# MarkLogic Hardening Procedures

1. **(SV-42806r3\_rule) Configure all of the application systems to require TLS encryption in accordance with data protection requirements.**

MarkLogic application servers are configured to use TLS by creating a digital certificate and associating the application server with the digital certificate. The general steps to do this using the MarkLogic Admin web application are as follows:

* Create a Certificate Template.
* Enable SSL for the App Server.
* Generate a certificate request and send it off to a certificate authority.
* When you receive the signed certificate from the certificate authority, import it into MarkLogic Server for use by your App Server.

Detailed instructions can be found in the “Configuring SSL on App Servers” chapter in the MarkLogic Server Security Guide, <https://docs.marklogic.com/guide/security/SSL>

1. **Remove embedded authentication data stored in code, configuration files, scripts, HTML file, or any ASCII files.**

This is the responsibility of the developers of the application. From a MarkLogic perspective, there is no need for authentication data to be stored in the files types listed. A recommended approach to configuration files that need to use and/or create passwords is to store the passwords in an encrypted file. Then the configuration file decrypts the password file based on a user entered key at runtime.

1. **(SV-72499r1\_rule, SV-72513r1\_rule, SV-72515r1\_rule) Configure the application to create an audit record for both successful and unsuccessful attempts to access security objects.**

The first step for creating any audit record is to ensure that auditing is enabled at the group level in the MarkLogic cluster. This can be accomplished via the Admin Interface in a browser. The general steps to do this are as follows:

* Access the Admin Interface with a browser.
* Open the Audit Configuration screen (Groups > group\_name > Auditing).
* Select True for the Audit Enabled radio button.
* Configure any audit events and/or audit restrictions you want.
* Click OK.

Detailed instructions can be found in the MarkLogic Administrator’s Guide, <https://docs.marklogic.com/guide/admin/auditing#id_68339>

Once auditing is enabled, specific audit events should be enabled. For this requirement, the “security-access” event should be enabled. This generates audit events when one of the following security-related functions are called: xdmp:can-grant-roles, xdmp:has-privilege, xdmp:user-roles, xdmp:role-roles, xdmp:privilege-roles, xdmp:amp-roles, xdmp:get-current-role, xdmp:user, xdmp:role, xdmp:amp.

1. **(SV-72509r2\_rule, SV-72511r2\_rule, SV-72521r1\_rule) Configure the application to create an audit record for both successful and unsuccessful attempts to delete database security objects.**

The user-role-removal audit event will produce this for failure and successes. Ref: MarkLogic SysAdmin guide, page 109. Ref: MarkLogic SysAdmin guide, page 110.

After enabling auditing as in item #3, ensure that the audit event, “security-access” is enabled.

1. **(SV-42711r3\_rule) Configure the application to log the identity of the user and/or the process associated with the event.**

The identity of the user that established the connection is automatically included in all audit log entries. No additional configuration is possible.

1. **Utilize and implement data mining protections when requirements specify it.**

* **Limiting the types of responses provided by database queries** – The default search results include information about the URI of the document and what terms matched the search. If those results are unacceptable, the results may be customized to include as much or as little information as is required.
* **Limiting the frequency of queries** – The number of concurrent requests may be limited by setting the “concurrent request limit” setting for the application server on the MarkLogic Admin web page.
* **Audit log creation of unusual queries (this would require an understanding of what queries are typical)** – This is not possible out of the box. Custom query handling would need to be implemented to generate audit logs based on the types of queries.

1. **Configure the application database to utilize transactional logging.**

The journaling setting is specific to each database. By default, journaling is set to the “fast” mode. With that setting, the journal protects against MarkLogic Server process failures but not against host operating system kernel failures or host hardware failures. To protect against host failures, set the journaling mode to “strict”.

1. **(SV-42908r5\_rule) Configure the application to not display technical details about the application architecture on error events.**

The use-case under consideration involves a middle-tier application that uses the MarkLogic Java client. Without administrative-level privileges, the Java client does not have access to application architecture information. Additionally, the error events are Java exceptions. Those exceptions should be handled by the middle-tier and are generic in nature. Exceptions such as ResourceNotFoundException are generated by the Java client.

1. **Disable unnecessary built-in userids, use other strong authentication when possible and use strong passwords if accounts are necessary for application operation.**

There are four built-in userids. Two of those may be deleted. Deletion may be performed using the Admin interface. Those users are as follows:

* “healthcheck” - The HealthCheck application server uses this user as the default. Therefore, you will need to delete that application server, or assign a different default user, before deleting this user.
* “infostudio-admin”

The other two users are necessary for application operation. The “admin” user is the super-user. An authorized administer must exist to properly configure, maintain, and operate the system.

The “nobody” user is a user with minimal roles assigned. This user is required to permit users to connect to a MarkLogic application server using LDAP authentication. When that connection is established, the user is connected as “nobody". After LDAP authentication and authorization, the proper roles are granted to the user.

