MetaMatrix™ Enterprise Release 5.5.3

Security Target

Version 1.5

July 10, 2009,

Prepared For

RedHat, Incorporated

Prepared By

Suite 5200♦7925 Jones Branch Drive♦McLean, VA 22102-3321♦703 848-0883♦Fax 703 848-0960

TABLE OF CONTENTS

Table of Tables and Figures

Table / Figure **Page**

1 Security Target Introduction

1.1 Security Target Reference

ST Title: MetaMatrix Enterprise Release 5.5.3 Security Target

ST Version: Version 1.5

ST Date: 7/10/2009

ST Author: CygnaCom Solutions

1.1.1 References

Table 1-1 provides the references used to develop this Security Target.

Table 1-1: References

Reference Title	ID
MetaMatrix Enterprise Administration Guide, MetaMatrix Products, Release 5.5.3	[ADMIN]
Common Criteria for Information Technology Security Evaluation, CCMB-2006-09-001, Version 3.1, Revision 2, September 2007.	[CC]
MetaMatrix Enterprise Console User's Guide, MetaMatrix Products, Release 5.5.3	[CONSOLE]
MetaMatrix Enterprise Designer User's Guide, MetaMatrix Products, Release 5.5.3	[DESIGN]
MetaMatrix Feature Overview and Value Proposition, MetaMatrix Products, Release 5.5.3	[FEATURES]
MetaMatrix Enterprise Installation Guide, MetaMatrix Products, Release 5.5.3	[INSTALL]
MetaMatrix Known Issues, MetaMatrix Products, Release 5.5.3	[ISSUES]
MetaMatrix Administration Shell Users Guide, MetaMatrix Products, Release 5.5.3	[MMAdmin]
MetaMatrix Metadata Repository User Guide, MetaMatrix Products, Release 5.5.3	[MMR]
MetaMatrix Enterprise QueryBuilder User's Guide, MetaMatrix Products, Release 5.5.3	[QUERY]
MetaMatrix Enterprise 5.5.3 – README, MetaMatrix Enterprise Server, Release 5.5.3	[README]
MetaMatrix Release Notes, MetaMatrix Products, Release 5.5.3	[RELEASE]
MetaMatrix Server Security, User Authentication, and Authorization, MetaMatrix Products, Release 5.5.3	[SECURITY]
MetaMatrix Enterprise SSL Guide, MetaMatrix Products, Release 5.5.3, Rev A	[SSL]
MetaMatrix Enterprise Server Tuning Guide, MetaMatrix Products, Release 5.5.3	[TUNING]

1.2 TOE Reference

TOE Identification: MetaMatrix Enterprise Release 5.5.3

The r553_090507_0021.jar[1] patch must be applied to the MetaMatrix server system.

TOE Vendor: RedHat, Incorporated

[1] The patch can be downloaded from http://ftp.redhat.com/pub/redhat/metamatrix/CCC_Configuration/

1.3 TOE Overview

MetaMatrix Enterprise Release 5.5.3 (MetaMatrix) is an enterprise information integration (EII) system. EII is based on the premise that enterprises have a variety of information sources and information types, distributed geographically, and owned by different parts of the enterprise. A basic tenet of EII is that information should be capable of integration regardless of its native physical storage characteristics.

MetaMatrix manages and describes information that is spread across disparate enterprise information systems. Using MetaMatrix these enterprise information systems can be integrated into a single, complete data access solution. It provides a way to define the characteristics of information and how information is related, and manage this "data about data", or "Metadata". MetaMatrix users can issue queries to any data source, process and integrate the results derived from multiple sources.

MetaMatrix protects the distributed data and Metadata through an access control policy, user identification and authentication, role-based management functions and auditing of security relevant events.

This Security Target (ST) defines the Information Technology (IT) security requirements for MetaMatrix Enterprise Release 5.5.3. The TOE is being evaluated at assurance level EAL2.

1.3.1 TOE Type

MetaMatrix Enterprise Release 5.5.3 is an enterprise information integration (EII) system.

1.3.2 Hardware/Firmware/Software Required by the TOE

- Server running Red Hat Enterprise Linux 5 to host the following product components:
  - MetaMatrix Server
  - MetaMatrix Metadata Repository
  - MetaMatrix Platform
  - MetaMatrix Web Services
  - Connectors
- Workstation running Windows XP to host the following product components:
  - MetaMatrix Enterprise Designer
  - MetaMatrix Enterprise Console
  - MetaMatrix QueryBuilder (running inside Internet Explorer)

- SSL Implementation (MetaMatrix uses the JSSE implementation in the Java Runtime Environment provided by the product installer, which is the Sun JRE version 1.5.0.11)
- Tomcat/Apache Web Services (version 5.0.25 is installed by the MetaMatrix product installer)
- Relational Database to implement MetaMatrix Metadata Repository:
  - Oracle 11g;

- JDBC Database Drivers
    - DataDirect Connect for JDBC version 3.7
- LDAP Server
    - Red Hat Directory Server 8

1.4 TOE Description

1.4.1 Acronyms

Table 1-2 and Table 1-3 define product specific and CC specific acronyms respectively.

Table 1-2: Product Specific Acronyms

Acronym	Definition
ACI	Access Control Item
API	Application Programming Interface
CDK	Connector Development Kit
CLI	Command Line Interface
DBMS	Data Base Management System
EII	Enterprise Information Integration
EIS	Enterprise Information Systems
GUI	Graphical User Interface
JDBC	Java Database Connectivity
HTTP	HyperText Transfer Protocol
MMR	MetaMatrix Metadata Repository
ODBC	Open Database Connectivity
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
UML	Unified Modeling Language
VDB	Virtual Database
XA	eXtended Architecture
XML	Extensible Markup Language

Table 1-3: CC Specific Acronyms

Acronym	Definition
CC	Common Criteria [for IT Security Evaluation]
EAL	Evaluation Assurance Level
IT	Information Technology
OSP	Organizational Security Policy
PP	Protection Profile
SAR	Security Assurance Requirement
SFP	Security Function Policy
SFR	Security Functional Requirement
ST	Security Target
TOE	Target of Evaluation
TSC	TSF Scope of Control
TSF	TOE Security Functions
TSFI	TOE Security Functions Interface
TSP	TOE Security Policy

1.4.2 Terminology

Table 1-4 and Table 1-5 define product-specific and CC-specific terminology respectively.

Table 1-4: Product-Specific Terminology

Term	Definition
Connector	A Connector represents a set of Java classes, including a Connector Connection and Connector Translator that handle the communications between the MetaMatrix Server and the enterprise information system.
Entitlement	A named set of access rights which control which data constructs, such as tables or columns, a user account can create, read, update, and / or delete.
Identity (ID)	A representation (e.g., a string) uniquely identifying an authorized user, which can either be the full or abbreviated name of that user or a pseudonym.
Membership Domain Providers	The third-party applications used to store and manage information about users.
Metabase	The previous name for the MetaMatrix Repository
Metadata	Metadata abstracts information from the database itself and becomes useful to describe the content of the enterprise information systems and to determine how a column in one enterprise information source relates to another, and how those two columns could be used together for a new purpose. A piece of Metadata, called a meta object in the MetaMatrix Designer, contains information about a specific information structure, irrespective of whatever individual data fields that may comprise that structure. Metadata is data about data.
MetaModeler	The previous name for the MetaMatrix Enterprise Designer
Virtual Model	An abstract view of the data sources
Virtual Database	A Virtual Database (VDB) is an abstraction that allows the user to treat the separate data sources that have been modeled and integrated within it, as a single ODBC data source. A VDB consists of models, categories, groups and elements.

Table 1-5: CC-Specific Terminology

Term	Definition
Authorized User	A user who may, in accordance with the TSP, perform an operation.
External IT Entity	Any IT product or system, untrusted or trusted, outside of the TOE that interacts with the TOE.
Role	A predefined set of rules establishing the allowed interactions between a user and the TOE.
TOE Security Functions (TSF)	A set consisting of all hardware, software, and firmware of the TOE that must be relied upon for the correct enforcement of the TSP.
User	Any entity (human user or external IT entity) outside the TOE that interacts with the TOE.

1.4.3 Product Description

MetaMatrix Enterprise Release 5.5.3 (MetaMatrix) is a software-only product whose components are shown in Figure 1 below.

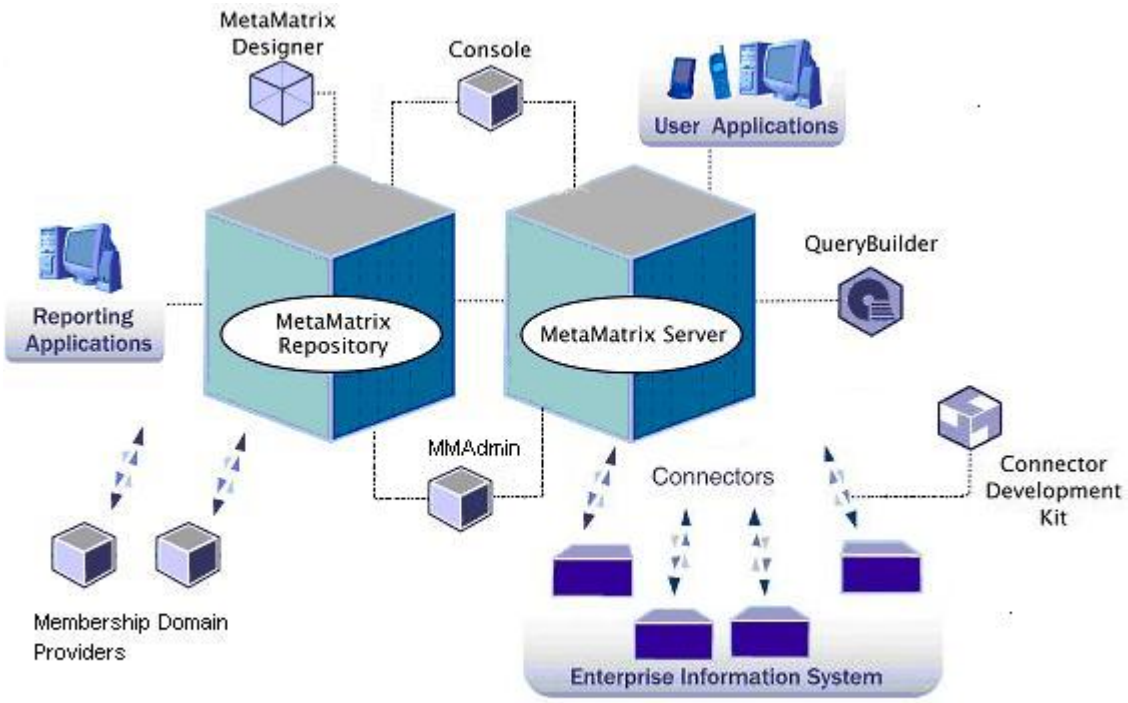

Figure 1: MetaMatrix Enterprise Release 5.5.3

MetaMatrix is an enterprise information integration (EII) system that manages and describes information across disparate enterprise information systems (EISs).

MetaMatrix has a federated data system that provides uniform access to all enterprise data sources using a variety of APIs. Information is accessed through the same standard APIs, regardless of whether the information is obtained from a single source or is consolidated from many sources, and regardless of whether the sources natively support the APIs. MetaMatrix provides access to information via SQL (or XQuery for XML), over JDBC, ODBC, SOAP/HTTP, or SOAP/JMS.

MetaMatrix enables end user applications to process queries that select (and update) data from one or more enterprise information sources, regardless of the native physical data storage method used by each enterprise information system. This means that a single query can access, reference, and return results from multiple integrated data sources.

