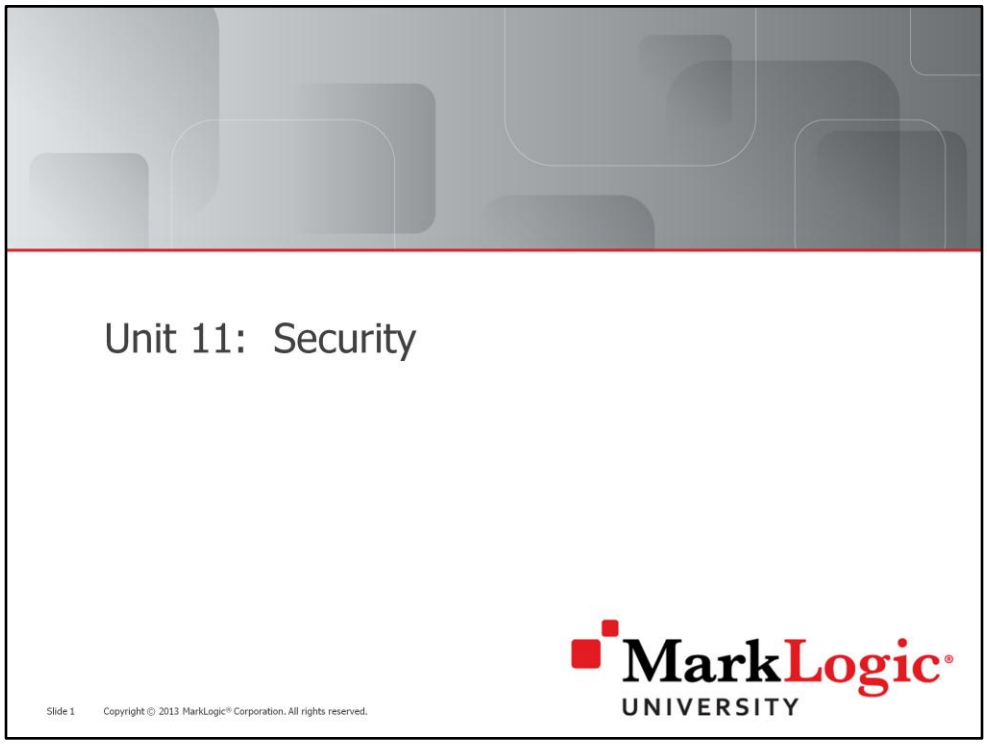


Learning Objectives

- Implement application server authentication
- Create a role
- Create a user
- Manage document permissions
- Set execute privileges
- Create a custom login interface
- Set URI privileges

MarkLogic UNIVERSITY

Security Database

- Contains security settings
- Default database name = "Security"
- Security database can be shared across applications