MarkLogic server passwords are completely user configurable. Therefore, it is the responsibility of the system administrators to use acceptable passwords for the two required users.

1. **(SV-42851r3\_rule) Configure the application server so that admin management functionality and hosted applications are separated.**

MarkLogic is configured this way out of the box. The Admin interface is hosted on port 8001 by default. Application services, such as those provided by a REST server for the MarkLogic Java client, are hosted on application-specific ports.

1. **(SV-42474r3\_rule) Database management includes the ability to control the number of users and user sessions utilizing a DBMS. Unlimited concurrent connections to the DBMS could allow a successful Denial of Service (DoS) attack by exhausting connection resources; and a system can also fail or be degraded by an overload of legitimate users. Limiting the number of concurrent sessions per user is helpful in reducing these risks. This requirement addresses concurrent session control for a single account. It does not address concurrent sessions by a single user via multiple system accounts; and it does not deal with the total number of sessions across all accounts. The capability to limit the number of concurrent sessions per user must be configured in or added to the DBMS (for example, by use of a logon trigger), when this is technically feasible. Note that it is not sufficient to limit sessions via a web server or application server alone, because legitimate users and adversaries can potentially connect to the DBMS by other means. The organization will need to define the maximum number of concurrent sessions by account type, by account, or a combination thereof. In deciding on the appropriate number, it is important to consider the work requirements of the various types of users. For example, 2 might be an acceptable limit for general users accessing the database via an application; but 10 might be too few for a database administrator using a database management GUI tool, where each query tab and navigation pane may count as a separate session. (Sessions may also be referred to as connections or logons, which for the purposes of this requirement are synonyms.)**

MarkLogic application servers have a configurable setting to control the number of concurrent requests made by a user. This setting allows the system to regulate user requests across applications. To configure a user concurrent session limit, perform the following steps in the Admin Interface:

* Click the Groups icon.
* Click the group in which the App Server you want to configure resides (for example, Default).
* Click the App Servers icon on the left tree menu.
* Select the App Server in which you want to configure concurrent session limits. The App Server Configuration page displays.
* In the concurrent request limit field, enter a value corresponding to the maximum number of concurrent user sessions you want to allow. For example, if you want only 3 concurrent sessions, enter 3. A value of 0 means there is no concurrent request limit (unlimited).
* Click OK to save the configuration change.

For additional information, please see <https://docs.marklogic.com/guide/admin/session-login>

1. **(SV-42509r3\_rule) Integrate DBMS security with an organization-level authentication/access mechanism providing account management for all users, groups, roles, and any other principals. (should be covered by task 1)**

MarkLogic 9 provides multiple options for using an organization-level authentication/access control mechanism. These include LDAP, Kerberos, and SAML. The example project delivered for task 1 demonstrates how to integrate MarkLogic with an LDAP server. In that project, MarkLogic is configured to get all authentication and authorization information from the LDAP server. No organizational users need to be defined or maintained in MarkLogic. As described in STIG requirement 9 above, only the “admin” and “nobody” users are needed for this example.

For additional information, please see the Chapter 10, External Security, of the MarkLogic Security Guide at <https://docs.marklogic.com/guide/security/external-auth>

1. **(SV-42520r3\_rule) Configure the DBMS settings and access controls to permit user access only to objects and data that the user is authorized to view or interact with, and to prevent access to all other objects and data. (should be covered by task 2)**

In a MarkLogic database, assigning permissions to data will protect the data. During system design and implementation, care should be taken to ensure that a reliable plan is in place to properly allocate roles, and to assign access permissions on data based on those roles. This capability was demonstrated in the example project delivered for task 1. In that project, roles were created that mapped directly to LDAP user groups. Then those roles are granted specific permissions on specific data. That design permits users to access data they are authorized to view, while also protecting data that they are not permitted to view.

1. **(SV-42684r4\_rule) Use accounts assigned to individual users. Where the application connects to the DBMS using a standard, shared account, ensure that it also captures the individual user identification and passes it to the DBMS. Modify application database tables and all supporting code to capture the necessary audit data.**

This requirement is not a MarkLogic configuration task, but is more of an organizational policy. The system administrator will need to enforce this policy by ensuring that only required accounts that fit with organizational policy are created on the MarkLogic server. If external security is used as in the example project from task 1, then this requirement becomes trivial.

1. **(SV-42699r3\_rule) Configure the DBMS to generate audit records for at least the DoD minimum set of events.**

Configurable - Auditing can be configurable, need to know what the minimum set of auditable events are to know if they are supported by the server

1. **(SV-42700r3\_rule) Configure the DBMS's settings to allow designated personnel to select which auditable events are audited.**

Once auditing is enabled as described in STIG requirement 3 above, individual auditable events may be selected. The following is the general procedure for configuring audit events and audit restrictions. Your procedure will vary depending on what events and restrictions you choose to configure.