For example a bank can gather data from its branches even through they may be using different operating systems on their data platforms. Or a local government can integrate data for law enforcement or accounting purposes from agencies that store data in various formats on differing databases or even flat files.

Within MetaMatrix, the design-time components (including the MetaMatrix Enterprise Designer, and the MetaMatrix Repository), enable users to create and manage Metadata Models: representations describing the nature and content of enterprise information systems.

Once captured, this Metadata can be searched, analyzed, and applied by applications throughout the enterprise.

These Metadata Models can be deployed to the MetaMatrix Server (Server). The Server can then use the Metadata at runtime to:

- Process queries posed by a user application
- Retrieve data from information sources
- Return the integrated results in a useful information format

The MetaMatrix Server parses queries based upon the Metadata information and distributes the sub-queries to the appropriate EIS(s) through Connectors. These Connectors are Java classes that translate queries into the EIS's native application programming interface (API). Once the various EISs return the data results, the MetaMatrix Server reassembles and returns those results to the client application.

The MetaMatrix Enterprise Release 5.5.3 Product is comprised of the following software components:

- Design-Time Components:
  - MetaMatrix Repository
  - MetaMatrix Enterprise Designer
- Run-Time Components:
  - MetaMatrix Server
  - Connectors
  - MetaMatrix QueryBuilder
- Supporting Software Components:
  - MetaMatrix Platform
  - MetaMatrix Web Services
  - MetaMatrix ODBC and JDBC Drivers
- Management Components:
  - MetaMatrix Enterprise Console
  - MMAdmin Scripting Environment

1.4.3.1 MetaMatrix Metadata Repository ('MMR ')

The MetaMatrix Metadata Repository (MMR) is used to manage and share an enterprise's Metadata assets, and expose them for programmatic access to Metadata-aware applications. The MetaMatrix Metadata Repository software provides for the storage, versioning, sharing, and searching of Metadata models created and edited using the MetaMatrix Enterprise Designer. The MMR also prevents concurrent editing of models using a familiar check out, check in paradigm.

The system is designed to accept new models and new versions of models without restart.

MetaMatrix Enterprise Release 5.5.3 relies on the use of a third party relational database for internal data storage; the internal repository is not included as part of the TOE.

1.4.3.2 MetaMatrix Enterprise Designer ('Designer')

The MetaMatrix Enterprise Designer is a graphical user interface (GUI) tool that is used to define and edit Metadata models, representing data sources to be integrated and abstract views (virtual models) of those data sources to be used by client applications using the MetaMatrix Server. Virtual models are defined in terms of transformations from physical models or other virtual models. The Designer presents the transformations that define the virtual model in a graphical interface, to enable understanding of relationships. Transformations once created can be navigated to see how elements at various parts of the models are related. Metadata models are hierarchically organized into folders and projects.

The MetaMatrix Designer comes with several standard plug-ins and importers that can be used to import Metadata from common sources.

Standard plug-ins are available to import data from:

- A JDBC-compliant database.
- The File System (any file from any location.)
- Zip files.

MetaMatrix Designer provides importers for:

- Common relational DBMS systems (relational)
- Erwin files (relational, with UML and relationships)
- Popkin System Architect (relational, with UML and relationships)
- Rational Rose (UML)

The Designer is a development tool and is not part of the run-time environment and will not be included in the scope of the evaluation.

1.4.3.3 MetaMatrix Server ('Server')

The MetaMatrix Server is the data integration engine that enables data access and integration from disparate enterprise information systems (EISs). The MetaMatrix Server resolves queries from applications that use its API and processes these queries, extracting information and updating data in disparate EISs. The Server provides a single access point to the applications for managing the EISs.

A sub-component of the Server is the Query Engine used to process queries from the client applications. The query engine may, during optimization, break a single user-level query into multiple independent queries against a particular connector.

Another sub-component of the Server is the Enterprise XA Server which provides multi-source updates and transactions. MetaMatrix supports distributed transactions across multiple sources, in a manner compliant with the XA specification. Within the context of a distributed transaction, updates (inserts, updates, or deletes, or any combination of them with selects) can be performed across multiple data sources. The MetaMatrix XA Server logs all transactions that it executes to a table in the MetaMatrix System database. Administrators can access this table with standard database reporting tools.

1.4.3.4 Connectors

The Connectors provide the access to the data sources on the EISs. Connectors are written as Java classes that implement well-defined interfaces. Connectors are deployed and configured using the MetaMatrix Console. The MetaMatrix Connector framework provides access to:

- Connection pooling
- Transaction management system
- Runtime Metadata
- The MetaMatrix logging system

MetaMatrix provides the following Connectors with the product:

- RDBMS: Oracle, SQL Server, DB2, Sybase, PostgreSQL, MySQL, Derby, MetaMatrix, Generic JDBC, Generic ODBC
- Other databases: Adabase, Allbase, Essbase
- Files: Delimited text, XML, Excel, Web Services
- ERP: SAP, PeopleSoft, JD Edwards, Oracle Applications, Siebel
- Mainframe: CICS, IMS, VSAM

In addition, MetaMatrix provides a Connector Development Kit (CDK) utility for defining, configuring, and testing new Connectors. (Custom developed Connectors will not be included in the scope of the evaluation.)

1.4.3.5 MetaMatrix QueryBuilder ('QueryBuilder')

The MetaMatrix QueryBuilder is a Web-based tool for issuing test queries to MetaMatrix Server, and viewing query plans for troubleshooting purposes. The QueryBuilder provides developers and database analysts the ability to create and test SQL queries run against the MetaMatrix Server. The MetaMatrix QueryBuilder accesses MetaMatrix in the same way as custom applications. The QueryBuilder is a web-based GUI tool that allows the user to create SQL queries to test the MetaMatrix Server's interaction with the EISs. At the most basic, the user can simply type in a query and submit it. The QueryBuilder also retains a list of queries and subscriptions that have been created and submitted during a session. The user can edit or resubmit items in the history and save these histories in an XML file to edit or submit them in a later session. Through the QueryBuilder tool, it is also possible to view extensive debugging information triggered through the DEBUG option passed in on a query.

1.4.3.6 MetaMatrix Platform ('Platform')

The MetaMatrix Platform is the basic cohesive functionality used by all MetaMatrix products. The Platform is the underlying infrastructure beneath both the Server and the MMR.

The following services comprise the MetaMatrix Platform:

- AuthorizationService

The MetaMatrix AuthorizationService provides for the definition and management of data access and repository access roles, as well as the configuration and management of entitlements granted to data roles and repository roles. The AuthorizationService also provides and controls access to the role and entitlement data stored in the server repository database.

- ConfigurationService

  The MetaMatrix ConfigurationService provides access to and storage of all platform and service configuration properties, in addition to implementing configuration aspects of the Console and MMAdmin CLI.

- SessionService

  The MetaMatrix SessionService manages active session information. Active sessions are stored in a distributed cache and shared between Session services in each VM. Sessions are also persisted in the server repository database.

- MembershipService

  The MetaMatrix MembershipService provides the capability to interact with the third-party Membership Domain Providers used to store and manage information about the TOE users. The MembershipService provide the functionality to cnfigure the Membership Domain Providers and transmit the user information needed for user identification and authentication, and support of the MetaMatrix Access Control policy. MetaMatrix supports three types of Membership Domain Providers:

  - LDAP Membership Domain – defines a connection to one or more LDAP servers
  - File Membership Domain – obtains user and group credentials from a file (not recommended for production use)
  - Custom Membership Domain – allows implementations of the MetaMatrix service provider interface (MembershipDomain SPI) to provide authentication and authorization.

  Only the LDAP Membership Domain will be included in the evaluation.

Additional information about these services, including the configuration properties that may be set for each service, can be found in Appendix B of the MetaMatrix Enterprise Console User's Guide.

1.4.3.7 MetaMatrix Web Services

The MetaMatrix Web Services provide the capability to expose information as web services. Any information integrated through a virtual XML Document model can be exposed as a standard web service. MetaMatrix Web Services are defined as a modeling exercise, using the MetaMatrix Designer. Each web service operation is defined to have a request XML document and a response XML document, where the response document is a virtual XML document (which itself is mapped to underlying information assets). A web service operation procedure can include one or more SQL selects, inserts,

updates, deletes or stored procedure executions, as long as it returns a single XML document conformant with the XML Schema component specified for the output message.

MetaMatrix Web Services can be accessed using SOAP (simple object access protocol) messages either over HTTP or JMS protocols in a request-reply mode. SOAP is a World Wide Web Consortium (W3C) specification.

1.4.3.8 MetaMatrix Enterprise Console ('Console')

The MetaMatrix Enterprise Console is a GUI external interface that enables the management of the MetaMatrix components, including the MMR, the Server, the Platform Services, and the Connectors to the EISs. Using the Console, MetaMatrix can be configured including the hosts, processes, and services that comprise the distributed system. Using the Console, administrators can:

- Deploy system services and manage their lifecycles.
- Monitor memory for each process in the system and monitor service queues.
- Deploy new virtual databases for federated data access to disparate data sources.
- Integrate new Connectors, and bind instances to virtual databases.
- Monitor the queries executing in the system.
- Manage users and groups and their authorizations for accessing Data and Metadata.

1.4.3.9 MMAdmin

MMAdmin is a script based programming environment that enables user to access, monitor and control the MetaMatrix Server and its distributed query processing environment. This tool is built using programming language called BeanShell (http://beanshell.org). MMAdmin can be used in ad-hoc scripting, or to run predefined scripts. MMAdmin is a CLI, not a graphical tool. Features provided by MMAdmin are:

- Programming Features – MMAdmin is a fully functional programming environment with resource flow control and exception management.
- Administrative Functions - The user can connect to a running MetaMatrix Server and invoke any of the Admin API methods to control the MetaMatrix System the same as using the Console. Since MMAdmin is script driven, these tasks can be automated and re-run at a later time.
- Data Access - The user can connect to a VDB, issue any SQL commands, and view the results of the query.
- Migration Tools – MMAdmin can be used to develop scripts like moving the Virtual Databases (VDB), Connector Bindings, and Configuration from one development environment to another enabling users to test and automate their migration scripts before production deployments.
- Testing Tools - The JUnit (http://junit.org) test framework is built into MMAdmin. Users can write regression tests for checking system health, or data integrity that can be used to validate system functionality automatically.

Only the administrative and data access functionality of MMAdmin will be included in the scope of the evaluation.

1.4.3.10 MetaMatrix ODBC and JDBC Drivers

MetaMatrix provides an ODBC Driver and a JDBC Driver that are used by user applications which access the Metadata in the MMR. These user applications can be third-party products such as Excel, or custom-coded user interfaces. The MetaMatrix ODBC or JDBC Driver is installed on a client workstation and supplies part of the translation between user applications that use ODBC or JDBC and the MMR. All transactions performed by user applications that access the MMR can be logged. The third-party and custom-coded user applications are not included in the TOE.

1.4.4 Data

TSF Data includes the systems parameters set by administrators to configure the security of the TOE. Examples of TSF Data include administrative roles and audit logging parameters.

User Data includes the Data collected by the Connectors from the data sources on the EISs and the Metadata which defines the structure and relationships of this data. Metadata is stored in the MetaMatrix Metadata Repository and can be accessed by both administrators and non-administrative users. Access to both Data and Metadata is controlled by the assignment of Entitlements to users and user groups.

1.4.5 Users

Administrators are those users who have access to the TSF Data through the administrative interface components: the MetaMatrix Console, MMAdmin CLI, and the MetaMatrix Designer. Access to administrative functions is restricted further by the defined administrative roles: SystemAdmin, ProductAdmin, and ReadOnlyAdmin*.*

Users may access user data through the MetaMatrix QueryBuilder, third-party user applications or user applications that have been created with third-party development tools to use the Metadata models configured by administrators with the Designer. Users do not have access to TSF Data or administrative functions.

