MarkLogic Security Overview

MD DHR THINK

August 2016

Table of Contents

[I. Introduction 2](#_Toc459198355)

[II. Requirements & Constraints 2](#_Toc459198356)

[III. Recommended Implementation 2](#_Toc459198357)

[a. External LDAP 2](#_Toc459198358)

[b. MarkLogic Configuration 2](#_Toc459198359)

[1. Roles & Compartments 2](#_Toc459198360)

[2. Privileges 2](#_Toc459198361)

[3. Protected Collections 2](#_Toc459198362)

[4. System Users 2](#_Toc459198363)

# Introduction

Provide context and scope for this document here.

# Requirements & Constraints

List requirements and constraints here.

# MarkLogic Security Features

This section provides a brief overview of the security features available in MarkLogic.

## Roles & Privileges

MarkLogic implements a role-based access model with each user being assigned 0 or more roles Each document and/or collection of documents are protected through 4 permissions: read, insert, update and execute that can be assigned to roles. Database functions are secured by privileges which can also be assigned to roles.

A user with roles 1, 2, 3 & 4. The user is capable of performing actions A & B within the database, can read document C, and read or write document E.

Role 3

Role 2

Role 1

Privilege B

Privilege A

Role 4

Update E

Read E

Read C

## Compartments

Compartment Roles is an optional feature that can greatly simplify role management in the case where a system contains multiple tenants and the roles used by each tenant are mostly the same. In the simplest case, imagine a system used by 5 different agencies with each agency having a read/write role. Using the standard role implementation, 10 roles are needed, a read and wite role for each agency. This grows quickly as the number of roles per agency increases. With compartment roles, only 7 roles are needed, a read, write and a role for each agency. If a user needs access to documents from another agency, the user can be assigned the other agency’s compartment role and the user would have the same level of access to the other agency’s documents.

There are limitations to using compartments however. When a user is added to a new compartment, that user will have the same level of access to the new compartment that they have from previous compartments. If a user needs to be able to read or write documents from compartment A and read-only access to compartment B, this does not work and compartments should not be used.

## Authentication & Authorization

MarkLogic access can be broken into 2 components authentication and authorization with each component supporting multiple schemes. Authentication is the means by which a user’s identity is verified and authorization is used to determine what a user can do within the system. Supported combinations of authentication and authorization schemes are described in the table below.

| **Authentication** | **Authorization** | **Details** |
| --- | --- | --- |
| MarkLogic | MarkLogic | Users and roles are maintained and managed within MarkLogic.  Users and roles shared with other applications need to be synced with external directories or applications handle authentication/authorization and access MarkLogic using service accounts. |
| Kerberos | MarkLogic | User authentication is handled using Kereberos tickets with a directory services provider. Authorization is handled within MarkLogic by checking roles assigned to a MarkLogic user that maps to the directory services authenticated user. |
| Kerberos | LDAP | User authentication is handled using Kerberos tickets with a directory services provider. Authorization is handled over LDAP using user group memberships from the directory services provider. MarkLogic relevant user groups in the directory services provider are mapped to MarkLogic roles. |
| LDAP | MarkLogic | User authentication is handled over LDAP with a directory services provider. Authorization is handled within MarkLogic by checking roles assigned to a MarkLogic user that maps to the directory services authenticated user. |
| LDAP | LDAP | User authentication is handled over LDAP with a directory services provider. Authorization is handled over LDAP using user group memberships from the directory services provider. MarkLogic relevant user groups in the directory services provider are mapped to MarkLogic roles. |

# Configuration Recommendations

The recommended approach is to implement role based authentication within MarkLogic with authentication and authorization handled by an external directory service over LDAP. This approach leverages existing users and groups within the organizations, allowing the use of single-sign on. Users can be managed entirely outside of MarkLogic though groups will need to be matched up with MarkLogic roles. Activity within MarkLogic is tracked at the individual user level and if necessary, the data lake/datahub APIs can be exposed for direct user access.

**LDAP**

username/password

username/password

**DJS**

**Applications**

**CHESSIE**

**Applications**

**CHESSIE**

**DJS**

**Data Lake**

username/password

username/password

application authentication

&

authorization

## External LDAP Configuration

While MarkLogic supports both LDAP and Keberos for external authentication, only LDAP is supported for external authorization. If Kerberos was used for authentication, authorization would have to be handled by LDAP or Kerberos users would have to be mapped to internal MarkLogic users.

MarkLogic makes use of the memberOf entity that is implemented in Active Directory and is available to most other directory service schemas as an overlay. When present, this displays a user’s group memberships as a property of the user instead of requiring a lookup of the group. A user is populated with memberOf information after the overlay is installed so preexisting groups will have to be repopulated after the overlay is added.