Access the Admin Interface with a browser.

Open the Audit Configuration screen (Groups > group\_name > Auditing).

Under Audit Events, choose the events you want audited. For a description of each event, see [Auditable Events](https://docs.marklogic.com/guide/admin/auditing#id_15948).

Under Audit Restrictions, enter any restrictions you want. For details on audit restrictions, see [Restricting Audit Events](https://docs.marklogic.com/guide/admin/auditing#id_23562).

Click OK to save your changes.

For additional information, please see the Chapter 12, Auditing, of the MarkLogic Administrator’s Guide at <https://docs.marklogic.com/guide/admin/auditing>

1. **(SV-42701r3\_rule) If currently required, configure the DBMS to produce audit records when privileges/permissions/role memberships are retrieved.**

This requirement is covered by the implementation of item #3 above.

1. **(SV-42702r2\_rule, SV-72537r1\_rule) Configure the DBMS software or third-party product to enable session auditing.**

Configurable - Audit-startup is enabled and monitors the audit system. Ref: MarkLogic SysAdmin guide, page 106.

1. **(SV-42703r3\_rule) Deploy a DBMS capable of capturing, recording, and logging all content related to an established user session, or acquire a third-party application to perform this function.**

Configurable - Auditing is configured. Note: Need to turn on the LAST USER LOGIN. (page 116). Ref: MarkLogic SysAdmin guide, page 117.

1. **(SV-42705r3\_rule) Configure DBMS audit settings to include event type as part of the audit record.**

Inherent - When auditing the entry contains the event type. Ref: MarkLogic SysAdmin guide, page 110.

1. **(SV-42706r3\_rule) Configure DBMS audit settings to include the date and time of the occurrence of the event as part of the audit record.**

Inherent - When enabled auditing log content is configured with time stamps. Ref: MarkLogic SysAdmin guide, page 110.

1. **(SV-42707r3\_rule) Configure DBMS audit settings to include where the event occurred as part of the audit record.**

Inherent - When auditing is configured the location of the event is recorded. Ref: MarkLogic SysAdmin guide, page 110.

1. **(SV-42708r3\_rule) Configure DBMS audit settings to include the source of the event as part of the audit record.**

Inherent - When auditing is configured the source of the event is recorded. Ref: MarkLogic SysAdmin guide, page 110.

1. **(SV-42710r3\_rule) Configure DBMS audit settings to include the outcome of the event as part of the audit record.**

Inherent - When auditing is configured the outcome of the event is recorded. Ref: MarkLogic SysAdmin guide, page 110.

1. **(SV-42711r3\_rule) Configure DBMS audit settings to include user name as part of the audit record.**

Inherent - When auditing is configured the user of the event is recorded. Ref: MarkLogic SysAdmin guide, page 110

1. **(SV-42712r4\_rule) Configure DBMS audit settings to include all organization-defined detailed information in the audit records for audit events identified by type, location, or subject.**

Configurable - Auditing can be configurable. We would need to know what organization-defined detailed information is required in order to know if the server supports it.

1. **(SV-42720r3\_rule) Configure the system to shut down, rolling back all in-flight transactions, in the case of an auditing failure.**

MarkLogic does not have any built-in capability to implement this requirement. However, the most likely cause of an auditing failure would be in the event of a storage failure. The best way to handle this situation would be to use third-party tools to monitor the storage device(s). If a problem is detected, initiate MarkLogic shutdown. That would automatically roll back all in-flight transactions.

1. **(SV-42728r3\_rule) Configure the DBMS to use time stamp values obtained from or synchronized with the internal system clock used by the operating system.**

Inherent -MarkLogic uses the local time synchronized to the local OS system timestamp.

1. **(SV-42779r4\_rule) Configure DBMS settings to uniquely identify and authenticate all organizational users who log on/connect to the system.**

All users are uniquely identified.

The example project delivered for task 1 demonstrates how to integrate MarkLogic with an LDAP server. In that project, MarkLogic is configured to get all authentication and authorization information from the LDAP server.

1. **(SV-42805r3\_rule) Record whether they do or do not contain DBMS passwords. If passwords are present, ensure that they are correctly hashed using one-way, salted hashing functions, and that the hashes are protected by host system security.**

Configurable - Configure LDAP / Kerberos for authentication to the admin consoles. Ref: MarkLogic SysAdmin guide, page 290. MarkLogic Security Guide, page 32-34

1. **(SV-42806r3\_rule) Configure encryption for transmission of passwords across the network. If the database does not provide encryption for logon events natively, employ encryption at the OS or network level.**

Configurable - Configure LDAP / Kerberos for authentication to the admin consoles. Ref: MarkLogic SysAdmin guide, page 290. MarkLogic Security Guide, page 32-34

All access to the MarkLogic server can be configured to run on SSL/TLS enabled ports. Additionally, digest authentication may be employed for HTTP ports. However, when using an LDAP server for authentication/authorization, basic authentication must be used so that the password may be forwarded to the LDAP server. In that case, the encryption must be done using SSL/TLS.