1.4.6 Product Guidance

The following product guidance documents are provided with MetaMatrix Enterprise Release 5.5.3. The documents are available in PDF format on the installation media and via FTP download.

Table 1-6: MetaMatrix User Guidance Documents

MetaMatrix Administration Shell Users Guide, MetaMatrix Products, Release 5.5.3, October 2008
MetaMatrix Connector Developer's Guide, MetaMatrix Products, Release 5.5.3, October 2008
MetaMatrix Custom Scalar Functions Tutorial, MetaMatrix Products, Release 5.5.3, October 2008
MetaMatrix Enterprise 5.5.3 – README, MetaMatrix Enterprise Server, Release 5.5.3, Build 3126, October 15, 2008
MetaMatrix Enterprise Administration Guide, MetaMatrix Products, Release 5.5.3, October 2008
MetaMatrix Enterprise Client Developer's Guide, MetaMatrix Products, Release 5.5.3, October 2008
MetaMatrix Enterprise Console User's Guide, MetaMatrix Products, Release 5.5.3, October 2008
MetaMatrix Enterprise Data Caching, MetaMatrix Products, Release 5.5.3, October 2008

MetaMatrix Enterprise Designer User's Guide, MetaMatrix Products, Release 5.5.3, October 2008
MetaMatrix Enterprise Installation Guide, MetaMatrix Products, Release 5.5.3, October 2008
MetaMatrix Enterprise QueryBuilder User's Guide, MetaMatrix Products, Release 5.5.3, October 2008
MetaMatrix Enterprise Server Tuning Guide, MetaMatrix Products, Release 5.5.3, October 2008
MetaMatrix Enterprise SSL Guide, MetaMatrix Products, Release 5.5.3, Rev A. May 2009
MetaMatrix Enterprise XQuery Guide, MetaMatrix Products, Release 5.5.3, October 2008
MetaMatrix Feature Overview and Value Proposition, MetaMatrix Products, Release 5.5.3, October 2008
MetaMatrix Guide to the Design Time Catalog, MetaMatrix Products, Release 5.5.3, October 2008
MetaMatrix JDBC Connector Integration Guide, MetaMatrix Products, Release 5.5.3, October 2008
MetaMatrix Known Issues, MetaMatrix Products, Release 5.5.3, October 2008
MetaMatrix Membership Domain Guide, MetaMatrix Products, Release 5.5.3, October 2008
MetaMatrix Metadata Repository User Guide, MetaMatrix Products, Release 5.5.3, October 2008
MetaMatrix Oracle Spatial Connector Integration Guide, MetaMatrix Products, Release 5.5.3, October 2008
MetaMatrix Release Notes, MetaMatrix Products, Release 5.5.3, October 2008
MetaMatrix Server Security, User Authentication, and Authorization, MetaMatrix Products, Release 5.5.3, October 2008
MetaMatrix Text File Connector Integration Guide, MetaMatrix Products, Release 5.5.3, October 2008
MetaMatrix Web Services Guide, MetaMatrix Products, Release 5.5.3, October 2008
MetaMatrix XML-Relational Connectors Reference Guide, MetaMatrix Products, Release 5.5.3, October 2008
SQL Query Web Service User's Guide, MetaMatrix Products, Release 5.5.3, October 2008

The Console and Designer User Interfaces also provide on-line help.

1.4.7 Physical Scope of the TOE

The physical boundary of the TOE is the entire product as commercially available from the developer except for non-runtime and deprecated components. The TOE consists of the MetaMatrix components described in Section 1.4.3. The TOE Boundary is depicted in the figure below.

.

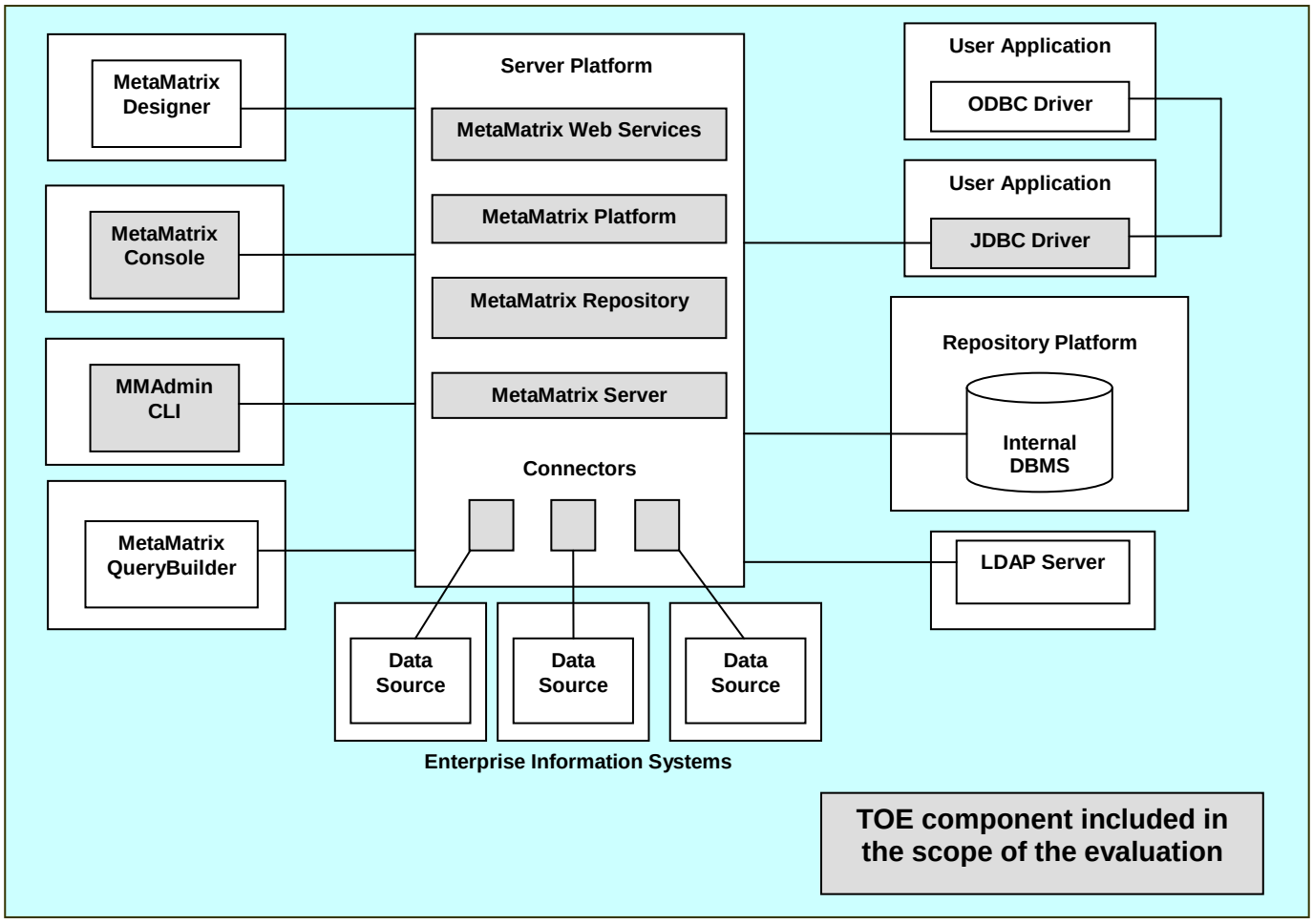

Figure 2: TOE Physical Boundary

1.4.7.1 Included in the TOE:

The scope of the evaluation will include the following product components and/or functionality:

- MetaMatrix Metadata Repository
- MetaMatrix Server
- Connectors
- MetaMatrix Platform
- MetaMatrix Web Services
- MetaMatrix Enterprise Console
- MMAdmin CLI (administration and data access functionality)
- MetaMatrix ODBC and JDBC Drivers installed on client workstations to support user applications.

The following product components and/or functionality are excluded from the scope of the evaluation: (While these components are included in the TOE, they are not part of the TSF and will not be tested.)

- Product components not used during normal operation (runtime) of the TOE:
  - MetaMatrix Enterprise Designer
  - MetaMatrix Query Builder
  - Connector Developer Kit
  - Command Line Interface utilities used during the initial installation and configuration of product
  - MMAdmin CLI (programming, migration and testing tools)
- Depreciated product components
  - MetaMatrix Dimension Designer
  - MetaMatrix Reporter
  - adminshell (precursor to MMAdmin CLI)

1.4.7.2 Excluded from the TOE:

The following are included in the IT Environment and are not part of the TOE:

- Underlying third party relational database (internal repository)
- Custom Created Connectors
- Third-party user applications
- Custom-coded user applications
- Underlying operating system (OS) software and hardware of the TOE component host platforms
- SSL implementation
- Tomcat/Apache web services
- Transport standards HTTP, HTTPS, and FTP implementations
- Membership Domain Providers
- LDAP server and interfaces
- Databases and applications used as data sources for testing

1.4.8 Logical Scope of the TOE

MetaMatrix provides the following security functionality:

- **Security audit**

  MetaMatrix's auditing capabilities include recording information about system processing and users' access to the TOE. Subject identity (user login name) and outcome are recorded for each

event audited. The audit records generated by MetaMatrix are protected by the TSF working in conjunction with the protection mechanisms of the IT Environment.

- **User data protection**

  MetaMatrix provides its own access control separate from the IT Environment between subjects and objects covered by the MetaMatrix Access Control SFP. MetaMatrix provides facilities to define and manage permissions to control a user's access to both Data from the EISs and Metadata stored in the MMR.

- **Identification and authentication**

  Each user must be successfully identified and authenticated with a username and password by the TSF or by an authentication service invoked by the TSF before access is allowed to MetaMatrix. There are two ways that users can be authenticated to the MetaMatrix system in the evaluated configuration. The first is through username and password. The second is through authentication by a third-party Membership Domain Provider. The TSF maintains security attributes for each individual TOE user for the duration of the user's login session.

- **Security management**

  MetaMatrix provides role-based security management functions through the use of the Console. Through the enforcement of the MetaMatrix Access Control SFP, the ability to manage various security attributes, system parameters and all TSF data is controlled and limited to those users who have been assigned the appropriate administrative role.

2 Conformance Claims

2.1 Common Criteria Conformance

The TOE is Part 2 extended, Part 3 conformant, and meets the requirements of Evaluation Assurance Level (EAL) 2 from the Common Criteria Version 3.1 R2.

2.2 Protection Profile Claim

This ST does not claim conformance to any existing Protection Profile.

2.3 Package Claim

This ST claims conformance to the EAL2 assurance requirements package.

3 Security Problem Definition

3.1 Threats

The TOE must counter the threats to security listed in Table 3-1. The assumed level of expertise of the attacker is unsophisticated, with access to only standard equipment and public information about the product.

Table 3-1: TOE Threats

Item	Threat ID	Threat Description
1	T.AdminError	An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.
2	T.AdminRogue	An administrator's intentions may become malicious resulting in user or TSF data being compromised.
3	T.AuditCompromise	A user or process may gain unauthorized access to the audit trail and cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking a security relevant event.
4	T.MaliciousTSFCompromise	A malicious user or process may cause TSF data or executable code to be inappropriately accessed (viewed, modified, or deleted).
5	T.Masquerade	A malicious user, process, or external IT entity may masquerade as an authorized entity in order to gain access to data or TOE resources.
6	T.UnauthorizedAccess	A user may gain access to user data for which they are not authorized according to the TOE security policy.

3.2 Organizational Security Policies

The Organizational Security Policies of the TOE are defined in Table 3-2.

