over setup, as indicated in O.Failover, in which case a backup server can record events during the unavailability of the primary server, and then synchronize its event database with the primary server upon its availability.
T.UNAUTH	O.Discretionary_Access mandates the implementation of administrator-defined access control, which allows to restrict access to specific information to specific users. This implies the necessity for authentication mechanisms. The operational environment supports the TOE by providing authentication decisions for TOE users as in OE.Authentication. O.Password_Encryption offers encryption of configuration data stored in the operational environment as protection to be implemented by the TSF in addition to the protection mechanisms provided by the runtime environment.

Table 3: Sufficiency of objectives countering threats

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

Assumption	Rationale for security objectives
A.INSTALL	The assumption that the TOE will be configured and operated in its evaluated configuration is achieved by the objective OE.Administrators to ensure that administrators obey by the guidance.
A.PHYSEC	The assumption on physical protection of the TOE is achieved by the objective OE.Runtime to provide such protection.
A.TRUSTED	The assumptions that administrators are trustworthy and well trained is achieved by the objective OE.Administrators to ensure these properties of administrators.
A.SEL_PRO	The assumption on exclusive TOE use of the underlying machines for the TOE and preventing unauthorized access is achieved by the objective OE.Runtime to implement corresponding measures for the runtime environment.
A.COMM	The assumption to provide protection of network communication between the TOE and systems in the operational environment is upheld by OE.Communication, which phrases a corresponding objective for the operational environment.

Table 4: Sufficiency of objectives holding assumptions

The following rationale provides justification that the security objectives are suitable to cover each individual organizational security policy, that each security objective that traces back to an OSP, when achieved, actually contributes to the implementation of the OSP, and that if all security objectives that trace back to an OSP are achieved, the OSP is implemented:

OSP	Rationale for security objectives
P.SNMP_SEC	O.SecureComms requires the TOE to provide mechanisms that allow for the secure transmission of SNMP v3 messages.

Table 5: Sufficiency of objectives enforcing Organizational Security Policies

5 Extended Components Definition

This Security Target does not extend the security components provided by the Common Criteria.

6 Security Requirements

6.1 TOE Security Functional Requirements

The following table shows the Security functional requirements for the TOE, and the operations performed on the components according to CC part 2: iteration (Iter.), refinement (Ref.), assignment (Ass.) and selection (Sel.).

Security functional group	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
FAU - Security audit	FAU_GEN.1 Audit data generation		CC Part 2	No	No	Yes	Yes
	FAU_SAR.1 Audit review		CC Part 2	No	No	Yes	No
	FAU_STG.1 Protected audit trail storage		CC Part 2	No	No	No	Yes
FCS - Cryptographic support	FCS_CKM.1 Cryptographic key generation		CC Part 2	No	No	Yes	No
	FCS_COP.1-a Cryptographic operation	FCS_COP.1	CC Part 2	Yes	No	Yes	No
	FCS_COP.1-b Cryptographic operation	FCS_COP.1	CC Part 2	Yes	No	Yes	No
	FCS_COP.1-c Cryptographic operation	FCS_COP.1	CC Part 2	Yes	No	Yes	No
	FCS_COP.1-d Cryptographic operation	FCS_COP.1	CC Part 2	Yes	No	Yes	No
	FCS_COP.1-e Cryptographic operation	FCS_COP.1	CC Part 2	Yes	No	Yes	No
FDP - User data protection	FDP_ACC.1-a Subset access control	FDP_ACC.1	CC Part 2	Yes	No	Yes	No
	FDP_ACC.1-b Subset access control	FDP_ACC.1	CC Part 2	Yes	No	Yes	No
	FDP_ACF.1-a Security attribute based access control	FDP_ACF.1	CC Part 2	Yes	No	Yes	No
	FDP_ACF.1-b Security attribute based access control	FDP_ACF.1	CC Part 2	Yes	No	Yes	No
FIA - Identification and authentication	FIA_AFL.1 Authentication failure handling		CC Part 2	No	No	Yes	Yes
	FIA_ATD.1 User attribute definition		CC Part 2	No	No	Yes	No
	FIA_SOS.1 Verification of secrets		CC Part 2	No	No	Yes	No
	FIA_UAU.2 User authentication before any action		CC Part 2	No	No	No	No

Security functional group	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	FIA_UID.2 User identification before any action		CC Part 2	No	No	No	No
	FIA_USB.1 User-subject binding		CC Part 2	No	No	Yes	No
FMT - Security management	FMT_MSA.1-a Management of security attributes	FMT_MSA.1	CC Part 2	Yes	No	Yes	Yes
	FMT_MSA.1-b Management of security attributes	FMT_MSA.1	CC Part 2	Yes	No	Yes	Yes
	FMT_MSA.3-a Static attribute initialisation	FMT_MSA.3	CC Part 2	Yes	No	Yes	Yes
	FMT_MSA.3-b Static attribute initialisation	FMT_MSA.3	CC Part 2	Yes	No	Yes	Yes
	FMT_SMF.1 Specification of Management Functions		CC Part 2	No	No	Yes	No
	FMT_SMR.1 Security roles		CC Part 2	No	Yes	Yes	No
FPT - Protection of the TSF	FPT_FLS.1 Failure with preservation of secure state		CC Part 2	No	No	Yes	No
	FPT_ITT.1 Basic internal TSF data transfer protection		CC Part 2	No	No	No	Yes
FRU - Resource utilisation	FRU_FLT.1 Degraded fault tolerance		CC Part 2	No	No	Yes	No

Table 6: Security functional requirements for the TOE

6.1.1 Security audit (FAU)

6.1.1.1 Audit data generation (FAU_GEN.1)

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

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

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

c)
- **use of authentication mechanism,**
- **access control requests,**
- **creation / deletion / modification of objects in the database**

.

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

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

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST,

- **first and last occurence of the event**
- **severity of the event**

.

Application Note: *Note that the requirement for generating records for the start-up and shutdown of the audit functions does not apply to the Web GUI, as the operational environment (WAS) is reponsible for such records in this case.*

6.1.1.2 Audit review (FAU_SAR.1)

FAU_SAR.1.1 The TSF shall provide **users that have been authorized in accordance with the ObjectServer access control policy** with the capability to read **all audit information** from the audit records.