1. **(SV-42812r3\_rule) Configure the DBMS to validate certificates by performing RFC 5280-compliant certification path validation.**

Configurable - Configure LDAP / Kerberos for authentication to the admin consoles. Ref: MarkLogic SysAdmin guide, page 290. MarkLogic Security Guide, page 32-34

MarkLogic stores and uses x.509 certificates for SSL/TLS layers. The certificates are validated using Certificate Authorities registered with the MarkLogic server. New certificates may be imported or created and registered with one of the registered CAs. Additional details may be found in Chapter 8, Configuring SSL on App Servers, in the MarkLogic Security Guide and at https://docs.marklogic.com/guide/security/SSL.

1. **(SV-42813r3\_rule) Store all DBMS PKI private keys in a FIPS 140-2 validated cryptographic module. Ensure access to the DBMS PKI private keys is restricted to only authenticated and authorized users.**

Configurable - Make sure MarkLogic Server runs in FIPS mode (which is the default model). MarkLogic Security Guide, page 70.

1. **(SV-42815r3\_rule) Configure the DBMS to map the authenticated identity directly to the DBMS user account.**

Configurable - Configure LDAP / Kerberos for authentication to the admin consoles. Ref: MarkLogic SysAdmin guide, page 290. MarkLogic Security Guide, page 32-34.

1. **(SV-42816r3\_rule) Modify and configure each non-compliant application, tool, or feature associated with the DBMS/database so that it does not display authentication secrets.**

Configurable - Configure LDAP / Kerberos for authentication to the admin consoles. Ref: MarkLogic SysAdmin guide, page 290. MarkLogic Security Guide, page 32-34.

1. **(SV-42817r3\_rule) Utilize NIST FIPS 140-2 validated cryptographic modules for all cryptographic operations.**

Configurable - Make sure MarkLogic Server runs in FIPS mode (which is the default model). MarkLogic Security Guide, page 70.For this system, MarkLogic will not be performing any cryptographic operations.

1. **(SV-42860r2\_rule) Configure DBMS settings to terminate sessions, invalidating their session identifiers, upon user logout.**

This requirement will primarily be handled by client applications. It will be incumbent upon the client application to close all user sessions upon user logout.

1. **(SV-42865r3\_rule) Configure DBMS settings so that, in the event of a system failure, the DBMS will roll back open transactions to a consistent state, to include a security configuration that is at least as restrictive as before the system failure.**

The database supports multiple types of rollback procedures in the event of system failure. Database failures are secure. Ref: MarkLogic SysAdmin guide, page 268.

The MarkLogic database is ACID-compliant. By default, open transactions are rolled-back in the event of a system failure and the database is left in consistent state. In MarkLogic, updated documents are initially saved to memory (in an in-memory stand). Transactions are not committed until the transaction is written to the on-disk journal for the database and can be replayed in the event of a failure. For a more in-depth discussion of this topic, please see <https://developer.marklogic.com/blog/how-marklogic-supports-acid-transactions#sthash.REvVXyIl.dpuf>

1. **(SV-42866r3\_rule) Configure DBMS settings to preserve any organization-defined system state information in the event of a system failure.**

Since MarkLogic is ACID-compliant, all data stored in MarkLogic is preserved in the event of a recoverable system failure. However, a reliable backup policy should be put in place to prevent the loss of information in the event of an unrecoverable storage failure.

1. **(SV-42871r3\_rule) Apply appropriate controls to protect the confidentiality and integrity of data at rest in the database.**

MarkLogic 9 includes the ability to encrypt data at rest. Encryption can be applied to newly created files, configuration files, or log files. Existing data files can be encrypted by triggering a merge or re-index of the data. For more information, please see Chapter 11, Encryption at Rest, in the MarkLogic Security Guide at <https://docs.marklogic.com/guide/security/encryption>.

1. **(SV-42873r3\_rule) Locate security-related database objects and code in a separate database, schema, or other separate security domain from database objects and code implementing application logic.**

MarkLogic Server uses a security database to store the user data, privilege data, role data, and other security information. Each database in MarkLogic Server references a security database. A database named Security, which functions as the default security database, is created as part of the installation process. Ref: MarkLogic Security Guide, page 9.

1. **(SV-72449r1\_rule) Implement the organization's DAC policy in the security configuration of the database and DBMS, and, if applicable, the security configuration of the application(s) using the database.**

MarkLogic uses DAC and Role Based Access Control. (RBAC). Ref: MarkLogic Security guide, page 9, 32 and 49. Common Criteria Report, page 9.