Table 3-2: TOE Organizational Security Policies

Item	Threat ID	Threat Description
1	P.Accountability	The authorized users of the TOE shall be held accountable for their actions within the TOE.
2	P.Password	The TOE users will select secure passwords that meet the Password Policy defined in the user guidance documentation

3.3 Assumptions

The assumptions regarding the security environment and the intended usage of the TOE are listed in Table 3-3.

Table 3-3: Assumptions

Item	Assumption ID	Assumption Description
1	A.Manage	The TOE assumes there will be one or more competent individuals assigned to manage the TOE and the security of the information it contains.
2	A.NoUntrusted	The TOE assumes that there will be no untrusted users and no untrusted software on the TOE server host.
3	A.Physical	The TOE assumes the hardware and software critical to the security policy enforcement will be protected from unauthorized physical modification.
4	A.ProtectComm	Those responsible for the TOE will ensure the communications between the TOE components and between the TOE components and the remote users are via a secure channel.
5	A.Users	The TOE assumes that its users will protect their authentication data.
6	A.UserApps	The TOE assumes that user applications that access the MMR data have been developed, installed and maintained in a secure manner.

4 Security Objectives

4.1 Security Objectives for the TOE

The security objectives for the TOE are listed in Table 4-1.

Table 4-1: TOE Security Objectives

Item	TOE Objective	Description
1	O.AdminRole	The TOE will provide administrator roles to isolate administrative actions, and to make the administrative functions available locally and remotely.
2	O.AuditGeneration	The TOE will provide the capability to detect and create records of security-relevant events associated with users.
3	O.AuditStorage	The TOE will provide the capability for secure storage and protection of the audit information from unauthorized users via the TOE interfaces.
4	O.Manage	The TOE will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.
5	O.Mediate	The TOE will protect user data in accordance with its security policy.
6	O.RobustTOEAccess	The TOE will provide mechanisms that control a user's logical access to the TOE.

4.2 Security Objectives for the Operational Environment

The security objectives for the Operational Environment are listed in Table 4-2.

Table 4-2: Security Objectives for the Operational Environment

Item	Environment Objective	Description
7	OE.AuditStorage	The IT Environment will provide a means for secure storage and protection of the TOE audit information from unauthorized users via the IT Environment interfaces.
8	OE.Install	Those responsible for the TOE must ensure that the TOE is delivered, installed, managed, and operated in a manner which is consistent with IT security.
9	OE.NoUntrusted	Those responsible for the TOE must ensure that there are no untrusted users and no untrusted software on the platforms that host the TOE components.
10	OE.PassSelect	Those responsible for the TOE will ensure the TOE users will select a password according to requirements in the user guidance.
11	OE.Person	Personnel working as authorized administrators shall be carefully selected and trained for proper operation of the system.
12	OE.Physical	Those responsible for the TOE must ensure that those parts of the TOE critical to the security policy are protected from any physical attack.
13	OE.ProtectAuth	Users must ensure that their authentication data is held securely and not disclosed to unauthorized persons.
14	OE.ProtectComm	Those responsible for the TOE will ensure the communications between the TOE components and between the TOE components and remote users are via a secure channel.

Item	Environment Objective	Description
15	OE.RobustTOEAccess	The IT Environment will provide an authentication service for user identification and authentication that can be invoked by the TSF to control a user's logical access to the TOE.
16	OE.SecureComms	The IT Environment will provide a secure line of communications between distributed portions of the TOE and between the TOE and remote users.
17	OE.TimeStamps	The underlying operating system will provide reliable time stamps.
18	OE.UserApps	Those responsible for the TOE will ensure that user applications that access the MMR data have been developed, installed and maintained in a secure manner.

4.3 Security Objectives Rationale

Table 4-3: Mapping of TOE Security Objectives to Threats/Policies

Item	TOE Objective	Threat
1	O.AdminRole	T.AdminError T.AdminRogue
2	O.AuditGeneration	T.AdminError
3	O.AuditStorage	T.AuditCompromise
4	O.Manage	T.AdminError T.MaliciousTSFCompromise
5	O.Mediate	T.UnauthorizedAccess
6	O.RobustTOEAccess	T.Masquerade

Table 4-4: Mapping of Security Objectives for the Operational Environment to Threats/Policies/Assumptions

Item	Environment Objective	Threat/Policy/Assumption
7	OE.AuditStorage	T.AuditCompromise
8	OE.Install	A.Manage
9	OE.NoUntrusted	A.NoUntrusted
10	OE.PassSelect	P.Password
11	OE.Person	A.Manage
12	OE.Physical	A.Physical
13	OE.ProtectAuth	A.Users
14	OE.ProtectComm	A.ProtectComm
15	OE.RobustTOEAccess	T.Masquerade
16	OE.SecureComms	T.MaliciousTSFCompromise
17	OE.TimeStamps	P.Accountability
18	OE.UserApps	A.UserApps

Table 4-5 shows that all the identified Threats to security are countered by Security Objectives. Rationale is provided for each Threat in the table.

Table 4-5: All Threats to Security Countered

Item	Threat ID	Objective	Rationale
1	T.AdminError An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.	O.AdminRole The TOE will provide administrator roles to isolate administrative actions, and to make the administrative functions available locally and remotely.	This objective plays a role in mitigating this threat by limiting the functions an administrator can perform in a given role.
		O.AuditGeneration The TOE will provide the capability to detect and create records of security-relevant events associated with users.	This objective also contributes to mitigating this threat by providing the TOE with an audit logging function. If an administrative error is made it will be traceable with the audit log.
		O.Manage The TOE will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.	This objective also contributes to mitigating this threat by providing management tools to make it easier for administrators to manage the TOE security functions. More specifically, providing administrators the capability to view configuration settings within a GUI.
2	T.AdminRogue An administrator's intentions may become malicious resulting in user or TSF data being compromised.	O.AdminRole The TOE will provide administrator roles to isolate administrative actions, and to make the administrative functions available locally and remotely.	This objective mitigates this threat by restricting the functions available to an administrator. This is somewhat different than the part this objective plays in countering T.AdminError, in that this presumes that separate individuals will be assigned separate roles.
3	T.AuditCompromise A user or process may gain unauthorized access to the audit trail and cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking a security relevant event.	O.AuditStorage The TOE will provide the capability for secure storage and protection of the audit information from unauthorized users via the TOE interfaces.	This objective mitigates this threat by protecting the audit records through the TOE interfaces.
		OE.AuditStorage The IT Environment will provide a means for secure storage and protection of the TOE audit information from unauthorized users via the IT Environment interfaces..	This objective mitigates this threat by protecting the audit records through the IT Environment interfaces. Together with O.AuditStorage, this provides full protection of the audit records.

Item	Threat ID	Objective	Rationale
4	T.MaliciousTSFCompromise A malicious user or process may cause TSF data or executable code to be inappropriately accessed (viewed, modified, or deleted).	O.Manage The TOE will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.	This objective provides the capability to restrict access to the TSF to those that are authorized to use the functions. Satisfaction of this objective (and its associated requirements) prevents unauthorized access to TSF functions and data through the administrative mechanisms (ex. GUI).
		OE.SecureComms The IT Environment will provide a secure line of communications between distributed portions of the TOE and between the TOE and remote users.	This objective provides for the protection of TSF data while in transmission.
5	T.Masquerade A malicious user, process, or external IT entity may masquerade as an authorized entity in order to gain access to data or TOE resources.	O.RobustTOEAccess The TOE will provide mechanisms that control a user's logical access to the TOE.	This objective mitigates this threat by controlling the logical access to the TOE and its resources. By constraining how authorized users can access the TOE, and by mandating the type and strength of the authentication mechanisms, this objective helps mitigate the possibility of a user attempting to login and masquerade as an authorized user. In addition, this objective allows the TOE to correctly interpret information used during the authentication process so that it can make the correct decisions when identifying and authenticating users.
		OE.RobustTOEAccess The IT Environment will provide an authentication service for user identification and authentication that can be invoked by the TSF to control a user's logical access to the TOE.	This objective mitigates this threat by controlling the logical access to the TOE and its resources. Authentication mechanisms in the IT Environment may also be used to control access to the TOE.
6	T.UnauthorizedAccess A user may gain access to user data for which they are not authorized according to the TOE security policy.	O.Mediate The TOE will protect user data in accordance with its security policy.	This objective works to mitigate this threat by requiring that TOE objects are protected using access control items. An access control item contains information about who is allowed to access an object, as well as the allowed modes of access. The settings present in the access control item selected in the access control decision process determine whether or not a user is authorized to access the object.

Table 4-6 shows that the security objectives for the operational environment enforce all Organizational Security Policies. Rationale is provided for each Policy in the table.

Table 4-6: All Security Policies Enforced

Item	OSP ID	Objective	Rationale
1	P.Accountability The authorized users of the TOE shall be held accountable for their actions within the TOE.	OE.TimeStamps The underlying operating system will provide reliable time stamps.	This objective ensures that the underlying operating system will provide reliable time stamps in support of the audit function.
2	P.Password The TOE users will select secure passwords that meet the Password Policy defined in the user guidance documentation	OE.PassSelect Those responsible for the TOE will ensure the TOE users will select a password according to requirements in the User Guidance.	The objective ensures that the policy for secure passwords is enforced by the TOE administrators.

Table 4-7 shows that the security objectives for the operational environment uphold all assumptions. Rationale is provided for each Assumption in the table.

Table 4-7: All Assumptions Upheld

Item	Assumption ID	Objective	Rationale
1	A.Manage The TOE assumes there will be one or more competent individuals assigned to manage the TOE and the security of the information it contains.	OE.Person Personnel working as authorized administrators shall be carefully selected and trained for proper operation of the system.	This objective provides for competent personnel to administer the TOE.
		OE.Install Those responsible for the TOE must ensure that the TOE is delivered, installed, managed, and operated in a manner which is consistent with IT security.	This objective ensures the TOE is delivered, installed, managed, and operated by competent individuals.
2	A.NoUntrusted The TOE assumes that there will be no untrusted users and no untrusted software on the TOE server host.	OE.NoUntrusted Those responsible for the TOE must ensure that there are no untrusted users and no untrusted software on the platforms that host the TOE components.	This objective provides for the protection of the TOE from untrusted software and users.

Item	Assumption ID	Objective	Rationale
3	A.Physical The TOE assumes the hardware and software critical to the security policy enforcement will be protected from unauthorized physical modification.	OE.Physical Those responsible for the TOE must ensure that those parts of the TOE critical to the security policy are protected from any physical attack.	This objective provides for the physical protection of the TOE software.
4	A.ProtectComm Those responsible for the TOE will ensure the communications between the TOE components and between the TOE components and the remote users are via a secure channel.	OE.ProtectComm Those responsible for the TOE will ensure the communications between the TOE components and between the TOE components and remote users are via a secure channel.	This objective provides for secure communications between the TOE components and between the TOE components and remote users.
5	A.Users The TOE assumes that its users will protect their authentication data.	OE.ProtectAuth Users must ensure that their authentication data is held securely and not disclosed to unauthorized persons.	This objective provides for users protecting their authentication data.
6	A.UserApps The TOE assumes that user applications that access the MMR data have been developed, installed and maintained in a secure manner.	OE.UserApps Those responsible for the TOE will ensure that user applications that access the MMR data have been developed, installed and maintained in a secure manner.	This objective provides that only user applications which have been developed, installed and maintained in a secure manner will access MMR data.

5 Extended Components Definition

All of the components defined below have been modeled on components from Part 2 of the CC Version 3.1. The extended components are denoted by adding "_EXT" in the component name.