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

6.1.1.3 Protected audit trail storage (FAU_STG.1)

FAU_STG.1.1 The TSF shall protect the stored audit records in the audit trail 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.

6.1.2 Cryptographic support (FCS)

6.1.2.1 Cryptographic key generation (FCS_CKM.1)

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm

1. **TLSv1 and TLSv1.1 symmetric key and secret generation**

and specified cryptographic key sizes

1. **168 Bits (three independent TDEA keys),**
2. **128 and 256 Bits (AES key),**
3. **160 Bits (HMAC SHA-1 secret)**

that meet the following:

1. **conformant to [TLSv1] and [TLSv1.1] with [TLS_AES] (symmetric key and secret generation)**

.

6.1.2.2 Cryptographic operation (FCS_COP.1-a)

FCS_COP.1.1-a 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 [RFC2437] and [RFC2313] (RSA) and encryption/decryption of session key related data as defined in [TLSv1] and [TLSv1.1]**.

6.1.2.3 Cryptographic operation (FCS_COP.1-b)

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

1. **TDEA with three independent keys (CBC mode),**
2. **AES (CBC mode)**

and cryptographic key sizes

1. **168 Bits (TDEA),**
2. **128, 256 Bits (AES)**

that meet the following:

1. **conformant to [FIPS46-3] (TDEA), conformant to [FIPS81] (CBC mode),**
2. **conformant to [FIPS197] (AES, CBC mode),**

and [FIPS140-2] approved.

6.1.2.4 Cryptographic operation (FCS_COP.1-c)

FCS_COP.1.1-c The TSF shall perform **digest generation and verification** in accordance with a specified cryptographic algorithm **SHA-1** and cryptographic key sizes **none** that meet the following: **conformant to the Secure Hash Standard (SHS) as defined in [FIPS180-2] and [FIPS140-2] approved (SHA-1)**.

6.1.2.5 Cryptographic operation (FCS_COP.1-d)

FCS_COP.1.1-d The TSF shall perform **digital signature verification** in accordance with a specified cryptographic algorithm **RSA with SHA-1, SHA-224, SHA-256, SHA-384 or SHA-512 message digest,** and cryptographic key sizes **1024 or 2048 or 4096 Bits** that meet the following: **conformant to [RFC2437] and [RFC2313] (RSA) and [FIPS180-2] (SHA-1, SHA-224, SHA-256, SHA-384, SHA-512)**.

6.1.2.6 Cryptographic operation (FCS_COP.1-e)

FCS_COP.1.1-e The TSF shall perform **the generation of an HMAC** in accordance with a specified cryptographic algorithm **SHA-1** and cryptographic key sizes **not applicable** that meet the following: **conformant to [SP800-135] section 5.4**.

6.1.3 User data protection (FDP)

6.1.3.1 Subset access control (FDP_ACC.1-a)

FDP_ACC.1.1-a The TSF shall enforce the **ObjectServer access control policy** on **users as subjects; objects stored in the ObjectServer; and view, modify, create, execute, and delete operations of subjects on objects**.

6.1.3.2 Subset access control (FDP_ACC.1-b)

FDP_ACC.1.1-b The TSF shall enforce the **Web GUI access control policy** on **users as subjects; objects stored in the ObjectServer as well as GUI pages, portlets, views, console preference profiles, tools, and CGI scripts; and access of subjects to objects**.

6.1.3.3 Security attribute based access control (FDP_ACF.1-a)

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

- **subjects and their following security attributes:**
  - **user name (ID)**
  - **ObjectServer group membership**
  - **restriction filters**
- **objects and their following security attributes:**
  - **permissions**
  - **sets of permissions (roles)**

.

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

1. **If the object permission, or role, assigned to an object allows the requested operation for the user, or for the group that the user is a member of, the operation is allowed.**
2. **If the system permission, or role, associated with a user allows the requested operation for the user, or for the group that the user is a member of, the operation is allowed.**

.

FDP_ACF.1.3-a The TSF shall explicitly authorise access of subjects to objects based on the following additional rules:

1. **If the requesting user, or a group the user is a member of, has been associated with the SuperUser role.**

.

FDP_ACF.1.4-a The TSF shall explicitly deny access of subjects to objects based on the **following rules:**

1. **If a user, or a group that the user is a member of, is connecting to the ObjectServer via the SQL interactive interface, but has not been associated with the permission, or a role that contains the permission, to connect to the ObjectServer via the SQL interactive interface, the access is denied.**
2. **If a restriction filter that prevents the requested operation on the requested object is assigned to a user, or to a group that the user is a member of, the access is denied.**

.

6.1.3.4 Security attribute based access control (FDP_ACF.1-b)

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

- **subjects and their following security attributes:**
  - **user name (ID)**
  - **ObjectServer group membership**
  - **Web GUI group membership**
  - **restriction filters**
  - **user filters**
  - **Web GUI roles and associated access levels (User, Privileged User, Editor)**
  - **User Preferences, in particular:**
    - **Allow filter and view selection**
    - **Allow filter builder access**
    - **Allow view builder access**
    - **Allow preference configuration**
    - **Allow event selection**
    - **Show basic event information**
    - **Show event details**
    - **Show journals**
    - **Edit journals**
- **objects and the following security attributes:**
  - **Object name**
  - **Object type (portlets, pages, gauges, filters, views, scripts)**
  - **registration of CGI scripts**

.

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

1. **Web GUI roles are used to restrict access as follows:**

a. In order to be able to access any event management features of the Web GUI, a user must be associated with the *ncw_user* Web GUI role in addition to one of the *netcool_ro* (view events and the AEL only) or *netcool_rw* (change events and use AEL tools) Web GUI roles. The following exceptions apply:

- **Users without the *ncw_user* Web GUI role can view the Gauges page if they have the *ncw_gauges_viewer* Web GUI role.**
- **Users without the *ncw_user* Web GUI role, but with the *netcool_ro*, *ncw_gauges_viewer*, and *ncw_gauges_editor* Web GUI roles can edit the portlet preferences of the Gauges page.**

b. GUI pages, portlets, views, and console preference profiles can be accessed only by a user who is associated with a Web GUI role authorizing access to that element. In addition, for the following elements, with the exception of the administrative portlets under the Settings node, access levels are taken into account:

- **Portlets: If a role grants access at the "User" level, the user can view and interact with the portlet and access the portlet help. In addition, at the "Privileged User" level, a user can edit personal settings for that portlet. In addition, at the "Editor" level, a user and edit global settings for that portlet.**
- **Pages: If a role grants access at the "User" or "Privileged User" level, users can launch the node from the navigation. In addition, at the "Editor" level, users can edit the content and layout of the page.**

c. Access to the portlet preferences of the Event Dashboard portlet requires the *ncw_dashboard_editor* Web GUI role.

d. The following administrative actions can only be performed by users with the *ncw_admin* and *netcool_rw* Web GUI roles:

- **Definition of global and system filters and views**
- **Definition of tools and menus for the AEL, including User Preferences for a user**
- **Registration of CGI scripts**
- **Definition of maps and map resources**
- **Edit a Gauges page (only if the *ncw_gauges_viewer* Web GUI role has been assigned in addition)**

e. The following administrative actions can only be performed by users with the *adminsecuritymanager* and *Administrator* Web GUI roles:

- **Management of Web GUI roles (creation, deletion, and change of role properties)**

f. The following administrative actions can only be performed by users with the *iscadmins* Web GUI role:

- **user management**
- **Web GUI group management**
- **assignment of Web GUI roles to users and Web GUI groups**

2. If a user filter or restriction filter is assigned to a user, or to an ObjectServer group that the user is a member of, and the filter is configured to be applied to the requested database table, access is denied to those table rows that do not match the filter condition. Restriction filters take precedence over user filters.

3. A user may only access functions within the AEL as permitted by the User Preferences set for the user

4. The SQL workbench can only be executed by users with the *ncw_admin* Web GUI role that are also members of the *ISQL* ObjectServer group.

5. Web GUI groups can be used to restrict access to the following GUI elements based on whether users are members of a Web GUI group associated with that element or not:

a. maps

b. smartPages and templates

c. CGI scripts (if configured via the ACCESS_POLICY_RESTRICTED attribute)

d. tools in the Active Events List

6. Furthermore, a user can only execute a tool on an event in the Active Events List if:

a. when the AlertSecurityModel is enabled: the user is the owner or a member of the same ObjectServer group as the owner of the event; or

b. when the AlertSecurityModel is disabled: the user is member of the *Normal* ObjectServer group and the event is owned by the user or owned by the *nobody* user; or, the user is member of the *Administrator* ObjectServer group and the event is owned by the user, by the *nobody* user, or by a member of the *Normal* ObjectServer group.

7. CGI scripts can only be accessed if they are registered in the Web GUI configuration.

.

FDP_ACF.1.3-b The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **None**.

FDP_ACF.1.4-b The TSF shall explicitly deny access of subjects to objects based on the **following rules: None**.

6.1.4 Identification and authentication (FIA)

6.1.4.1 Authentication failure handling (FIA_AFL.1)

FIA_AFL.1.1 The TSF shall detect when **an administrator configurable positive integer within 1 and 10** unsuccessful authentication attempts occur related to **user authentication** .

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been **met**, the TSF shall **lock the user account**.

6.1.4.2 User attribute definition (FIA_ATD.1)

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

1. **user name (ID)**
2. **password**
3. **ObjectServer group membership**
4. **Web GUI group membership**
5. **restriction filters**
6. **user filters**
7. **account status**
8. **associations with Web GUI roles**
9. **the "PAM" attribute (determining whether a user is authenticated based on external authentication providers or not)**

.

6.1.4.3 Verification of secrets (FIA_SOS.1)

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet **an administrator-defined minimum number of:**

1. **8 or more overall characters;**
2. **1 or more numeric characters;**
3. **2 or more alphabetic characters; and**
4. **0 or more punctuation characters.**

.

6.1.4.4 User authentication before any action (FIA_UAU.2)

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

6.1.4.5 User identification before any action (FIA_UID.2)

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

6.1.4.6 User-subject binding (FIA_USB.1)

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

- **user name,**
- **ObjectServer group memberships,**
- **Web GUI group memberships,**
- **restriction filters,**
- **user filters,**
- **association with Web GUI roles**

.

FIA_USB.1.2 The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: **none** .

FIA_USB.1.3 The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: **changes shall be effective immediately, with the exception of:**

- **ObjectServer user lists, group membership, and restriction filters cached in the Web GUI - these will only be updated in an administrator-defined time interval or upon request of an administrator**
- **certain objects (tool pages) requested by users from the Web GUI, such as the Table View and TIP navigation tree, where changes will only become effective the next time the tool is requested from the user (but not while it is still loaded and available in the user's web browser)**

.

6.1.5 Security management (FMT)

6.1.5.1 Management of security attributes (FMT_MSA.1-a)

FMT_MSA.1.1-a The TSF shall enforce the **ObjectServer access control policy** to restrict the ability to **query , modify , delete** the security attributes

- **user names (IDs)**
- **ObjectServer group memberships**
- **passwords**
- **restriction filters**
- **permissions**
- **roles**

to **users that are associated with, or are members of a group that is associated with, the required permissions or roles**.

6.1.5.2 Management of security attributes (FMT_MSA.1-b)

FMT_MSA.1.1-b The TSF shall enforce the **Web GUI access control policy** to restrict the ability to **query , modify , delete** the security attributes

- **user names (IDs)**
- **ObjectServer group memberships**
- **Web GUI group memberships**
- **passwords**
- **restriction filters**
- **user filters**
- **Web GUI roles**

to **users that are associated with, or are members of a group that is associated with, the required permissions or roles**.

6.1.5.3 Static attribute initialisation (FMT_MSA.3-a)

FMT_MSA.3.1-a The TSF shall enforce the **ObjectServer access control policy** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2-a The TSF shall allow the **users that have been authorized in accordance with the ObjectServer access control policy** to specify alternative initial values to override the default values when an object or information is created.

6.1.5.4 Static attribute initialisation (FMT_MSA.3-b)

FMT_MSA.3.1-b The TSF shall enforce the **Web GUI access control policy** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2-b The TSF shall allow the **users that have been authorized in accordance with the Web GUI access control policy** to specify alternative initial values to override the default values when an object or information is created.

6.1.5.5 Specification of Management Functions (FMT_SMF.1)

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

- **user and group management, including password management,**
- **permissions and role management,**
- **management of ObjectServer and Web GUI groups,**
- **management of restriction and user filters,**
- **management of audit triggers**

.