1. **(SV-72453r1\_rule) Configure DBMS security to protect all privileged functionality.**

Configurable and Inspection - There are four capabilities in the MarkLogic server: read, insert, update and execute. Access to these capabilities is controlled through a user’s role. Roles should be reviewed to verify the effective permissions. By default, all privileged functionality is restricted to MarkLogic users with “administrator” privileges. The system administrators will need to ensure that they grant only the necessary roles to users.

1. **(SV-72455r1\_rule) Configure the DBMS to automatically terminate a user session after organization-defined conditions or trigger events requiring session termination.**

This requirement will primarily be handled by client applications. The client applications will need to maintain a list of user sessions and, when conditions are met, close the relevant sessions.

1. **(SV-72465r1\_rule) Where relevant, modify the configuration to allow the user to manually terminate a session initiated by that user.**

This requirement will also need to be implemented by the client applications. Generally, in three-tier applications, the users do not have direct access to the MarkLogic database.

For the built-in web applications (admin and qconsole), this is an open requirement. There are no logout buttons for those applications.

1. **(SV-72467r1\_rule) Enable DBMS features, deploy third-party software, or add custom data structures, data elements and application code, to provide reliable security labeling of information in storage.**

MarkLogic does not provide any tools specifically for security labeling. However, the database is flexible enough to be able to work with just about any labeling system required. MarkLogic has many customers that follow DoD security markings standards. Additionally, element-level security and redaction can be used to prevent unauthorized access.

1. **(SV-72469r1\_rule) Enable DBMS features, deploy third-party software, or add custom data structures, data elements and application code, to provide reliable security labeling of information in process.**

Please see the response to item #46.

1. **(SV-72471r1\_rule) Enable DBMS features, deploy third-party software, or add custom data structures, data elements and application code, to provide reliable security labeling of information in transmission.**

Please see the response to item #46.

1. **(SV-72479r1\_rule) Configure and/or deploy software tools to ensure that DBMS audit records are written directly to or systematically transferred to a centralized log management system.**

Centralized management can collect all database audit data. MarkLogic logs, including audit records, are written to a directory under the default data directory. The organization should develop a plan for ensuring long-term maintenance of the logs. This is frequently implemented using sym-links that point to directories monitored by a centralized log management system.

1. **(SV-72481r1\_rule) Deploy a DBMS that provides a unified tool for audit configuration.**

MarkLogic audit configuration is done via the MarkLogic web-based administration tool. Detailed instructions can be found in the MarkLogic Administrator’s Guide, [https://docs.marklogic.com/guide/admin/auditing](https://docs.marklogic.com/guide/admin/auditing#id_68339)

1. **(SV-72589r1\_rule) Implement NIST FIPS 140-2 validated cryptographic modules to provision digital signatures.**

For this system, MarkLogic will not be provisioning digital signatures.

1. **(SV-72591r1\_rule) Implement a NIST FIPS 140-2 validated cryptographic module in the DBMS for generation and verification of cryptographic hashes.**

For this system, MarkLogic will not be generating or verifying any cryptographic hashes.

1. **(SV-42764r3\_rule) Disable use of or remove any external application executable object definitions that are not authorized.**

MarkLogic embeds several third-party products in the server application. The server is not configured for removal or disabling of those products. If any of these products are determined to be unauthorized, client applications should be designed to not make use of those products. For more information regarding combined products, please refer to: http://docs.marklogic.com/guide/copyright/legal.

1. **(SV-42734r3\_rule) The DBMS must protect its audit features from unauthorized access.**

By default users must have administrative privileges to view or manipulate MarkLogic audit features. Care should be taken to prevent administrative access to the MarkLogic Admin web page.

Additionally, audit events are stored in log files. Those log files must be protected by the system administrator.

1. **(SV-42735r3\_rule) The DBMS must protect its audit configuration from unauthorized modification.**

Please see the response to item #54.

1. **(SV-42736r3\_rule) The DBMS must protect its audit features from unauthorized removal.**

Please see the response to item #54.

1. **(SV-42749r3\_rule) Database objects (including but not limited to tables, indexes, storage, stored procedures, functions, triggers, links to software external to the DBMS, etc.) must be owned by database/DBMS principals authorized for ownership.**

Configurable and Manual Review - Accounts/Users on the system can be listed and their ownership reviewed for document permissions. Ref: MarkLogic SysAdmin guide, page 289.

Database objects within MarkLogic may have read and/or update permissions set for specific roles. If a document has update permissions set to a specific role, then the users that have been granted that role, are the effective owners of the document.

1. **(SV-42751r3\_rule) The DBMS software installation account must be restricted to authorized users.**

There is no software installation account. Server is LINUX and root account used to install application. The system administrators will enforce this requirement. They should ensure that only authorized users have the necessary privileges on the servers to install or update software.