Table 5-1: Extended Components

Item	SFR ID	SFR Title
1	FAU_STG_EXT.1	Partial protection of the audit trail storage
2	FIA_UAU_EXT.2	TSF user authentication before any action

5.1 FAU_STG_EXT.1 Partial protection of the audit trail storage

5.1.1 Extended Component Definition

5.1.1.1 Class

FAU: Security Audit

5.1.1.2 Family

Security audit event storage (FAU_STG)

5.1.1.3 Family Behaviour

This family defines the requirements for the TSF to be able to create and maintain a secure audit trail. Stored audit records refers to those records within the audit trail, and not the audit records that have been retrieved (to temporary storage) through selection.

At FAU_STG_EXT.1 Partial protection of the audit trail storage, requirements are placed on the audit trail. It will be protected from unauthorised deletion and/or modification through the security functionality of the TSF via the TOE interfaces.

5.1.1.4 Management

There are no management activities foreseen.

5.1.1.5 Audit

There are no auditable events foreseen.

5.1.1.6 Definition

FAU_STG_EXT.1 Partial protection of the audit trail storage

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation

FAU_STG_EXT.1.1 The TSF shall protect the stored audit records in the audit trail from unauthorised deletion through the TSFI.

FAU_STG_EXT.1.2 The TSF shall be able to [selection, choose one of: *prevent*, *detect*] unauthorised modifications to the stored audit records in the audit trail through the TSFI.

5.1.2 Rationale

FAU_STG_EXT.1 is modeled closely on the standard component FAU_STG.1: Protected audit storage. FAU_STG_EXT.1 needed to be defined as an extended component because the TOE requires the TSF and IT Environment working in tandem to provide complete protection of the audit records.

5.2 FIA_UAU_EXT.2 TSF user authentication before any action

5.2.1 Extended Component Definition

5.2.1.1 Class

FIA: Identification and authentication

5.2.1.2 Family

User authentication (FIA_UAU)

5.2.1.3 Family Behaviour

This family defines the types of user authentication mechanisms supported by the TSF. This family also defines the required attributes on which the user authentication mechanisms must be based.

FIA_UAU_EXT.2 TSF user authentication before any action, requires that users are authenticated either by the TSF or by an authentication service invoked by the TSF in conjunction with the IT Environment before any other action will be allowed by the TSF.

5.2.1.4 Management

The following actions could be considered for the management functions in FMT:

- management of the authentication data by an administrator;
- management of the authentication data by the user associated with this data.

5.2.1.5 Audit

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

- Minimal: Unsuccessful use of the authentication mechanism;
- Basic: All use of the authentication mechanism.

5.2.1.6 Definition

FIA_UAU_EXT.2 TSF user authentication before any action

Hierarchical to: FIA_UAU.1 Timing of authentication

Dependencies: FIA_UID.1 Timing of identification

FIA_UAU_EXT.2.1 The TSF shall require each user to be successfully authenticated either by the TSF or by an authentication service in the IT Environment invoked by the TSF before allowing any other TSF-mediated actions on behalf of that user.

5.2.2 Rationale

FIA_UAU_EXT.2 is modeled closely on the standard component FIA_UAU.2: User authentication before any action. FIA_UAU_EXT.2 needed to be defined as an extended component because the standard component was broadened by adding the text *"either by the TSF or by an authentication service in the IT Environment invoked by the TSF".*

6 Security Requirements

This section provides the security functional and assurance requirements for the MetaMatrix TOE.

6.1 Security Functional Requirements for the TOE

Formatting Conventions

The notation, formatting, and conventions used in this security target (ST) are consistent with version 3.1 of the Common Criteria for Information Technology Security Evaluation.

The CC permits four functional component operations: assignment, iteration, refinement, and selection to be performed on functional requirements. These operations are defined as:

iteration: allows a component to be used more than once with varying operations;

assignment: allows the specification of parameters;

selection: allows the specification of one or more items from a list; and

refinement: allows the addition of details.

This ST indicates which text is affected by each of these operations in the following manner:

- *Assignments* and *Selections* specified by the ST author are in **[*italicized bold text*]**.
    - *Refinements* are identified with "**Refinement:**" right after the short name. Additions to the CC text are specified in ***italicized bold and underlined text***.
- *Iterations* are identified with a dash number "-#". These follow the short family name and allow components to be used more than once with varying operations. "*" refers to all iterations of a component.
- *Application notes* provide additional information for the reader, but do not specify requirements. Application notes are denoted by *italicized text.*
- *Extended components* defined in Section 5 have been denoted with the suffix "_EXT" following the family name.

The functional security requirements for the TOE consist of the following components taken directly from Part 2 of the CC and the extended components defined in Section 5, and summarized in Table 6-1 below.

Table 6-1: Functional Components

Item	SFR ID	SFR Title
1	FAU_GEN.1	Audit data generation
2	FAU_GEN.2	User identity association
3	FAU_STG_EXT.1	Partial protection of the audit trail storage
4	FDP_ACC.1	Subset access control
5	FDP_ACF.1	Security attribute based access control
6	FIA_ATD.1	User attribute definition
7	FIA_UAU_EXT.2	TSF user authentication before any action
8	FIA_UID.2	User identification before any action

Item	SFR ID	SFR Title
9	FMT_MSA.1	Management of security attributes
10	FMT_MSA.3	Static attribute initialisation
11	FMT_MTD.1	Management of TSF data
12	FMT_SMF.1	Specification of management functions
13	FMT_SMR.1	Security roles

6.1.1 Class FAU: Security Audit

6.1.1.1 FAU_GEN.1 Audit data generation

Hierarchical to: No other components.

Dependencies: FPT_STM.1 Reliable time stamps

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

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

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

c) **[*the following auditable events:***

- ***User commands***
- ***Access to data resources including attempts where users are denied access as well as granted access***

].

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: **[*the additional information identified in Table 6-2 and Table 6-3*]**.

Table 6-2: Auditable Events

Item	SFR ID	Auditable Event
1	FAU_GEN.1	None
2	FAU_GEN.2	None
3	FAU_STG_EXT.1	None
4	FDP_ACC.1	None
5	FDP_ACF.1	All requests to access EIS data and Metadata
6	FIA_ATD.1	None
7	FIA_UAU_EXT.2	Failed logins Successful logins
8	FIA_UID.2	Failed logins Successful logins

Item	SFR ID	Auditable Event
9	FMT_MSA.1	Add/delete/modify Admin Roles Add/delete/modify Data Roles Add/delete/modify Repository Roles
10	FMT_MSA.3	Add/delete/modify Admin Roles Add/delete/modify Data Roles Add/delete/modify Repository Roles
11	FMT_MTD.1	Modify Audit Log Parameters Configure System Parameters Define Membership Domain Providers Configure Membership Domain Provider Properties Add/Remove Admin Role to/from Membership Domain Group Create and modify Data Role Add/Remove Data Role to/from Membership Domain Group Create and modify Repository Role Add/Remove Repository Role to/from Membership Domain Group Create, modify and delete Data Object Security Attributes Create, modify and delete Metadata Object Security Attributes Configure Server Parameters Terminate Queries Start/Stop MetaMatrix TOE Services
12	FMT_SMF.1	Same as auditing for FMT_MTD.1
13	FMT_SMR.1	Modification of user roles (Admin, Data, and Repository roles)

Table 6-3: Audit Record Fields

	Field	Description
1	TIMESTAMP	YYYY.MM.DD HH:MM:SS
2	PRINCIPAL	Login Name of the User
3	HOSTNAME	Host Server Name
4	VMID	Name of the MetaMatrix Java Virtual Machine
5	CONTEXT	“query”, “update”, “insert”, or “delete”
6	ACTIVITY	Defines whether the entry is a request or a response. “getInaccessibleResources-” is a standard description of an internal method call which is then followed by “request”, “denied”, “granted all”, or “not granted”
7	RESOURCES	List of all data source nodes being accessed by the request or response, delimited by semi-colon. For example, “Books_Source.AUTHORS.AUTHOR_ID; Books_Source.AUTHORS.FIRSTNAME; Books_Source.AUTHORS.LASTNAME.”

6.1.1.2 FAU_GEN.2 User identity association

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation

FIA_UID.1 Timing of identification

FAU_GEN.2.1 For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

6.1.1.3 FAU_STG_EXT.1 Partial protection of the audit trail storage

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation

FAU_STG_EXT.1.1 TSF shall protect the stored audit records in the audit trail from unauthorised deletion through the TSFI.

FAU_STG_EXT.1.2 The TSF shall be able to **[*prevent*]** unauthorised modifications to the stored audit records in the audit trail through the TSFI.

6.1.2 Class FDP: User Data Protection

6.1.2.1 FDP_ACC.1 Subset access control

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1 The TSF shall enforce the **[*MetaMatrix Access Control SFP*]** on **[**

Subjects: process acting on behalf of users

Objects: Data, Metadata

Operations: create, read, update and delete

].

6.1.2.2 FDP_ACF.1 Security attribute based access control

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control

FMT_MSA.3 Static attribute initialisation

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

[

- ***Subjects: process acting on behalf of users***
- ***Subject security attributes:***

  - ***Username***
  - ***Membership Domain Group(s),***

    where each group may have assigned:
    - ***Admin Role***
    - ***Data Entitlements***
    - ***Metadata Entitlements***
- ***Objects: Data, Metadata***
- ***Object security attributes:***
  - ***VDB***
  - ***Model***
  - ***Category***
  - ***Group***

].

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

1. ***A group must be explicitly or implicitly be granted explicit or implicit access to a specific column to perform an enumerated set of operations - Create, Read, Update, or Delete - on that VDB, model, category, and/or group.***
2. ***A user is implicitly granted permission if he/she belongs to a group which has been granted the permission.***
3. ***Access to a model, table, and/or column can be explicitly granted by specifying the VDB, model, category, and/or group column's name directly.***
4. ***Any combination of Create, Read, Update, and Delete operations may be granted for each VDB, model, category, and/or group column and group.***
5. ***If all groups assigned to a user are not granted a privilege to perform an operation on a specific column (as stated above), the user is by default denied access to that VDB, model, category, and/or group. The default for the system is that no users have access to any information.***

].

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

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

6.1.3 Class FIA: Identification and authentication

6.1.3.1 FIA_ATD.1 User attribute definition

Hierarchical to: No other components.

Dependencies: No dependencies.

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

[

- ***Username***
- ***Password***
- ***Membership Domain Group(s)***

  Where each group may have assigned:
  - ***Admin Role***
  - ***Data Entitlements (Data Roles)***
  - ***Metadata Entitlements (Repository Roles)***

]

6.1.3.2 FIA_UAU_EXT.2 TSF user authentication before any action

Hierarchical to: FIA_UAU.1 Timing of authentication

Dependencies: FIA_UID.1 Timing of identification

FIA_UAU_EXT.2.1 The TSF shall require each user to be successfully authenticated either by the TSF or by an authentication service in the IT Environment invoked by the TSF before allowing any other TSF-mediated actions on behalf of that user.

6.1.3.3 FIA_UID.2 User identification before any action

Hierarchical to: FIA_UID.1 Timing of identification

Dependencies: No dependencies.

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 Class FMT: Security Management (FMT)

6.1.4.1 FMT_MSA.1 Management of security attributes

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or

FDP_IFC.1 Subset information flow control]

FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions

FMT_MSA.1.1 The TSF shall enforce the **[*MetaMatrix Access Control SFP*]** to restrict the ability to **[*query, modify, delete, [other operations as specified in Table 6-4]*]** the security attributes **[*as specified in Table 6-4]*** to ***[the role as specified in Table 6-4*** **].**

Table 6-4: Management of Security Attributes

Operation	Security Attributes	Role
Add/Remove Admin Role to/from Membership Domain Group	Admin Role Membership Domain Groups	SystemAdmin
Create and modify	Data Role (set of Data Entitlements)	SystemAdmin
Add/Remove Data Role to/from Membership Domain Group	Data Role Membership Domain Groups	SystemAdmin
Create and modify	Repository Role (set of Metadata Entitlements)	SystemAdmin
Add/Remove Repository Role to/from Membership Domain Group	Repository Role Membership Domain Groups	SystemAdmin

6.1.4.2 FMT_MSA.3 Static attribute initialisation

Hierarchical to: No other components.