6.1.5.6 Security roles (FMT_SMR.1)

FMT_SMR.1.1 The TSF shall maintain the roles

- ***SuperUser* role,**
- ***iscadmins* Web GUI role (Tivoli Integrated Portal administrators)**

- ***iscusers* Web GUI role (Tivoli Integrated Portal users)**
- ***adminsecuritymanager* Web GUI role**
- ***Administrator* Web GUI role**
- ***ncw_admin* Web GUI role (Web GUI administrators)**
- ***ncw_user* Web GUI role**
- ***netcool_ro* Web GUI role**
- ***netcool_rw* Web GUI role**
- ***ncw_dashboard_editor* Web GUI role**
- ***ncw_gauges_viewer* Web GUI role**
- ***ncw_gauges_editor* Web GUI role**
- **administrator-defined Web GUI roles**

.

FMT_SMR.1.2 The TSF shall be able to associate users *directly, or indirectly through groups,* with roles.

6.1.6 Protection of the TSF (FPT)

6.1.6.1 Failure with preservation of secure state (FPT_FLS.1)

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur: **unavailability of an ObjectServer** .

6.1.6.2 Basic internal TSF data transfer protection (FPT_ITT.1)

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

6.1.7 Resource utilisation (FRU)

6.1.7.1 Degraded fault tolerance (FRU_FLT.1)

FRU_FLT.1.1 The TSF shall ensure the operation of **event collection** when the following failures occur:

- **no ObjectServer is available to receive the events**
- **the primary ObjectServer in a failover configuration is not available**

.

6.2 Security Functional Requirements Rationale

6.2.1 Coverage

The following table provides a mapping of SFR to the security objectives, showing that each security functional requirement addresses at least one security objective.

Security Functional Requirements	Objectives
FAU_GEN.1	O.Auditing
FAU_SAR.1	O.Auditing
FAU_STG.1	O.Auditing
FCS_CKM.1	O.SecureComms
FCS_COP.1-a	O.SecureComms
FCS_COP.1-b	O.Password_Encryption, O.SecureComms
FCS_COP.1-c	O.SecureComms
FCS_COP.1-d	O.SecureComms
FCS_COP.1-e	O.SecureComms
FDP_ACC.1-a	O.Discretionary_Access
FDP_ACC.1-b	O.Discretionary_Access
FDP_ACF.1-a	O.Discretionary_Access
FDP_ACF.1-b	O.Discretionary_Access
FIA_AFL.1	O.Discretionary_Access
FIA_ATD.1	O.Discretionary_Access
FIA_SOS.1	O.Discretionary_Access
FIA_UAU.2	O.Discretionary_Access
FIA_UID.2	O.Discretionary_Access
FIA_USB.1	O.Discretionary_Access
FMT_MSA.1-a	O.Discretionary_Access
FMT_MSA.1-b	O.Discretionary_Access
FMT_MSA.3-a	O.Discretionary_Access
FMT_MSA.3-b	O.Discretionary_Access
FMT_SMF.1	O.Auditing, O.Discretionary_Access
FMT_SMR.1	O.Discretionary_Access
FPT_FLS.1	O.Failover
FPT_ITT.1	O.SecureComms

Security Functional Requirements	Objectives
FRU_FLT.1	O.Failover

Table 7: Mapping of security functional requirements to security objectives

6.2.2 Sufficiency

The following rationale provides justification for each security objective for the TOE, showing that the security functional requirements are suitable to meet and achieve the security objectives:

Security objectives	Rationale
O.Auditing	The objective to provide means to audit changes to configuration data and other security-relevant actions is met by requirement for audit record generation (FAU_GEN.1). Administrators have the ability to review audit data (FAU_SAR.1), which is protected by the TOE against unauthorized access (FAU_STG.1). Supportive management functionality is called out in FMT_SMF.1.
O.Discretionary_Access	The objective to allow the restriction of access to managed objects is implemented by two discretionary access control policies, one enforced by the ObjectServer and one by the Web GUI, as specified in FDP_ACC.1-a, FDP_ACC.1-b, FDP_ACF.1-a, and FDP_ACF.1-b. On the ObjectServer, this is supported by identification (FIA_UID.2) and authentication (FIA_UAU.2) of users, including the enforcement of authentication failure policies (FIA_AFL.1) and a password policy (FIA_SOS.1), and user-subject binding (FIA_USB.1). Relevant user attributes are identified in FIA_ATD.1. Access control and authentication are supported by requirements pertaining to the management of the access control enforcement (FMT_MSA.1-a, FMT_MSA.1-b, FMT_MSA.3-a. FMT_MSA.3-b, FMT_SMF.1, and FMT_SMR.1).
O.Password_Encryption	FCS_COP.1-b implements the encryption mechanism used to encrypt configuration data.
O.Failover	The functionality of a backup ObjectServer in a failover configuration is defined in FRU_FLT.1 and FPT_FLS.1.
O.SecureComms	FPT_ITT.1 spells out the requirement to protect information that is being transmitted between TOE parts, while the cryptographic mechanisms used to implement TLS channels, as well as AES encryption and SHA-1 HMACs for SNMP v3 messages, are defined in FCS_CKM.1, FCS_COP.1-a, FCS_COP.1-b, FCS_COP.1-c, FCS_COP.1-d, and FCS_COP.1-e.

Table 8: Security objectives for the TOE rationale

6.2.3 Security Requirements Dependency Analysis

Dependencies within the EAL4 package selected for the security assurance requirements have been considered by the authors of CC Part 3 and are not analyzed here again. The included component on flaw remediation, ALC_FLR.3, has no dependencies on other requirements.

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

The following table demonstrates the dependencies of SFRs modeled in CC Part 2 and how the SFRs for the TOE resolve those dependencies:

Security Functional Requirement	Dependencies	Resolution
FAU_GEN.1	FPT_STM.1	The dependency FPT_STM.1 of FAU_GEN.1 has not been resolved. The TOE relies on the underlying runtime environment to provide a reliable time source for the generation of audit records.
FAU_SAR.1	FAU_GEN.1	FAU_GEN.1
FAU_STG.1	FAU_GEN.1	FAU_GEN.1
FCS_CKM.1	[FCS_CKM.2 or FCS_COP.1]	FCS_COP.1-b FCS_COP.1-c
	FCS_CKM.4	Session keys are not destroyed explicitly. They expire when an TLS session expires.
FCS_COP.1-a	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1
	FCS_CKM.4	Session keys are not destroyed explicitly. They expire when an TLS session expires.
FCS_COP.1-b	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	Keys for the encryption of configuration properties are generated in the environment, using Java's secure random number interface.
	FCS_CKM.4	Keys are stored persistently and not deleted explicitly.
FCS_COP.1-c	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	This SFR itself specifies the generation of digests.
	FCS_CKM.4	Digests are not destroyed explicitly.