MarkLogic UNIVERSITY

App Server Authentication

- HTTP & WebDAV app servers support multiple types of authentication

BASIC

- Common
- Requires Login
- PW Masked
- Unencrypted

DIGEST

- Requires Login
- Encrypted

DIGEST-BASIC

- Digest First
- Basic Second
- In case user on old browser

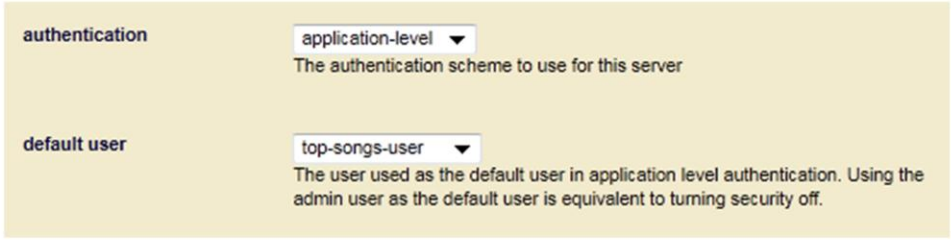
APP LEVEL

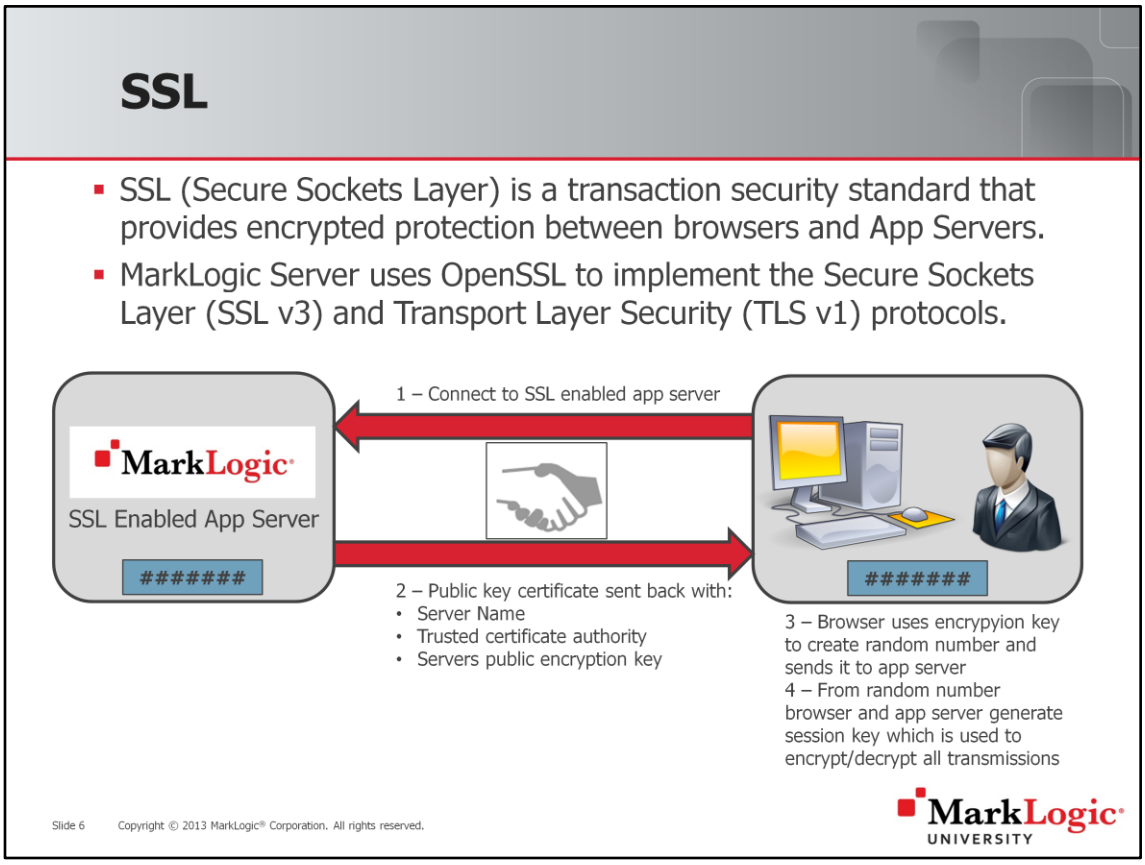
- None
- Authentication handled in App or will use default user

MarkLogic UNIVERSITY

App Server Authentication

- Set through Admin Tool
  - Configure→Groups→Default→App Servers→[*AppServerName*]
- Default User is only for application level authentication

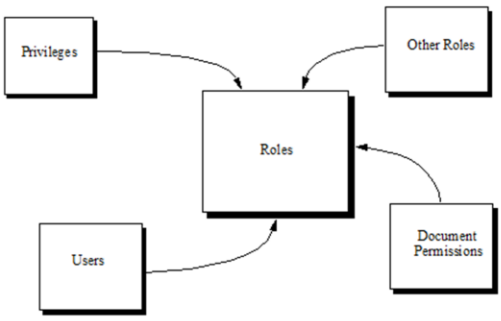




A browser and App Server create a secure HTTPS connection by using a handshaking procedure. When browser connects to an **SSL**-enabled App Server, the App Server sends back its identification in the form of a digital certificate that contains the server name, the trusted certificate authority, and the server's public encryption key. The browser uses the server's public encryption key from the digital certificate to encrypt a random number and sends the result to the server. From the random number, both the browser and App Server generate a *session key*. The session key is used for the rest of the session to encrypt/decrypt all transmissions between the browser and App Server, enabling them to verify that the data didn't change in route.

Role Based Security

- Model is based on roles
- Roles are hierarchical and define:
  - What content users can query
  - What code users can run
  - What locations users can modify (insert, update, delete)



MarkLogic UNIVERSITY

Users

- The primary identifier for authentication
- Assigned 1 : N roles
  - Must have at least one role to do anything
- Defined using Admin Tool
  - Configure→Security→Users
  - Set Password
  - Set Roles
  - Set Default Permissions

MarkLogic UNIVERSITY

Document Permissions

- Document permissions authorize users of particular roles to access particular documents in queries
- Permissions are assigned explicitly through xdmp: functions or implicitly through default settings on roles or users
- **Documents with no document permissions can only be accessed by users with the admin role**
- 3 types of document permission capabilities:

1. Read
2. Update
3. Insert
4. Execute (modules database)

- Every permission consists of a capability and a role

MarkLogic

Document Permissions

```
xdmp:document-set-permissions(
"/songs/The-Doobie-Brothers+Black-Water.xml",
xdmp:permission("top-songs-user-role", "read"))
```