Dependencies: FMT_MSA.1 Management of security attributes

FMT_SMR.1 Security roles

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

FMT_MSA.3.2 The TSF shall allow the **[*SystemAdmin*]** to specify alternative initial values to override the default values when an object or information is created.

6.1.4.3 FMT_MTD.1 Management of TSF data

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1 The TSF shall restrict the ability to **[*query, modify, delete, and [other operations as specified in Table 6-5*]** the **[*TSF Data as specified in Table 6-5]* to **[*the role as specified in Table 6-5].***

Table 6-5: Management of TSF Data

Operation	TSF Data	Role
Modify	Audit Log Parameters	SystemAdmin
Configure	System Parameters	SystemAdmin
Define	Membership Domain Providers	SystemAdmin
Configure	Membership Domain Provider Properties	SystemAdmin
Add/Remove Admin Role to/from Membership Domain Group	Admin Role Membership Domain Groups	SystemAdmin

Operation	TSF Data	Role
Create and modify	Data Role (set of Data Entitlements)	SystemAdmin
Add/Remove Data Role to/from Membership Domain Group	Data Role Membership Domain Groups	SystemAdmin
Create and modify	Repository Role (set of Metadata Entitlements)	SystemAdmin
Add/Remove Repository Role to/from Membership Domain Group	Repository Role Membership Domain Groups	SystemAdmin
Create, modify and delete	Data Object Security Attributes Metadata Object Security Attributes	SystemAdmin ProductAdmin
Deploy	Virtual Databases	SystemAdmin ProductAdmin
Configure	Server Parameters	SystemAdmin ProductAdmin
Terminate	Queries	SystemAdmin ProductAdmin
Start/Stop	TOE Services	SystemAdmin ProductAdmin
View (read-only access)	TSF Data displayed within the Console	ReadOnlyAdmin

6.1.4.4 FMT_SMF.1 Specification of Management Functions

Hierarchical to: No other components.

Dependencies: No dependencies.

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

[

- ***operations on the security attributes as specified in Table 6-4 (see FMT_MSA.1)***
- ***operations as specified in Table 6-5 on the TSF Data as specified in Table 6-5 (See FMT_MTD.1)***

].

6.1.4.5 FMT_SMR.1 Security roles

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification

FMT_SMR.1.1 The TSF shall maintain the roles **[*SystemAdmin, ProductAdmin, ReadOnlyAdmin*].**

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

6.2 Security Assurance Requirements for the TOE

The Security Assurance Requirements for the TOE are the assurance components of Evaluation Assurance Level 2 (EAL2) taken from Part 3 of the Common Criteria. None of the assurance components are refined. The assurance components are listed in Table 6-6.

Table 6-6: EAL2 Assurance Components

Item	Class	Component	Component Title
1	ADV: Development	ADV_ARC.1	Security architecture description
2		ADV_FSP.2	Security-enforcing functional specification
3		ADV_TDS.1	Basic design
4	AGD: Guidance documents	AGD_OPE.1	Operational user guidance
5		AGD_PRE.1	Preparative procedures
6	ALC: Life-cycle support	ALC_CMC.2	Use of a CM system
7		ALC_CMS.2	Parts of the TOE CM coverage
8		ALC_DEL.1	Delivery procedures
9	ASE: Security Target evaluation	ASE_CCL.1	Conformance claims
10		ASE_ECD.1	Extended components definition
11		ASE_INT.1	ST introduction
12		ASE_OBJ.2	Security objectives
13		ASE_REQ.2	Derived security requirements
14		ASE_SPD.1	Security problem definition
15		ASE_TSS.1	TOE summary specification
16	ATE: Tests	ATE_COV.1	Evidence of coverage
17		ATE_FUN.1	Functional testing
18		ATE_IND.2	Independent testing – sample
19	AVA: Vulnerability assessment	AVA_VAN.2	Vulnerability analysis

Further information on these assurance components can be found in the Common Criteria for Information Technology Security Evaluation (CCITSE) Part 3.

6.3 Security Requirements Rationale

6.3.1 Dependencies Satisfied

Table 6-7 shows the dependencies between the functional requirements including the extended components defined in Section 5. Dependencies that are satisfied by a hierarchical component are denoted by an (H) following the dependency reference.

Table 6-7: TOE Dependencies Satisfied

Item	SFR ID	SFR Title	Dependencies	Item Reference
1	FAU_GEN.1	Audit data generation	FPT_STM.1	IT Environment*
2	FAU_GEN.2	User identity association	FAU_GEN.1	1
			FIA_UID.1	9 (H)
3	FAU_STG_EXT.1	Partial protection of the audit trail storage	FAU_GEN.1	1
4	FDP_ACC.1	Subset access control	FDP_ACF.1	5
5	FDP_ACF.1	Security attribute based access control	FDP_ACC.1	4
			FMT_MSA.3	11
6	FIA_ATD.1	User attribute definition	None	None
7	FIA_UAU_EXT.2	Cooperative user authentication before any action	FIA_UID.1	9 (H)
8	FIA_UID.2	User identification before any action	None	None
9	FMT_MSA.1	Management of security attributes	FDP_ACC.1	4
			FMT_SMF.1	13
			FMT_SMR.1	14
10	FMT_MSA.3	Static attribute initialisation	FMT_MSA.1	10
			FMT_SMR.1	14
11	FMT_MTD.1	Management of TSF data	FMT_SMF.1	13
			FMT_SMR.1	14
12	FMT_SMF.1	Specification of management functions	None	None
13	FMT_SMR.1	Security roles	FIA_UID.1	9 (H)

* Reliable timestamps are provided by the hardware and OS of the platforms that host the TOE components.

6.3.2 Functional Requirements

Table 6-8 traces each SFR back to the security objectives for the TOE.

Table 6-8: Mapping of TOE SFRs to TOE Security Objectives

Item	SFR ID	TOE Security Objective
1	FAU_GEN.1 Audit data generation	O.AuditGeneration
2	FAU_GEN.2 User identity association	O.AuditGeneration
3	FAU_STG_EXT.1 Partial protection of the audit trail storage	O.AuditStorage
4	FDP_ACC.1 Subset access control	O.Mediate
5	FDP_ACF.1 Security attribute based access control	O.Mediate
6	FIA_ATD.1 User attribute definition	O.RobustTOEAccess
7	FIA_UAU_EXT.2 TSF user authentication before any action	O.RobustTOEAccess

Item	SFR ID	TOE Security Objective
8	FIA_UID.2 User identification before any action	O.RobustTOEAccess
9	FMT_MSA.1 Management of security attributes	O.Manage
10	FMT_MSA.3 Static attribute initialisation	O.Manage
11	FMT_MTD.1 Management of TSF data	O.Manage
12	FMT_SMF.1 Specification of management functions	O.Manage
13	FMT_SMR.1 Security roles	O.AdminRole

Table 6-9 demonstrates that the SFRs meet all security objectives for the TOE. Rationale for each objective is included in the table.

Table 6-9: All TOE Objectives Met by Security Functional Requirements

Item	Objective ID	SFR ID/Title	Rationale
1	O.AdminRole The TOE will provide administrator roles to isolate administrative actions, and to make the administrative functions available locally and remotely.	FMT_SMR.1 Security roles	FMT_SMR.1 requires that the TSF maintain multiple roles. The TSF is able to associate a human user with one or more roles and these roles isolate administrative functions in that the functions of these roles do not overlap. If a security administrator were to perform a malicious action, the auditing requirements enable detection of the rogue platform administrator's actions.
2	O.AuditGeneration The TOE will provide the capability to detect and create records of security-relevant events associated with users.	FAU_GEN.1 Audit data generation	FAU_GEN.1 defines the set of events that the TOE must be capable of recording. This requirement ensures that an administrator has the ability to audit any security relevant event that takes place in the TOE. This requirement also defines the information that must be contained in the audit record for each auditable event. There is a minimum of information that must be present in every audit record and this requirement defines that, as well as the additional information that must be recorded for each auditable event.
		FAU_GEN.2 User identity association	FAU_GEN.2 ensures that the audit records associate a user identity with the auditable event.
3	O.AuditStorage The TOE will provide the capability for secure storage and protection of the audit information from unauthorized users via the TOE interfaces.	FAU_STG_EXT.1 Partial protection of the audit trail storage	FAU_STG_EXT.1 ensures that the TSF provides secure storage and complete protection of the audit records through the TOE's own interfaces.

Item	Objective ID	SFR ID/Title	Rationale
4	O.Manage The TOE will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.	FMT_MSA.1 Management of security attributes	FMT_MSA.1 requires that the TSF restricts the ability to manage security attributes to specified roles. These attributes are defined with respect to the user data access control policy (FDP_ACC.1)
		FMT_MSA.3 Static attribute initialisation	FMT_MSA.3 requires that the TSF enforces the TOE access control policy (FDP_ACC.1) to provide restrictive default values for security attributes
		FMT_MTD.1 Management of TSF data	The FMT requirements are used to satisfy this management objective, as well as other objectives that specify the control of functionality. The requirement's rationale for this objective focuses on the administrator's capability to perform management functions in order to control the behavior of security functions. FMT_MTD.1 specifies the management of TSF Data according to assigned roles.
		FMT_SMF.1 Specification of management functions	FMT_SMF.1 requires the TSF be capable of performing the specified security management functions.
5	O.Mediate The TOE will protect user data in accordance with its security policy.	FDP_ACC.1 Subset access control	The FDP_ACC.1 and FDP_ACF.1 requirements were chosen to define the policies, the subjects, objects, and operations for how and when mediation of access to the user data takes place. FDP_ACC.1 specifies that the subjects under control of the policy are to be defined, and that all operations that involve access to (minimally) the data are controlled by the policy. These objects contain the user data to be protected.
		FDP_ACF.1 Security attribute based access control	FDP_ACF.1 details the manner in which the user data are to be protected. The basics called for by the requirement is to match a set of attributes associated with a subject to a set of "access control items" associated with the object they wish to access; all applicable ACIs need to grant access in order for the subject to perform the operation on the object. The details of how the ACIs are collected and the specific operations supported are specified in FDP_ACF.1.2, and with the attributes define the security policy to be enforced. Setting the attributes (implementing the security policy) is a function of the administrator.
6	O.RobustTOEAccess The TOE will provide mechanisms that control a user's logical access to the TOE.	FIA_UAU_EXT.2 TSF user authentication before any action	FIA_UAU_EXT.2 requires that administrators, authorized IT entities, and other users authenticate themselves to the TOE either through the TSF's authentication mechanism or by a mechanism in the IT Environment that has been invoked by the TSF before performing administrative duties.
		FIA_UID.2 User identification before any action	FIA_UID.2 plays a role in satisfying this objective by ensuring that every user is identified before the TOE performs any mediated functions.

Item	Objective ID	SFR ID/Title	Rationale
		FIA_ATD.1 User attribute definition	FIA_ATD.1 defines the attributes of users, including a user name that is used by the TOE to determine a user's identity and enforce what type of access the user has to the TOE (e.g., the TOE associates a user name with any role(s) they may assume). This requirement allows a human user to have more than one user identity assigned, so that a single human user could assume all the roles necessary to manage the TOE. In order to ensure a separation of roles, this ST requires a single role to be associated with a user name. This is inconvenient in that the administrator would be required to log in with a different account name each time they wish to assume a different role, but this helps mitigate the risk that could occur if an administrator were to execute malicious code.