Security Functional Requirement	Dependencies	Resolution
FCS_COP.1-d	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	Certificates are generated in the operational environment.
	FCS_CKM.4	There is no requirement to destroy certificates.
FCS_COP.1-e	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	No key needs to be generated for the HMAC. Rather, user-supplied passwords are used.
	FCS_CKM.4	There is no need to destroy HMACs.
FDP_ACC.1-a	FDP_ACF.1	FDP_ACF.1-a
FDP_ACC.1-b	FDP_ACF.1	FDP_ACF.1-b
FDP_ACF.1-a	FDP_ACC.1	FDP_ACC.1-a
	FMT_MSA.3	FMT_MSA.3-a
FDP_ACF.1-b	FDP_ACC.1	FDP_ACC.1-b
	FMT_MSA.3	FMT_MSA.3-b
FIA_AFL.1	FIA_UAU.1	FIA_UAU.2
FIA_ATD.1	No dependencies.	
FIA_SOS.1	No dependencies.	
FIA_UAU.2	FIA_UID.1	FIA_UID.2
FIA_UID.2	No dependencies.	
FIA_USB.1	FIA_ATD.1	FIA_ATD.1
FMT_MSA.1-a	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1-a
	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MSA.1-b	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1-b
	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MSA.3-a	FMT_MSA.1	FMT_MSA.1-a
	FMT_SMR.1	FMT_SMR.1

Security Functional Requirement	Dependencies	Resolution
FMT_MSA.3-b	FMT_MSA.1	FMT_MSA.1-b
	FMT_SMR.1	FMT_SMR.1
FMT_SMF.1	No dependencies.	
FMT_SMR.1	FIA_UID.1	FIA_UID.2
FPT_FLS.1	No dependencies.	
FPT_ITT.1	No dependencies.	
FRU_FLT.1	FPT_FLS.1	FPT_FLS.1

Table 9: TOE SFR dependency analysis

6.3 Security Assurance Requirements

The security assurance requirements for the TOE are the Evaluation Assurance Level 4 components as specified in [CC] part 3, augmented by ALC_FLR.3.

The following table shows the Security assurance requirements, and the operations performed on the components according to CC part 3: iteration (Iter.), refinement (Ref.), assignment (Ass.) and selection (Sel.).

Security assurance class	Security assurance requirement	Source	Operations			
			Iter.	Ref.	Ass.	Sel.
ADV Development	ADV_ARC.1 Security architecture description	CC Part 3	No	No	No	No
	ADV_FSP.4 Complete functional specification	CC Part 3	No	No	No	No
	ADV_IMP.1 Implementation representation of the TSF	CC Part 3	No	No	No	No
	ADV_TDS.3 Basic modular design	CC Part 3	No	No	No	No
AGD Guidance documents	AGD_OPE.1 Operational user guidance	CC Part 3	No	No	No	No
	AGD_PRE.1 Preparative procedures	CC Part 3	No	No	No	No
ALC Life-cycle support	ALC_CMC.4 Production support, acceptance procedures and automation	CC Part 3	No	No	No	No
	ALC_CMS.4 Problem tracking CM coverage	CC Part 3	No	No	No	No
	ALC_DEL.1 Delivery procedures	CC Part 3	No	No	No	No
	ALC_DVS.1 Identification of security measures	CC Part 3	No	No	No	No
	ALC_FLR.3 Systematic flaw remediation	CC Part 3	No	No	No	No

Security assurance class	Security assurance requirement	Source	Operations			
			Iter.	Ref.	Ass.	Sel.
	ALC_LCD.1 Developer defined life-cycle model	CC Part 3	No	No	No	No
	ALC_TAT.1 Well-defined development tools	CC Part 3	No	No	No	No
ASE Security Target evaluation	ASE_INT.1 ST introduction	CC Part 3	No	No	No	No
	ASE_CCL.1 Conformance claims	CC Part 3	No	No	No	No
	ASE_SPD.1 Security problem definition	CC Part 3	No	No	No	No
	ASE_OBJ.2 Security objectives	CC Part 3	No	No	No	No
	ASE_ECD.1 Extended components definition	CC Part 3	No	No	No	No
	ASE_REQ.2 Derived security requirements	CC Part 3	No	No	No	No
	ASE_TSS.1 TOE summary specification	CC Part 3	No	No	No	No
ATE Tests	ATE_COV.2 Analysis of coverage	CC Part 3	No	No	No	No
	ATE_DPT.1 Testing: basic design	CC Part 3	No	No	No	No
	ATE_FUN.1 Functional testing	CC Part 3	No	No	No	No
	ATE_IND.2 Independent testing - sample	CC Part 3	No	No	No	No
AVA Vulnerability assessment	AVA_VAN.3 Focused vulnerability analysis	CC Part 3	No	No	No	No

Table 10: Security assurance requirements

6.4 Security Assurance Requirements Rationale

The evaluation assurance level has been chosen commensurate with the threat environment that is experienced by typical consumers of the TOE.

7 TOE Summary Specification

7.1 TOE Security Functionality

7.1.1 Identification and authentication

The ObjectServer identifies the users that connect to it (both human users and distributed TOE components) by a unique user name (FIA_UID.2) and associates them with relevant security attributes (FIA_USB.1). The ObjectServer also enforces authentication (FIA_UAU.2). The way authentication decisions are derived differs depending on the configuration of the server:

- If the ObjectServer's internal authentication mechanism is used, the ObjectServer will compare the password sent by a client to the corresponding user entry (FIA_ATD.1) in the ObjectServer's user registry (database).
- If the ObjectServer (on UNIX) is configured to use PAM for authentication, the authentication decision to be enforced will be provided by an external PAM module.
- If the ObjectServer is configured to use an external LDAP server for authentication, the LDAP server (which may be Active Directory on Windows) will provide the authentication decision.