1. **(SV-42752r3\_rule) The DBMS must limit privileges to change software modules, to include stored procedures, functions and triggers, and links to software external to the DBMS.**

MarkLogic stored procedures, functions, and triggers are stored in “modules” (code) databases within MarkLogic. Access to those modules databases need to be protected in the same manner as data. Specifically, do not permit non-administrative users to write to those databases.

1. **(SV-42760r3\_rule) Default demonstration and sample databases, database objects, and applications must be removed.**

Inherent - No demonstration or sample databases, database applications, objects or files are provided with MarkLogic. However, there is an empty database, “Documents” that is installed. If that database is not used for the system, then it should be removed.

1. **(SV-42761r3\_rule) Unused database components, DBMS software, and database objects must be removed.**

Configurable - There are two unused databases that are installed by default and which can be removed (along with their associated Forests): Documents and Modules.

1. **(SV-42763r4\_rule) Unused database components that are integrated in the DBMS and cannot be uninstalled must be disabled.**

As stated in item #61, any of the database components may be removed. However, the risk is that a feature that could be necessary in the future would be lost. A second approach might be to disable any database that is currently unused, leaving the option to re-enable the database if the need arises.

1. **(SV-42765r3\_rule) The DBMS must be configured to prohibit or restrict the use of organization-defined functions, ports, protocols, and/or services, as defined in the PPSM CAL and vulnerability assessments.**

This will be a system administrator task. They should ensure that all services deployed to the MarkLogic are configured within the PPSM CAL and pass vulnerability assessments.

1. **(SV-42818r3\_rule) The DBMS must uniquely identify and authenticate non-organizational users (or processes acting on behalf of non-organizational users).**

As in the example project developed for task 1, all users are authenticated via LDAP. If the non-organizational users are also authenticated via LDAP, then no additional configuration is necessary. If non-organizational users are created directly in the database, then the system administrator should ensure that each user is uniquely identified by their credentials in the database.

1. **(SV-42863r2\_rule) The DBMS must recognize only system-generated session identifiers.**

Configurable - Configure TLS on all web pages. Ref: MarkLogic SysAdmin guide, page 78. MarkLogic Security Guide, page 32-34.

1. **(SV-42884r3\_rule) Database contents must be protected from unauthorized and unintended information transfer by enforcement of a data-transfer policy.**

By default, only MarkLogic users with administrative privileges have access to database contents. The responsibility for meeting this requirement will be on the system architect and the system administrator. The architect will need to design a data-transfer policy that protects against unauthorized and unintended information transfer by limiting these activities to authorized users. The administrator will need to enforce that policy by assigning roles to users that reflect the policy.

1. **(SV-42907r3\_rule) The DBMS must provide non-privileged users with error messages that provide information necessary for corrective actions without revealing information that could be exploited by adversaries.**

Within the database, there are no non-privileged users. Outside the database, the information provided to non-privileged users in error messages will be managed by the client applications.

1. **(SV-72451r1\_rule) Execution of software modules (to include stored procedures, functions, and triggers) with elevated privileges must be restricted to necessary cases only.**

Configurable and Inspection - There are four capabilities in the MarkLogic server: read, insert, update and execute. Access to these capabilities is controlled through a user’s role. Roles should be reviewed to verify the effective permissions. MarkLogic software modules will only be executed using normal privileges unless additional privileges are granted. Therefore, it is up to the system developers and the system administrators to ensure additional privileges are not granted unless necessary.

1. **(SV-72483r1\_rule) The DBMS must allocate audit record storage capacity in accordance with organization-defined audit record storage requirements.**

MarkLogic does not allocate audit record storage. Audit records are stored in log files on system storage. The system administrator must ensure that an appropriate amount of disk space is available for the audit log files. Additionally, the system administrator must ensure that a reliable rollover policy is in-place. Finally, if long-term access to audit records is required, the system administrator should configure an automated mechanism for moving the audit log files to external storage.

1. **(SV-72493r1\_rule) The DBMS must record time stamps, in audit records and application data, that can be mapped to Coordinated Universal Time (UTC, formerly GMT).**

All access logs have time stamps that map to UTC, Error and Audit logs have the timestamp in the time zone of the OS. Configure the OS to run at UTC. Please see the response to items #21 & #28.

1. **(SV-72495r1\_rule) The DBMS must generate time stamps, for audit records and application data, with a minimum granularity of one second**

MarkLogic server creates time stamps with a granularity of milliseconds.

1. **(SV-72497r1\_rule) The DBMS must provide the means for individuals in authorized roles to change the auditing to be performed on all application components, based on all selectable event criteria within organization-defined time thresholds.**

Inherent - Authorized users can reconfigure auditing at any time and takes effect immediately. MarkLogic 8 AdminGuide page 106.

1. **(SV-72501r2\_rule) The DBMS must generate audit records when privileges/permissions are added.**

After enabling auditing as in item #3, ensure that the audit event, “user-role-addition” is enabled. Ref: MarkLogic SysAdmin guide, page 110.