6.3.3 Assurance Rationale

Evaluation Assurance Level EAL2 was chosen to provide a moderate level of assurance due to the low level threat of malicious attacks.

7 TOE Summary Specification

7.1 IT Security Functions

Section 7.1 describes the specific Security Functions of MetaMatrix Enterprise Release 5.5.3 that meet the criteria of the security features that are described in Section 1.4.8 Logical Scope of the TOE.

The following sub-sections describe how the TOE meets each SFR listed in Section 6.

Table 7-1: Security Functional Requirements Mapped to Security Functions

Security Class	Item	SFRs	Security Functions
Security audit	1	FAU_GEN.1	SA-1
	2	FAU_GEN.2	
	3	FAU_STG_EXT.1	SA-2
User Data Protection	4	FDP_ACC.1	AC-1
	5	FDP_ACF.1	
Identification and authentication	6	FIA_ATD.1	IA-1
	7	FIA_UAU_EXT.2	IA-2
	8	FIA_UID.2	
Security management	9	FMT_MSA.1	SM-1
	10	FMT_MSA.3	SM-2
	11	FMT_MTD.1	SM-3
	12	FMT_SMF.1	SM-4
	13	FMT_SMR.1	SM-5

7.1.1 Security Audit

7.1.1.1 SA-1: Audit Generation

(FAU_GEN.1 and FAU_GEN.2)

MetaMatrix's auditing capabilities include recording information about system processing and users' access to the TOE.

A user must have the SystemAdmin role to modify the system properties related to auditing through the Console and MMAdmin.

The audit functionality can be broken into three areas:

- **System Logging:** records operations the MetaMatrix system has been performing in response to system usage
- **Command Logging:** records SQL commands issued by users to MetaMatrix, as well as SQL commands issued by MetaMatrix to the data source connectors.
- **Audit Logging:** records attempts to access data resources and whether or not access was granted

Subject identity (user login name) and outcome are listed for audit events and command logs, but not for system logs.

System Logging

MetaMatrix provides the capability to view System Log information on the Log Viewer detail panel. The Log Viewer detail panel displays.

- A table of log messages returned
- Details about an individual log message you select from the table.

The table includes the following information for each log entry:

Timestamp - A date and time stamp of when this entry was written.
Message - The text of the log message.
Context - The portion of the MetaMatrix Server or other software that wrote this entry.
Level - The logging level of this message.
Host - The host machine reporting this message.

Further each log entry could be reviewed in greater detail. For each log message the following information is stored and displayed:

Time - A date and time stamp of when this entry was written.
Level - The logging level of this message.
x of - The number of this log message in the table of results.
Context - The portion of the MetaMatrix Server or other software that wrote this entry.
Thread - The thread reporting this exception.
Host - The host machine reporting this message.
Process - The MetaMatrix process reporting this message.
Message - The text of the log message.
Exception - The exception text of this log message, if any.

.

Command Logging

MetaMatrix provides the capability to save a record of all SQL commands or transactions issued to the MetaMatrix Server. In addition, MetaMatrix provides the capability to save a record of all commands or transactions issued against a Connector, which effectively records all interactions with underlying physical data sources The MetaMatrix Server logs all transactions that it executes to a table in the MetaMatrix Metadata Repository. All audit records contain the login name of the user who issued the command.

Administrators can access these tables with standard database reporting tools.

Commands issued to the MetaMatrix Server are logged in the database table **TX_MMXCMDLOG**, which contains the following fields:

RequestID | TxnUID | CmdPoint | SessionUID | App_Name | Principal | VDBName | VDBVersion | Created_TS | Ended_TS | SQL_ID | FINL_ROWCNT

where:

REQUESTID is a unique identifier for every command issued to MetaMatrix Server;

TXNUID is a unique identifier for every transaction;

CMDPOINT is the point in the command being logged, for example ‘BEGIN’ or ‘END';

SESSIONUID is a unique identifier for the user session making the request;

APP_NAME is the name of the client application making the request (may be null);

PRINCIPAL_NA is the username of the user logged into the session;

VDBNAME is the name of the VDB being accessed during the session;

VDBVERSION is the version of the VDB being accessed during the session;

CREATED_TS is the BEGIN command timestamp, formatted YYYY.MM.DD HH:MM:SS;

ENDED_TS is the END command timestamp, formatted YYYY.MM.DD HH:MM:SS;

SQL_ID is the index to the table TX_SQL where the request string for this command is stored;

FINL_ROWCNT is the ROWCOUNT of the query result;

Commands issued by the MetaMatrix Server to the Connectors are logged in the database table TX_SRCCMDLOG, which contains the following fields:

RequestID | NodeID | SubTxnID | CMD_Status | MDL_NM | CNCTRNAME | CMDPOINT | SessionUID | Principal_NA | Created_TS | Ended_TS | SQL_ID | FINL_ROWCNT

where:

REQUESTID is a unique identifier for every command issued to MetaMatrix Server (the same ID value found in the TX_MMXCMDLOG table;

NODEID is an identifier for the connector command logged in this table;

SUBTXNUID is a unique identifier for the sub-transaction to the connector;

CMD_STATUS is the type of request to the connector, either ‘NEW’ or ‘CANCEL’;

MDL_NM is the name of the model being accessed by this connector command;

CNCTRNAME is the name of the connector binding being accessed by this connector command;

CMDPOINT is the point in the command being logged, for example ‘BEGIN’ or ‘END';

SESSIONUID is a unique identifier for the user session making the request;

PRINCIPAL_NA is the username of the user logged into the session;

CREATED_TS is the BEGIN command timestamp, formatted YYYY.MM.DD HH:MM:SS;

ENDED_TS is the END command timestamp, formatted YYYY.MM.DD HH:MM:SS;

SQL_ID is the index to the table TX_SQL where the request string for this command is stored;

FINL_ROWCNT is the ROWCOUNT of the query result;

The string values of the commands logged in both the TX_MMXCMDLOG and TX_SRCCMDLOG tables are stores in a table named **TX_SQL**, which contains the following fields:

SQL_ID | SQL_VL

where:

SQL_ID is a unique identifier (primary key) for every SQL string stored in this table;

SQL_VL is the value of the SQL command, stored as a CLOB;

Audit Logging

MetaMatrix provides also an auditing function that records when users try to execute particular data access commands. Auditing will record all requests for access to entitled resources and whether that access is granted or denied. All requests are logged to two destinations, the AuditEntries table in the MMR and the audit log file (metamatrix.audit in the /Log folder on the Server host).

Auditing is only enabled if set up through the Console and MMAdmin. If auditing is enabled in the system, an audit record is initiated for each requested action to a data source. When auditing is enabled or disabled through the Console, a log entry is made to the MetaMatrix log, as long as the 'audit' logging context is enabled.

Audit logging also only works if the system is configured to use Data Entitlements. The auditing of commands against virtual or physical data tables in a virtual database (VDB) is relative to the Data Entitlements defined for the user trying to execute commands against that VDB.

There are currently no facilities provided with MetaMatrix to view or analyze these audit records. However, standard database tools can be used by administrators to view or create reports against the database table.

The audit log contains entries for requests and entries for responses. The fields recorded in the logfile and AuditEntries database table are:

Timestamp | Principal | Hostname | VMID | Context | Activity | Resources

where:

Timestamp - YYYY.MM.DD HH:MM:SS

Principal - Login Name of the User

Hostname - Host Server Name

VMID – Name of the MetaMatrix Java Virtual Machine

Context – "query", "update", "insert", or "delete"

Activity – Defines whether the entry is a request or a response. If a response, then whether the requested action has been granted. Activity field values are:

- *request* - the requested activity
- *denied* - the full set of actions was disallowed
- *granted all* – the full set of actions was allowed
- *not granted* – the set of actions is only partially allowed, which results in no activity occurring

Resources – List of the data source nodes that were accessed by the request, delimited by semi-colons

7.1.1.2 SA-2: Audit Protection

(FAU_STG_EXT.1)

The audit records generated by the TOE are protected by the TSF working in conjunction with the protection mechanisms of the IT Environment.

Audit records stored in the MetaMatrix Metadata Repository are protected from unauthorized access through the TOE interfaces by the MetaMatrix Access Control Policy. Only authorized administrators, those users assigned one of the three administrative roles, are allowed to access the tables that contain the audit records in the MMR through the Console.

The TOE depends on the protection mechanisms of the third-party RDBMS that implements the MMR to protect the audit records from unauthorized access directly through the RDBMS interfaces. Audit records stored in logfiles are protected from unauthorized access by the access control mechanisms of the OS of the TOE component host platforms. (See OE.AuditStorage)

7.1.2 User Data Protection

7.1.2.1 AC-1: Access Control

(FDP_ACC.1 and FDP_ACF.1)

Authorization controls the privileges of users to access information. This is also referred to as “Entitlements”. Entitlements represent named sets of access rights. Entitlements control which data constructs, such as tables or columns, a user account can create, read, update, and / or delete. MetaMatrix provides facilities to define, manage, and use Entitlements for both data access, and for controlling access to information Metadata.

MetaMatrix Data Entitlements control access to the underlying enterprise information systems. The MetaMatrix Server checks Data Entitlements when requests are made for Data or runtime Metadata.

Metadata Entitlements enable users to log into the MMR and to create, read, update, or delete Metadata models and Metadata model folders and projects within the MMR. The Repository checks Entitlements for design time Metadata when someone tries to check in, check out, or obtain a read-only copy of a model.

Entitlements are defined in the administrative interfaces (Console and MMAdmin CLI). Each Entitlement is named, and defines the access rights to a set of resources. For Data Entitlements, the resources are columns, tables, and models in a specific VDB. For Metadata Entitlements, the resources are Metadata models residing in the MMR. For each resource, the type of access granted is defined (create, read, update, delete). Since resources are defined hierarchically, it is possible to grant access to a set of resources by defining them at a higher-level node, and having those definitions apply to all child nodes. So for example, Data authorizations can be defined at the model level, and apply to all tables (and therefore all columns) within that model.

Similarly, Metadata models can be authorized at the folder level, and will apply to all sub-folders or models contained within that folder.

With MetaMatrix Enterprise Release 5.5.3, all Entitlement permission sets are known as Roles. Admin Roles (authorization to use the management functions, See Section 6.1.4.5 SM-5: Security Roles), Data Roles (Data Entitlement sets) and Repository Roles (Metadata Entitlement sets) are configured independently. Roles are mapped to one or more groups from the Membership Domain Providers. The administrative interfaces allow for the assignment of groups from any of the defined Membership Domain Providers.

For the evaluated configuration, authorization using MetaMatrix-managed Entitlements information will be used for both Data and Metadata.

The Data and Metadata access control algorithm is defined in the following set of rules:

1. ***A group must be explicitly or implicitly be granted explicit or implicit access to a specific column to perform an enumerated set of operations - Create, Read, Update, or Delete - on that VDB, model, category, and/or group.***

   Roles are the "buckets" of access permissions. Roles are assigned to Groups. Groups are defined in the Membership Domain Provider (File, LDAP, Custom). User-Group associations are managed in the Membership Domain Provider also.

2. ***A user is implicitly granted permission if he/she belongs to a group which has been granted the permission.***

   A user has a permission if they belong to a group which is assigned a role containing that permission.

3. ***Access to a model, table, and/or column can be explicitly granted by specifying the VDB, model, category, and/or group column's name directly.***

   Through the Console and MMAdmin applications, models, tables and columns can be individually selected for permissions assignment.

4. ***Any combination of Create, Read, Update, and Delete operations may be granted for each VDB, model, category, and/or group column and group.***

   Any combination of CRUD operation permissions may be granted for each VDB, model, category, and/or group column and group. Those permissions are assigned to a Role which is then associated with a Group.