If external authentication providers (PAM, LDAP) are configured as authentication providers for an ObjectServer, they will only be used for the individual users in the ObjectServer's registry for which external authentication has been enabled. Other users will still be authenticated by the internal authentication mechanism.

If the RestrictPasswords property for an ObjectServer is set to true, the ObjectServer will enforce a password policy for the definition of new user passwords defined in the PasswordFormat property conformant with FIA_SOS.1. This applies only for passwords stored in the ObjectServer's user registry.

The ObjectServer maintains in its user registry (FIA_ATD.1) the status of users (active: yes/no). Users who are not active are disabled and cannot access the TOE until an administrator resets their status to active. A trigger (disable_user) is provided that can be used to disable users after a configured number of failed logon attempts (the default is 5) under that user name (FIA_AFL.1). This is enforced regardless of whether or not external authentication providers are used for authentication.

When users connect to ObjectServer Gateways or Process Agents, these TOE components merely enforce authentication decisions (FIA_UAU.2) provided to them by their runtime environment. In the evaluated configuration, they derive authentication decisions either via PAM (on UNIX) or the native operating system authentication mechanism (on Windows). They will reject connection requests if the authentication provider in the environment does not confirm that a user has been succesfully authenticated.

ObjectServer Gateways and Process Agents do not implement user identification - when connecting to the ObjectServer, the ObjectServer considers the Gateway or Process Agent its user.

The Web GUI relies on its underlying runtime environment for the authentication of users. It is, however, able to associate users connecting to the Web GUI with their user name based on the identity provided to it by the runtime environment (FIA_USB.1).

An additional form of authentication is used througout the distributed TOE components when communicating via TLS: Clients will authenticate the servers they connect to by verifying the provided server certificate. This is further described below.

7.1.2 Discretionary Access Control

The TOE has two distinct access control policies: One enforced by the ObjectServer for users accessing data held in its database, and one by the Web GUI for users of the graphical user interface.

7.1.2.1 ObjectServer

ObjectServers implement the ObjectServer access control policy required in FDP_ACC.1-a. As a consequence, access to the objects stored in the ObjectServer is limited based on the permissions given to users for accessing these objects. Since the ObjectServer is a database and holds almost all TSF data (except for some encryption keys, see below, and configuration data), this access control mechanism is used to restrict access to user data (collected events) and TSF data (such as the security attributes defined in FMT_MSA.1-a) alike.

Permissions are assigned to objects and define which operation is granted on the object. Multiple permissions can be bundled into roles. Permissions and/or roles can be associated with users. Users on the other hand can be bundled into groups, and permissions and/or roles can be assigned to groups (FIA_ATD.1) as well. The most complex access control attribution is therefore a quadruple of {user - group - role - permission}, while the easiest attribution is a direct assignment of permissions to a user. The access control rules that are enforced are spelled out in FDP_ACF.1-a. When users are first created, they are provided with a minimal set of permissions, unless specified otherwise by an administrator at that time (FMT_MSA.3-a).

In addition to permissions that are assigned to objects, commonly referred to as object permissions, the ObjectServer also enforces system permissions. System permissions, such as ALTER SYSTEM SET PROPERTY, allow users to execute certain commands that can be run in the ObjectServer. While system permissions are, from a technical point of view, not associated with a discrete object in the ObjectServer database, but rather with a specific command made available by the ObjectServer, the access control mechanism described above is otherwise the same both for system and object permissions.

Note that FMT_SMR.1 explicitly distinguishes between users and users that are associated with the SuperUser role. This distinction is being made because of the special notion of the SuperUser role - this role is not directly assigned to any object, but indirectly assigned to all objects that are subject to access control enforcement. Users that are associated with this role (directly or through group membership) have access to all these objects (FDP_ACF.1-a). Apart from this, the ObjectServer implements the concept of "roles" envisioned in FMT_SMR.1 through the combination of flexible, administrator-defined associations between users or groups and permissions or roles as defined above. Pre-defined roles and groups other than the SuperUser are available, but can be modified by administrators.

Note, furthermore, that the groups and roles defined in the ObjectServer are not to be confused with Web GUI groups and Web GUI roles (as described below).

In addition to the rules described above, the ObjectServer access control policy also implements the concept of restriction filters (FDP_ACF.1-a), allowing administrators to further filter access to objects in the database for individual users or groups (FIA_ATD.1). Essentially, this is implemented by adding an administrator-defined SQL 'WHERE ...' clause to every request that is issued by the user or group associated with the filter. When several restriction filters apply to one user, the user can access the intersection of the events in all filters, unless the "RestrictionFilterAND" property in the ObjectServer is set to "FALSE", in which case the user can access the union of the events in all filters.

7.1.2.2 Web GUI

The Web GUI offers access to objects stored in the ObjectServer database, pre-dominantly through the Active Events List page served to users. However, the Web GUI also offers various other graphical interfaces that require an access control policy more focused on the capabilities of a highly configurable graphical user interface (as called out in FDP_ACC.1-b and defined in FDP_ACF.1-b). The security attributes used by the Web GUI access control policy are defined in FMT_MSA.1-b. When users are first created, they are provided with a minimal set of permissions, unless specified otherwise by an administrator at that time (FMT_MSA.3-b).

The Web GUI connects to the ObjectServer as an administrator-defined user, typically the "root" user, which in the ObjectServer is associated with privileges to fully (read, write, modify) access all event data. Upon establishment of this connection, the Web GUI will retrieve the existing restriction filters and lists of configured groups and users from the ObjectServer and store them in a local cache that is synchronized in an administrator-defined interval (by default: hourly), or on demand if requested by an administrator using the osresync utility. If a user generates a request for data stored in an ObjectServer through the Web GUI, the ObjectServer will obviously not be able to enforce the {user - group - role - permission} association defined in the ObjectServer for that user, since the Web GUI is not connecting under that user's ID to the ObjectServer. Rather, the Web GUI itself will apply any restriction filters for that particluar user when generating a request to the ObjectServer, as well as Web GUI-specific user filters that can be defined and assigned to users by administrators. Restriction filters take precedence over user filters.