### LDAP Users

In addition to the basic data lake users, the following users must be present within the directory service.

|  |  |
| --- | --- |
| **Username** | **Description** |
| marklogicLDAP | Service account used by MarkLogic to query LDAP for user authentication and authorization. |

### LDAP Groups

The groups below are provided at the most granular level with each group providing a particular level of access. For example, a DJS user with read-only access to the data lake would be a member of the following groups: DJS, DataLake, DataLakeRead whereas a user with read and write privileges would also have the DataLakeWrite role. As an alternative, it would also be possible to have a group convey multiple roles such as DJS-ReadOnly and DJS-ReadWrite.

| **Group Name** | **Description** |
| --- | --- |
| DataLake | Users that belong to this group have access to content stored within the data lake/datahub. Users without this group cannot get any information from the data lake either through their client application or through any directly exposed data lake APIs. |
| DataLakeRead | Users that belong to this group have read access to non-agency specific records within the data lake as well as any records restricted to any agencies they belong to. Additional Read groups can be added for different record types. |
| DataLakeWrite | Users that belong to this group have write access to non-agency specific records within the data lake as well as any records restricted to any agencies they belong to. Additional Write groups can be added for different record types. |
| DJS | Users that belong to this group have access to DJS restricted records. Their level of access depends on which read and write group(s) they belong to. Additional groups can be defined for different agencies and/or different groups within an agency. |
| FIA | See DJS. |
| CARES | See DJS. |
| CHESSIE | See DJS. |

## MarkLogic Configuration

Aside from some basic internal MarkLogic users, application users will be maintained in an external directory server. Collections and URIs still remain protected by internal MarkLogic roles and privileges and compartments are still assigned to internal MarkLogic roles. Any MarkLogic roles that are available to external users are then mapped to existing LDAP groups. When new roles are required, a corresponding group will be created in LDAP and the new group’s DN will be mapped to the new MarkLogic role.

### Roles

This section lists the proposed roles in MarkLogic, what they are used for and what LDAP group(s) they would map to. This implementation assumes the use of compartment roles.

**Issue:** How should roles be handled for merged documents from two different agencies?

| **Role Name** | **External LDAP Group** | **Description** |
| --- | --- | --- |
| mddhr-deployer | N/A | Internal role for application deployment |
| mddhr-ingest | N/A | Internal role for data ingest and transformation |
| mddhr-manager | N/A | Internal role for MarkLogic setup and configuration |
| mddhr-user | DataLake | Role used restrict access to DataLake. Users must have this role in order to use any DataLake APIs |
| mddhr-read | DataLakeRead | Read access to DataLake records |
| mddhr-write | DataLakeWrite | Write access to DataLake records |
| mddhr-CARES | CARES | Role used to access CARES records |
| mddhr-CHESSIE | CHESSIE | Role used to access CHESSIE records |
| mddhr-DJS | DJS | Role used to access DJS records |
| mddhr-FIA | FIA | Role used to access FIA Records |

### Privileges

Privileges are used to restrict access to MarkLogic app servers, database functions and URIs within the database. Privileges are granted to users through the roles assigned to the user.

| **Privilege Name** | **Privilege Type** | **Description** |
| --- | --- | --- |
| mddhr-datalake-access | execute | Restricts access to DataLake app servers/APIs |

### Protected Collections

One use of collections will be to segregate documents by agency or security restriction. Until more detailed requirements are provided, the list below is provided as an example. The list assumes the use of compartments, but could easily be modified for organizational specific roles eg: mddhr-read and mddhr-CARES would become mddhr-CARES-read, mddhr-write and mddhr-CARES would become mddhr-CARES-write, etc.

| Collection | Read | Insert | Update |
| --- | --- | --- | --- |
| CARES | mddhr-read,  mddhr-CARES | mddhr-write,  mddhr-CARES | mddhr-write,  mddhr-CARES |
| CHESSIE | mddhr-read,  mddhr-CHESSIE | mddhr-write,  mddhr-CHESSIE | mddhr-write,  mddhr-CHESSIE |
| DJS | mddhr-read,  mddhr-DJS | mddhr-write,  mddhr-DJS | mddhr-write,  mddhr-DJS |
| FIA | mddhr-read,  mddhr-FIA | mddhr-write,  mddhr-FIA | mddhr-write,  mddhr-FIA |

### System Users

This section lists MarkLogic users that are maintained within MarkLogic’s internal security database rather than an external directory service. At least one administrator login must be available for system recovery operations and in case the external directory service is not available.

|  |  |
| --- | --- |
| **Username** | **Description** |
| mddhr-deploy-user | Service account used for MarkLogic code/application module deployment |
| mddhr-ingest-user | Service account used for data lake ingest/transformation operations |
| mddhr-manager-user | Service account used for MarkLogic configuration |
| Administrator | Internal administrator login used to manage MarkLogic. |

# Alternative Implementation – Application based Security

A simpler variant of the approach above would be to implement application based security. In this approach, authentication and authorization is handled by the client application with an external directory services via LDAP. The client application then connects to MarkLogic using an application specific service account. The application specific service accounts have all the necessary access needed by any of the application users and it is assumed that any access controls are implemented within the application. Auditing of individual user actions would be tracked by the client application.

application authentication

&

authorization

**LDAP**

username/password

username/password

svcacct.djs

svcacct.chessie

**DJS**

**Applications**

**CHESSIE**

**Applications**

**CHESSIE**

**DJS**

**Data Lake**