5. ***If all groups assigned to a user are not granted a privilege to perform an operation on a specific column (as stated above), the user is by default denied access to that VDB, model, category, and/or group. The default for the system is that no users have access to any information.***

   When Data Access Authorizations are enabled, Groups must explicitly have a permission through one of the assigned roles before access is granted. If permission is granted through at least one role, then that group (and any users in that group) has that permission. By default, Data Access Authorizations are not Enabled. However by default there is only the "root" user defined for the system so no one but the root user has access. Part of the security configuration performed during system setup (establishing Membership Domain Provider, creating Roles and assigning permissions) is enabling Data Access Authorizations.

Note: A user must have the SystemAdmin role to modify Entitlements and assign Entitlement sets to groups.

Note: Audit logging only works if the system is configured to use Data Entitlements. The auditing of commands against virtual or physical data tables in a virtual database (VDB) is relative to the Data Entitlements defined for the user trying to execute commands against that VDB.

7.1.3 Identification and Authentication

7.1.3.1 IA-1: User Attributes

(FIA_ATD.1)

The TSF maintains the following security attributes for each individual TOE user:

- Username
- Password
- Membership Domain Group(s)

  Where each group may have assigned:

  - Admin Role – used for access control of the TOE management functions
  - Data Entitlements (Data Roles) – used for access control of the EIS Data
  - Metadata Entitlements (Repository Roles) – used for access control of the Metadata

MetaMatrix has one pre-defined user account: the MetaMatrix Admin. This is the only account with attributes permanently stored within the TOE.

All other user accounts are created, modified and deleted and assigned to groups through the interfaces to the third party Membership Domain Providers.

All role assignment operations in the administrative interfaces are based upon Membership Domain groups. The administrative interfaces do not have access to user level information from Membership Domain Providers. Roles (Administrative, Data and Repository) are assigned to Membership Domain groups through the administrative interfaces (Console and MMAdmin CLI). Access to the management functions for user security attributes is limited to administrators having the SystemAdmin role.

The Admin Roles panel in the Console GUI provides for the assignment of Membership Domain groups to one of the three predefined administrator roles.

Data Roles are a defined set of Entitlements for a specified VDB and VDB version. The Data Roles panel in the MetaMatrix Console provides the controls necessary to create and modify Data Roles. Each Data Role definition must specify a name, a VDB, and VDB version. Groups are mapped to Data Roles exactly the same way as they are mapped to Admin Roles. Create, Read, Update, and Delete authorizations may be set across the models within the Data Role's VDB.

Similarly to Data Roles, Repository Roles are a defined set of Entitlements for accessing models and files within the MMR. The Repository Roles panel in the MetaMatrix Console provides the controls necessary to create and modify Repository Roles in the MetaMatrix Metadata Repository. The SystemAdmin can set the Create, Read, Update, and Delete authorizations for users with this role accessing projects and files within the MMR.

Once a user logs into the TOE and is successfully identified and authenticated, the security attributes belonging to his/her assigned groups are obtained and kept in TOE memory for the duration of the user session.

7.1.3.2 IA-2: User I&A

(FIA_UAU_EXT.2 and FIA_UID.2)

Each user must be successfully identified and authenticated with a username and password by the TSF or by an authentication service in the IT Environment that has been invoked by the TSF before access is allowed to the TOE.

There are two ways that users can be authenticated to the MetaMatrix system in the evaluated configuration. The first is through username and password. The second is through authentication by a third-party Membership Domain Provider.

Username/password authentication is performed directly by the TSF only for the MetaMatrix Admin user account.

The MetaMatrix Admin account is used to log into the MetaMatrix System immediately after installation. MetaMatrix recommends that the MetaMatrix Admin account should only be used to initially configure the Membership Domain Provider(s) and assign administrative roles to one or more users before performing other system administration tasks. The MetaMatrix Admin user account cannot be used to access the MMR or to access virtual databases because Entitlements cannot be assigned to the MetaMatrix Admin account. However, in the event of a Membership Domain failure, the MetaMatrix Admin account will be able to log on and administer the system. The MetaMatrix Admin account credentials are stored in an encrypted format in the system configuration.

All other user identification and authentication is provided by the Membership Domain Provider(s) in the IT Environment upon invocation by the TSF. (See OE.RobustTOEAccess)

MetaMatrix supports three types of Membership Domain Providers:

- LDAP Membership Domain – defines a connection to one or more LDAP servers
- File Membership Domain – obtains user and group credentials from a file (not recommended for production use)
- Custom Membership Domain – allows implementations of the MetaMatrix service provider interface (MembershipDomain SPI) to provide authentication and authorization.

Because the File Member Domain Provider stores passwords in plain text and Custom Membership Domain Providers require custom implementation, only the LDAP Membership Domain Provider will be used in the evaluated configuration. All users, except the MetaMatrix Admin account, will use LDAP authentication.

The LDAP Membership Domain Provider is defined and configured by a user with the SystemAdmin role through the administrative interfaces. More than one LDAP Membership Domains may be active at one time. User accounts do not need to be uniquely named across all Membership Domains. Instead a domain qualifier can be used to indicate which Membership Domain should be used to authenticate the user. For example, user@domain can be used as the user name to include the “user” account in the

“domain” Membership Domain. If a domain qualifier is not provided, then the installed Membership Domains will be accessed in order until a Membership Domain is found that contains the user.

By default, the relative distinguished name (RDN) will be used for LDAP authentication. The RDN must be unique and it must be present for each user within a domain. A user will use the RDN as their login name and the LDAP domain will resolve the RDN to the proper DN before authentication.

7.1.4 Security Management

7.1.4.1 SM-1: Management of Security Attributes

(FMT_MSA.1)

The allowed operations on the security attributes used to enforce the MetaMatrix Access Control SFP and the administrative roles required to execute them are defined in Table 6-4: Management of Security Attributes.

(See Section 6.1.4.1 FMT_MSA.1 Management of security attributes).

7.1.4.2 SM-2: Default Values of Security Attributes

(FMT_MSA.3)

The values of the user security attributes listed in Section 7.1.3.1 IA-1: User Attributes are null by default. Only an administrator with the SystemAdmin role is authorized to change the default values of the user security attributes.

The values of the object security attributes: Model, Table and Column for Data and Metadata objects, are also null by default. Only an administrator with either the SystemAdmin or ProductAdmin role is authorized to change the default values of the object security attributes.

7.1.4.3 SM-3: Management of TSF Data

(FMT_MTD.1)

The allowed operations on TSF Data and the administrative roles required to execute them are defined in Table 6-5: Management of TSF Data (See Section 6.1.4.3 FMT_MTD.1 Management of TSF data).

7.1.4.4 SM-4: Specification of Management Functions

(FMT_SMF.1)

The MetaMatrix System is capable of performing the security management functions as defined in

Table 6-4: Management of Security Attributes (See Section 6.1.4.1 FMT_MSA.1 Management of security attributes) and Table 6-5: Management of TSF Data (See Section 6.1.4.3 FMT_MTD.1 Management of TSF data).

All management functions are limited to the three administrative roles as defined in Section 7.1.4.5 SM-5: Security Roles below.

7.1.4.5 SM-5: Security Roles

(FMT_SMR.1)

The TOE maintains the administrative roles listed below:

- **SystemAdmin –** May perform all functionality in the administrative interfaces (except changes to the Security Summary Panel in the Console, which is changeable only by the MetaMatrix Admin account user).
- **ProductAdmin –** May deploy VDBs, start and stop services, and deploy VDBs.
- **ReadOnlyAdmin –** May monitor the system but may not change any settings.

The administrative roles act as hierarchical roles or levels: having the SystemAdmin role implies having all the capabilities of a ProductAdmin role which, in turn, implies having all of the capabilities of a ReadOnlyAdmin.

These roles determine the access a user has to administrative functions within the MetaMatrix System. User roles do not control access to the underlying data of the enterprise information systems or the MMR; to control this access, Entitlements must be created.

In addition to the roles listed above the TOE provides a pre-defined user account: the MetaMatrix Admin account which has the SystemAdmin role that cannot be removed.

7.2 TOE Protection against Interference and Logical Tampering

The TSF when invoked by the underlying host OS maintains a security domain that protects it from interference and tampering by untrusted subjects in the TOE's Scope of Control. The MetaMatrix protected domain includes all MetaMatrix software components.

Access to user Data, Metadata, system configuration data and management functions is controlled by the MetaMatrix Access Control SFP which is based on the user's assigned security attributes.

In addition to TOE component software, other software files such as configuration files are also stored on disk. These files can be modified only by an authorized user accessing them through the Console, MMAdmin, or an application that uses the MetaMatrix Administrative API. The MetaMatrix Access Control SFP also provides protection when accessing these files.

The underlying assumption regarding the operation of MetaMatrix is that it is maintained in a physically secure environment.

Because MetaMatrix is a software-only TOE, it relies on the security functionality of the IT Environment to provide complete protection.

MetaMatrix relies on the Operating Systems of the TOE components host platforms to provide file access control and process separation at the OS level. It also relies on the third-party RDBMS to protect the data

stored in the MMR and on OpenSSL to provide secure communications between TOE components and between the TOE components and remote users.

Administrator guidance for configuring SSL is provided in the [SSL] document. The following modes are available for SSL connections:

- None – This is the installation default. Socket connections to the server are not encrypted and do not require authentication at the socket layer. NOTE: connections still require user authentication to establish a session.
- Anonymous SSL - Encrypts the socket connections to the server. No socket level authentication is required.
- 1-way SSL - Encrypts the socket connections to the server. Only authenticates the server to the client at the socket level
- 2-way SSL - Encrypts the socket connections to the server. Mutual client and server authentication is required.

The following Java system properties can be used to explicitly configure MetaMatrix client SSL in the recommended MetaMatrix specific configuration:

- 1-Way SSL

  -Dcom.metamatrix.ssl.trustStore=<dir>/metamatrix.truststore (required)

  -Dcom.metamatrix.ssl.trustStorePassword=<password> (required)

  -Dcom.metamatrix.ssl.protocol (optional;default=SSLv3)

  -Dcom.metamatrix.ssl.algorithm (optional;default=SunX509)

  -Dcom.metamatrix.ssl.keyStoreType (optional;default=JKS)

- 2-Way SSL

  -Dcom.metamatrix.ssl.keyStore=<dir>/metamatrix.keystore (required)

  -Dcom.metamatrix.ssl.keyStorePassword=<password> (required)

  -Dcom.metamatrix.ssl.trustStore===<dir>/metamatrix.truststore (required)

  -Dcom.metamatrix.ssl.trustStorePassword=<password> (required)

  -Dcom.metamatrix.ssl.protocol (optional;default=SSLv3)

  -Dcom.metamatrix.ssl.algorithm (optional;default=SunX509)

  -Dcom.metamatrix.ssl.keyStoreType (optional;default=JKS)

7.3 TOE Protection against Bypass of Security Functions

The TSF when invoked by the underlying host OS ensures that TOE Security Policy enforcement functions are invoked and succeed before each function within the TOE's Scope of Control is allowed to proceed. All user operations are conducted in the context of an associated user session. This user session is allocated only after successful identification and authentication by the TSF and IT Environment working together. The user session is destroyed when the corresponding user logs out of that session.

Access to management functions, user data and TSF data is controlled by a user's assigned security attributes. User security attributes are kept in memory and are destroyed when the user session is terminated.

Access to management functions are allowed only for users who have been assigned the required administrative role. Authorized administrators can only view the security attributes and TSF data through the administrative interfaces and only after successfully identifying and authenticating themselves.

User operations on Data and Metadata are checked for conformance to the granted level of access, and rejected if not conformant.