Before a user can cause the Web GUI to generate a request for data to an ObjectServer, the user will have to have access to specific GUI pages. Access to these pages is determined based on the Web GUI role membership of users. Within a GUI page, access to individual elements (such as, the Active Events List) can be further controlled by means of those Web GUI roles. (For example, a user may have access to a certain GUI page, but not to all portlets in that GUI page.) Netcool/OMNIbus comes with some pre-defined roles (relevant ones spelled out in FMT_SMR.1) and allows administrators to define further roles.

A combination of Web GUI roles and the user groups defined in the ObjectServer are applied by the Web GUI when determining access to the SQL workbench. This page can only be accessed and used by users who have both the *ncw_admin* Web GUI role and are members of the *ISQL* ObjectServer group.

In addition to roles, the Web GUI access control policy also offers to group users into Web GUI groups. These can be used to restrict access to members of particular Web GUI groups for:

- Maps
- SmartPages
- templates
- CGI scripts
- tools in the Active Events List

For the execution of tools on events in the Active Events List (AEL), the TOE - in addition to enforcing access based on Web GUI group membership as describe above - also defines a configuration option referred to as AlertSecurityModel, which further determines whether a user is allowed to execute a particular tool on a particular event as specified in FDP_ACF.1.2-b.

Further, the following User Preferences that can be configured for the AEL are enforced by the TOE:

- Allow filter and view selection
- Allow filter builder access

- Allow view builder access
- Allow preference configuration
- Allow event selection
- Show basic event information
- Show event details
- Show journals
- Edit journals (read write role)

(Note that there are other User Preferences, such as "User's home-page", that merely define preferences and do not contribute to access control as defined in this ST.)

If ObjectServer and Web GUI use disjunct user registries, administrators need to take care of assigning the same user IDs in both registries in order for the above described control mechanisms to work. There may be cases, however, where this is not required by an organization: If a Web GUI user does not have a corresponding user ID defined in the ObjectServer, the Web GUI will only provide read-only access to events stored in the ObjectServer database to that user.

Lastly, CGI scripts can only be accessed by users if an administrator has registered them in the Web GUI configuration.

7.1.3 Communications security

The TOE provides a secure communication channel for its distributed components based on TLS (FPT_ITT.1). Only FIPS-compliant cipher suites are used in the evaluated configuration:

- for TLS:
  - SSL_RSA_FIPS_WITH_3DES_EDE_CBC_SHA = { 0xFE,0xFF }
  - TLS_RSA_WITH_3DES_EDE_CBC_SHA = { 0x00,0x0A }
  - TLS_RSA_WITH_AES_128_CBC_SHA = { 0x00, 0x2F }
  - TLS_RSA_WITH_AES_256_CBC_SHA = { 0x00, 0x35 }

The related SFRs are FCS_CKM.1 and FCS_COP.1-a for the generation of symmetric keys during session establishment, FCS_COP.1-b for the en-/decryption of payload in sessions, and FCS_COP.1-c for generating and verifying SHA-1 hashes.

As mentioned above, during TLS session establishment the client will verify the server certificate presented to it (FCS_COP.1-d) in order to authenticate the server.

The Web GUI makes use of the TLS facilities provided by its runtime environment. The only SFR applicable in this context to the Web GUI is FDP_ITT.1 for invoking the WAS-provided communication channels.

The TOE also implements a secure form of SNMP v3 messages by providing AES encryption (FCS_COP.1-b) and SHA-1 HMACs (FCS_COP.1-e) for these messages.

7.1.4 Property encryption

The TOE provides encryption of authentication credentials and configuration properties based on AES (FCS_COP.1-b) in various occasions:

- Property value encryption on ObjectServers, Probes, and Gateways. The nco_aes_crypt command-line utility can be used to encrypt data stored in configuration files, including the authentication credentials sent to an ObjectServer by Probes and Gateways. A key

(random number) can be generated using the nco_keygen utility (the ST does not make any claims on the quality of the resulting key), which is stored in a flat file in the operating system's file system.

- TLS password encryption on Web GUI servers. The ncw_fips_crypt command-line utility can be used to encrypt passwords stored on Web GUI servers.
- Password encryption in ObjectServers. The ObjectServer encrypts passwords stored in its database if the PasswordEncryption property is set to AES.

7.1.5 Fault tolerance

Tivoli Netcool/OMNIbus allows the setup of multiple ObjectServers in a primary/backup failover configuration, so that Probes can deliver event information to a backup ObjectServer in case the primary one is not available (FRU_FLT.1, FPT_FLS.1). User data will be re-synchronized between the primary and backup ObjectServer when both of them are available again - by design, users are allowed to make changes to objects in the ObjectServer database during re-synchronization, which in rare circumstances might lead to inconsistencies in the primary and backup ObjectServer's databases. The operational environments that the TOE is used in require that the availability of and ability to process events takes precedence over absolute consistency between the databases.

The TOE supports automated failover and fallback, where instances that communicate with an ObjectServer, such as probes, can be configured with the address of a backup ObjectServer to connect to in case the primary ObjectServer is unavailable.

Both in primary/backup mode and when operating only with a single ObjectServer, the following TOE parts can use local store + forward files to queue events until an ObjectServer becomes available again (FRU_FLT.1, FPT_FLS.1).

7.1.6 Auditing

The ObjectServer employs two mechanisms to generate audit records (FAU_GEN.1):

1. Database auditing.

   Audit records are created by triggers that are provided with the TOE. Administrator can enable these triggers using the Administrator client or by issuing the following SQL statement:

   ```
   alter trigger group audit_config set enabled true;
   ```

   Enabling the group of triggers, as well as disabling it with the corresponding statements, will create an audit record.

   Additional events that can be audited by triggers include the creation (create), modification (alter), and deletion (drop) of:

   - all database objects (including, for example, the contents of the user registry),
   - ObjectServer properties (configuration values), including the disabling of the audit trigger group,
   - various information stored in the ObjectServer database and made available to users of the Administrator client, such as SQL tools and associated prompts, definition of menus and column visuals,
   - event classes and conversions;

   as well as denied permission requests (permission_denied).

   The information recorded for each event includes (amongst others):

- date and time,
- a unique identifier,
- the originator (user identity) of the event,
- the event type,
- the target object
- first and last occurence of the event,
- severity of the event

The records generated here always represent events that succeeded, resulting in the failure/success not being explicitly recorded.