1. **(SV-72503r2\_rule) The DBMS must generate audit records when unsuccessful attempts to add privileges/permissions occur.**

MarkLogic does not generate audit records for this event. In general, this attempts to add privileges/permissions only fail when the privileges/permissions do not exist, or the user does not the authority to add them.

1. **(SV-72505r2\_rule) The DBMS must generate audit records when privileges/permissions are modified.**

The user-role-addition audit event will produce this for failure and successes. Ref: MarkLogic SysAdmin guide, page 110.

1. **(SV-72507r2\_rule) The DBMS must generate audit records when unsuccessful attempts to modify privileges/permissions occur.**

The user-role-addition audit event will produce this for failure and successes. Ref: MarkLogic SysAdmin guide, page 109.

1. **(SV-72517r1\_rule) The DBMS must generate audit records when security objects are modified.**

The security-access audit event will produce this for failure and successes. Ref: MarkLogic SysAdmin guide, page 109.

1. **(SV-72519r1\_rule) The DBMS must generate audit records when unsuccessful attempts to modify security objects occur.**

The security-access audit event will produce this for failure and successes. Ref: MarkLogic SysAdmin guide, page 109.

1. **(SV-72525r1\_rule) The DBMS must generate audit records when categories of information (e.g., classification levels/security levels) are accessed.**

MarkLogic does not automatically recognize different categories of information. As such, no audit records are generated for this type of event.

1. **(SV-72527r1\_rule) The DBMS must generate audit records when unsuccessful attempts to access categories of information (e.g., classification levels/security levels) occur.**

Please see the response to item #79.

1. **(SV-72529r1\_rule) The DBMS must generate audit records when categories of information (e.g., classification levels/security levels) are modified.**

Please see the response to item #79.

1. **(SV-72531r1\_rule) The DBMS must generate audit records when unsuccessful attempts to modify categories of information (e.g., classification levels/security levels) occur.**

Please see the response to item #79.

1. **(SV-72533r1\_rule) The DBMS must generate audit records when categories of information (e.g., classification levels/security levels) are deleted.**

Please see the response to item #79.

1. **(SV-72535r1\_rule) The DBMS must generate audit records when unsuccessful attempts to delete categories of information (e.g., classification levels/security levels) occur.**

Please see the response to item #79.

1. **(SV-72539r1\_rule) The DBMS must generate audit records when unsuccessful logons or connection attempts occur.**

Configurable - Audit configuration can be configured to collect unsuccessful logons or connection attempts occur. MarkLogic 8 Admin Guide page 106.

After enabling auditing as in item #3, ensure that the audit events, “authentication-failure” and “external-authentication-failure” are enabled.

1. **(SV-72541r1\_rule) The DBMS must generate audit records showing starting and ending time for user access to the database(s).**

MarkLogic does not generate audit records specifically for start and end times. These actions should be tracked by the client applications.

1. **(SV-72543r1\_rule) The DBMS must generate audit records when concurrent logons/connections by the same user from different workstations occur.**

MarkLogic does not generate audit records specifically for start and end times. These actions should be tracked by the client applications.

1. **(SV-72547r1\_rule) The DBMS must generate audit records when unsuccessful attempts to execute privileged activities or other system-level access occur**

After enabling auditing as in item #3, ensure that the audit events, “no-privilege” and “no-permission” are enabled.

1. **(SV-72555r1\_rule) The DBMS must enforce access restrictions associated with changes to the configuration of the DBMS or database(s).**

By default, the configuration of the DBSMS and its databases is considered privileged functionality and is restricted to MarkLogic users with “administrator” privileges.

1. **(SV-72557r1\_rule) The DBMS must produce audit records of its enforcement of access restrictions associated with changes to the configuration of the DBMS or database(s).**

After enabling auditing as in item #3, ensure that the audit event, “configuration-change” is enabled.

1. **(SV-72565r1\_rule) The role(s)/group(s) used to modify database structure (including but not necessarily limited to tables, indexes, storage, etc.) and logic modules (stored procedures, functions, triggers, links to software external to the DBMS, etc.) must be restricted to authorized users.**

Non-privileged users do not have the ability to install software.

1. **(SV-72561r1\_rule) The DBMS must be configured in accordance with the security configuration settings based on DoD security configuration and implementation guidance, including STIGs, NSA configuration guides, CTOs, DTMs, and IAVMs.**

This document satisfies this requirement.

1. **(SV-72567r1\_rule) The DBMS must prohibit the use of cached authenticators after an organization-defined time period.**

When external security is configured, the “cache timeout” setting should be set to the organization-defined time period. This will cause the DBMS to re-authenticate with the LDAP server.

1. **(SV-72577r1\_rule) The DBMS must require users to re-authenticate when organization-defined circumstances or situations require re-authentication.**

This requirement will primarily be handled by client applications. It will be incumbent upon the client application to force users to re-authenticate when circumstances or situations require.