MarkLogic

Execute Privileges

- Authorizes users to run specified pieces of code
- Assigned on a per role basis in the Admin interface
- Security > Roles > Configure (tab)
- MarkLogic comes with a large number of predefined execute privileges
  - See "Appendix B" of the MarkLogic Administrator's Guide for a complete description
- Execute privileges can be applied to custom code

MarkLogic

URI Privileges

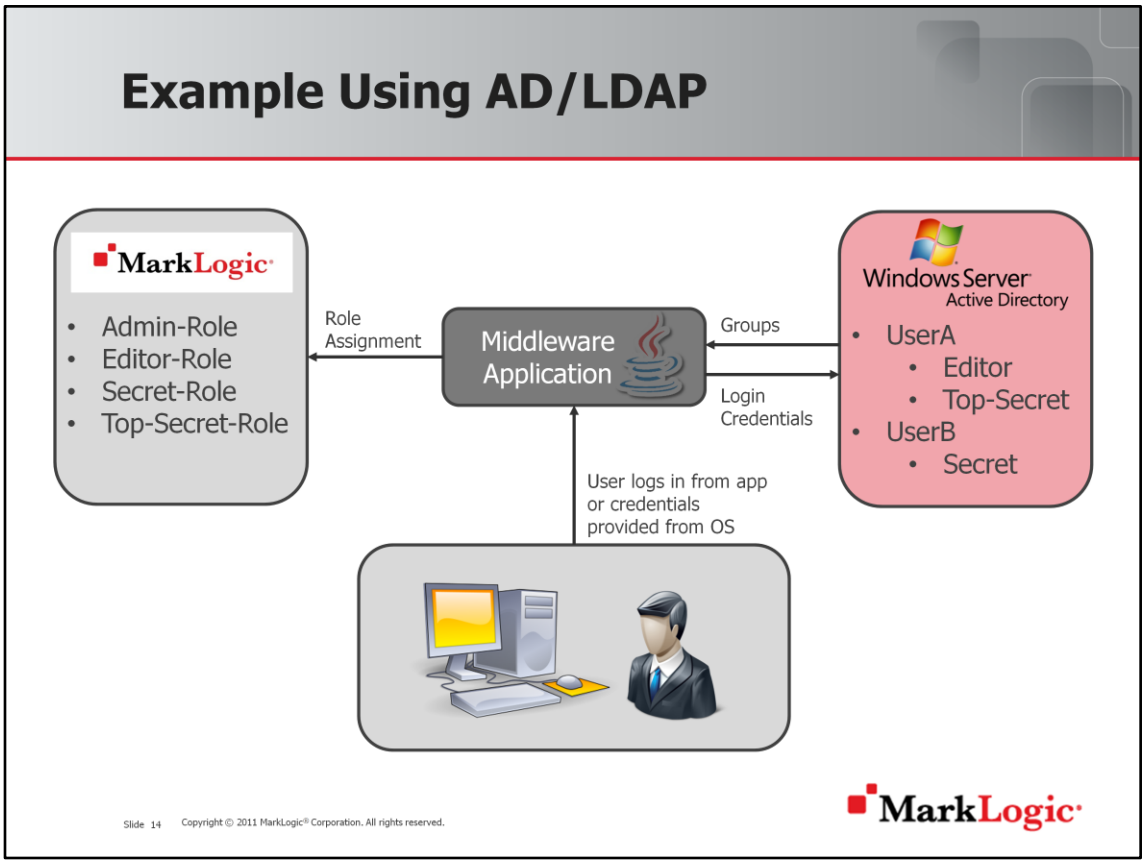
- Authorizes users to create within a specific URI space
- URI space is defined as a URI prefix
  - e.g. Users containing the role "top-songs-uri-priv" can only create new documents whose URIs start with "/songs/"
- All roles must have some sort of URI privilege
  - the any-uri privilege will authorize a role to write to any URI

MarkLogic

Custom Authentication

- A web app written in XQuery can manage its own user authentication:
  - Application-Level authentication must be turned on
  - Application must authenticate users using xdmp:login() and close their session using xdmp:logout()
  - Main modules must check existing roles and change behaviour accordingly
    - e.g. xdmp:get-current-roles()

MarkLogic



Often times customers do not want to manage users inside of MarkLogic because this information is already managed by their implementation of Active Directory. In this scenario, only roles (not users) are defined within MarkLogic. Authentication must occur against Active Directory, and based on the users group membership in Active Directory, an appropriate MarkLogic role can be assigned.

Unit 11: Applying the Learning Objectives

- Implement application server authentication
  - Exercise 1
- Create a role
  - Exercise 2
- Create a user
  - Exercise 3
- Manage document permissions
  - Exercise 4
- Create a custom login interface
- Set URI privileges
  - Exercise 5

MarkLogic UNIVERSITY