Records are stored in the alerts.status table of the ObjectServer database, and can be reviewed by administrators that are authorized to access the information in this table (FAU_SAR.1) by using the TOE-provided means, i.e. the SQL Interactive Interface and the Web GUI (FAU_STG.1).

2. Log-file auditing.

   Audit records are generated in the ObjectServer code for certain events, and are categorized into the following, hierarchical levels: debug, info, warn, error, fatal. The log level that an ObjectServer runs under can be configured, and only messages greater than or equal to the configured level are logged.

   The events that can be recorded include:

Event type	Outcome	Level	Information recorded
Authentication (user lookup)	lookup failure	error	authentication failure for user user_name : error_message
Authentication (user lookup)	user disabled	error	User 'user_name' disabled
ObjectServer authentication	failure	error	authentication failure - cannot authenticate user \"user\" : reason
ObjectServer authentication	success	info	authentication success - session opened for user \"user\" : success
authorization	failure	error	authorisation failure for user user_name : app_id = (application id, constant 1) object_id = (id for object on which permission is being checked) string = (name of object) : reason
authorization	success	info	authorisation success for user user_name : app_id = (application id, constant 1) object_id =

Event type	Outcome	Level	Information recorded
			(id for object on which permission is being checked) string = (name of object)

Authorization results are not normally logged for performance reasons. They can be enabled by setting an environment variable.

Records are written to a flat file in the underlying operating systems, and means provided in the operational environment can be used to review and search audit data. The underlying operating system is responsible for the protection of the log file.

As far as audit record generation by the Web GUI is concerned:

The Web GUI writes audit records for its security functions to the ncw.N.log file, using an eWAS-provided logging framework. The Web GUI component of the TOE does not offer any facilities to review these records.

7.1.7 Management

The TOE offers a number of management interfaces to manage the TSF (FMT_SMF.1) as described above, including:

- The SQL interactive interface (started with nco_sql on UNIX systems, isql on Windows systems) provides users with a command line SQL client for ObjectServers.
- The Netcool/OMNIbus Administrator (started with nco_config on UNIX, nco_config.vbs on Windows), a GUI client for the management and configuration of ObjectServers and Process Agents, including thre review of audit triggers.
- The Web GUI, a web-based GUI for administration of ObjectServers.
- The Server Editor (nco_xigen on UNIX, omnictl on Windows) can be used to maintain communication information for the TOE components.
- The Properties Editor provides support for editing properties files.

8 Abbreviations, Terminology and References

8.1 Abbreviations

AD
Active Directory

AEL
Active Event List, a GUI provided by Web GUIbtop

AEN
Accelerated Event Notification

AES
Advanced Encryption Mechanism

Ass.
Assignment

CA
Certificate Authority

CC
Common Criteria

GSKit
Global Security Kit, a part of the TOE

ICC
IBM Crypto for C

Iter.
Iteration

JDBC
Java Database Client

LDAP
Lightweight Directory Access Protocol

PAM
Pluggable Authentication Module

PKI
Public Key Infrastructure

Ref.
Refinement

Sel.
Selection

SQL
Structured Query Language

TIP
Tivoli Integrated Portal, a part of the TOE.

TLS
Transaction Layer Security

WAS
WebSphere Application Server

8.2 Terminology

This section contains definitions of technical terms that are used with a meaning specific to this document. Terms defined in the [CC] are not reiterated here, unless stated otherwise.

administrator
The TOE guidance uses the terms "administrator" and "operator" to distinguish between users who are tasked with installing, configuring and maintaining the product (administrators), and users who actually employ the product's event management functionality (operators). In the same spirit, the term "administrator" is used in this ST occasionally to reference users of the TOE who are using the provided administrative interfaces, such as the SQL Interactive Interface, and have been granted one or more permissions that relate to the administration of the TOE or the management of users data hosted by the TOE (such as user management, definition of access control rules, etc.). There is no formal distinction between "administrators", "operators", and "users" in this Security Target, though - all users have more or less administrative rights, depending on the roles or privileges assigned to them. The only specific role besides users with varying degrees of administrative access is the SuperUser.

operator
See administrator.

root
The root user is a pre-defined user in the ObjectServer that has the SuperUser role.

SuperUser
The SuperUser is a user that has been associated with the SuperUser role, either directly or through group membership. The SuperUser is not subject to any access control restrictions.

user
Humans or machines interacting with the TOE via the provided user and programmatic interfaces. See also "Administrator".

8.3 References

CC	**Common Criteria for Information Technology Security Evaluation**	
	Version	3.1R3
	Date	July 2009
	Location	http://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R3.pdf
	Location	http://www.commoncriteriaportal.org/files/ccfiles/CCPART2V3.1R3.pdf
	Location	http://www.commoncriteriaportal.org/files/ccfiles/CCPART3V3.1R3.pdf
FIPS140-2	**FIPS PUB 140-2: SECURITY REQUIREMENTS FOR CRYPTOGRAPHIC MODULES.**	
	Date	Issued May 25, 2001, including CHANGE NOTICES (12-03-2002)

FIPS180-2 **FIPS PUB 180-2: Specification for the SECURE HASH STANDARD, including Change Notice to include SHA-224**
Date August 1, 2002

FIPS197 **FIPS PUB 197: Specification for the ADVANCED ENCRYPTION STANDARD (AES).**
Date November 26, 2001

FIPS46-3 **FIPS PUB 46-3: DATA ENCRYPTION STANDARD (DES).**
Date October 25, 1999

FIPS81 **FIPS PUB 81: DES MODES OF OPERATION.**
Date Issued December 2, 1980, including CHANGE NOTICES 2 and 3

RFC2313 **RFC 2313: PKCS#1: RSA Cryptography Specification, Version 1.5.**
Date March 1998

RFC2437 **RFC 2437: PKCS #1: RSA Cryptography Specifications, Version 2.0.**
Date October 1998

SP800-135 **NIST Special Publication 800-135: Recommendation for Existing Application-Specific Key Derivation Functions**
Date December 2010

TLS_AES **P. Chown: Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS); RFC 3268.**
Date June 2002

TLSv1 **T. Dierks, C.Allen: The TLS Protocol Version 1.0; RFC 2246.**
Date January 1999

TLSv1.1 **T. Dierks, E. Rescorla: The Transport Layer Security (TLS) Protocol Version 1.1; RFC 4346.**
Date April 2006