The MarkLogic database will force users to reauthenticate when authentication is set to digest or digestbasic, however we are using LDAP authentication

1. **(SV-72579r1\_rule) The DBMS must prevent unauthorized and unintended information transfer via shared system resources.**

MarkLogic protects prevent unauthorized and unintended information transfer via shared system resources by use of the security database. Please see the response to items #67. Additionally, the system administrator should also ensure that log files are kept in a protected area of the file system. This will help prevent unauthorized users from accessing the log files.

1. **(SV-72581r1\_rule) Access to database files must be limited to relevant processes and to authorized, administrative users.**

It will be the responsibility of the system administrators to ensure that MarkLogic processes and files are properly protected from unauthorized access.

1. **(SV-72593r1\_rule) The DBMS must implement NIST FIPS 140-2 validated cryptographic modules to protect unclassified information requiring confidentiality and cryptographic protection, in accordance with the data owners requirements.**

For this system, MarkLogic will not be encrypting data at rest.

1. **(SV-72601r1\_rule) The DBMS must implement cryptographic mechanisms preventing the unauthorized disclosure of organization-defined information at rest on organization-defined information system components.**

For this system, MarkLogic will not be encrypting data at rest.

1. **(SV-72603r1\_rule) The DBMS must maintain a separate execution domain for each executing process.**

Inherent - MarkLogic isolates execution domains via individual transactions MarkLogic handles all incoming requests individually in their own workspace. Requests may not interact with each other.

1. **(SV-72605r1\_rule) When updates are applied to the DBMS software, any software components that have been replaced or made unnecessary must be removed.**

New install. It is standard practice to review the post-installation and verify older versions have been removed.When MarkLogic is updated, only the binaries are replaced. The databases remain as they were. No new components are added.

1. **(SV-72609r2\_rule) The DBMS and associated applications must reserve the use of dynamic code execution for situations that require it.**

This requirement will need to be enforced by the whole team. The developers of any code that is deployed to MarkLogic need to be instructed on the proper use of dynamic code. The security team needs to be able to recognize and flag dynamic code execution points.

1. **(SV-72611r1\_rule) The DBMS and associated applications, when making use of dynamic code execution, must scan input data for invalid values that may indicate a code injection attack.**

In addition to the caveats mentioned in item #101, the team must also know how to properly implement dynamic code without introducing security flaws.

1. **(SV-75897r1\_rule) If DBMS authentication, using passwords, is employed, the DBMS must enforce the DoD standards for password complexity and lifetime.**

LDAP is used as the authentication mechanism for this system. Therefore, password complexity and lifetime will be the responsibility of the LDAP administrator.

1. **(SV-72549r1\_rule) Deploy a DBMS capable of producing the required audit records when object access occurs. Configure audit settings to create audit records when the specified access to the specified objects occurs.**

After enabling auditing as in item #3, ensure that the audit events, “document-insert”, “document-read”, and “document-update” are enabled.

1. **(SV-72551r1\_rule) Deploy a DBMS capable of producing the required audit records when object access occurs. Configure audit settings to create audit records when the specified access to the specified objects is unsuccessfully attempted.**

MarkLogic does not generate audit records specifically for unsuccessful attempts to access objects. These events should be tracked by the client applications.

1. **(SV-72551r1\_rule) Develop, document, and maintain a list of DBMS database objects, database configuration files, associated scripts, applications defined within or external to the DBMS that access the database, and DBMS/user environment files/settings in the System Security Plan. Record whether they do or do not contain DBMS passwords. If passwords are present, ensure that they are correctly hashed using one-way, salted hashing functions, and that the hashes are protected by host system security.**

This task will be the responsibility of the system administrators and the system developers.

1. **(SV-72565r1\_rule) Document and obtain approval for any non-administrative users who require the ability to create, alter or replace logic modules. Implement the approved permissions. Revoke any unapproved permissions.**

This task will be the responsibility of the system administrators.

1. **(SV-72585r1\_rule) Implement protective measures against unauthorized disclosure and modification during reception.**

TLS connectivity must be implemented between user queries and database results.

1. **(SV-72563r1\_rule) Deploy a DBMS capable of disabling a network function, port, protocol, or service prohibited by the PPSM guidance. Disable each prohibited network function, port, protocol, or service.**

All network functions, ports, protocols, and services are configurable and may be disabled. Upon installation, any unrequired network functionality may be disabled. After installation, it will be the system administrator’s responsibility to ensure that any new network functionality that is enabled fits within PPSM guidance.

1. **(SV-42750r3\_rule) Install all applications on directories separate from the DBMS software library directory. Relocate any directories or reinstall other application software that currently shares the DBMS software library directory.**

MarkLogic is installed on the /opt partition and in it’s own directory. Simply install all other applications in separate directories.