# Security Management for ECLI, NETCONF, and SFTP Users

## MTAS

## DESCRIPTION

2/1551-AVA 901 29/9 Uen B

# Contents

# 1 Introduction

This document provides an overview of the management model and concepts associated with the Security Management managed area.

A managed area is represented by a group of Managed Object Classes (MOCs) within the Managed Object Model (MOM).

# 2 Functions and Concepts

Security Management provides a management interface to configure the following on the Managed Element (ME):

- Lightweight Directory Access Protocol (LDAP) authentication of Northbound Interface (NBI) users
- Local authorization of NBI users
- Transport Layer Security (TLS)

An overview of Security Management is shown in Figure 1.

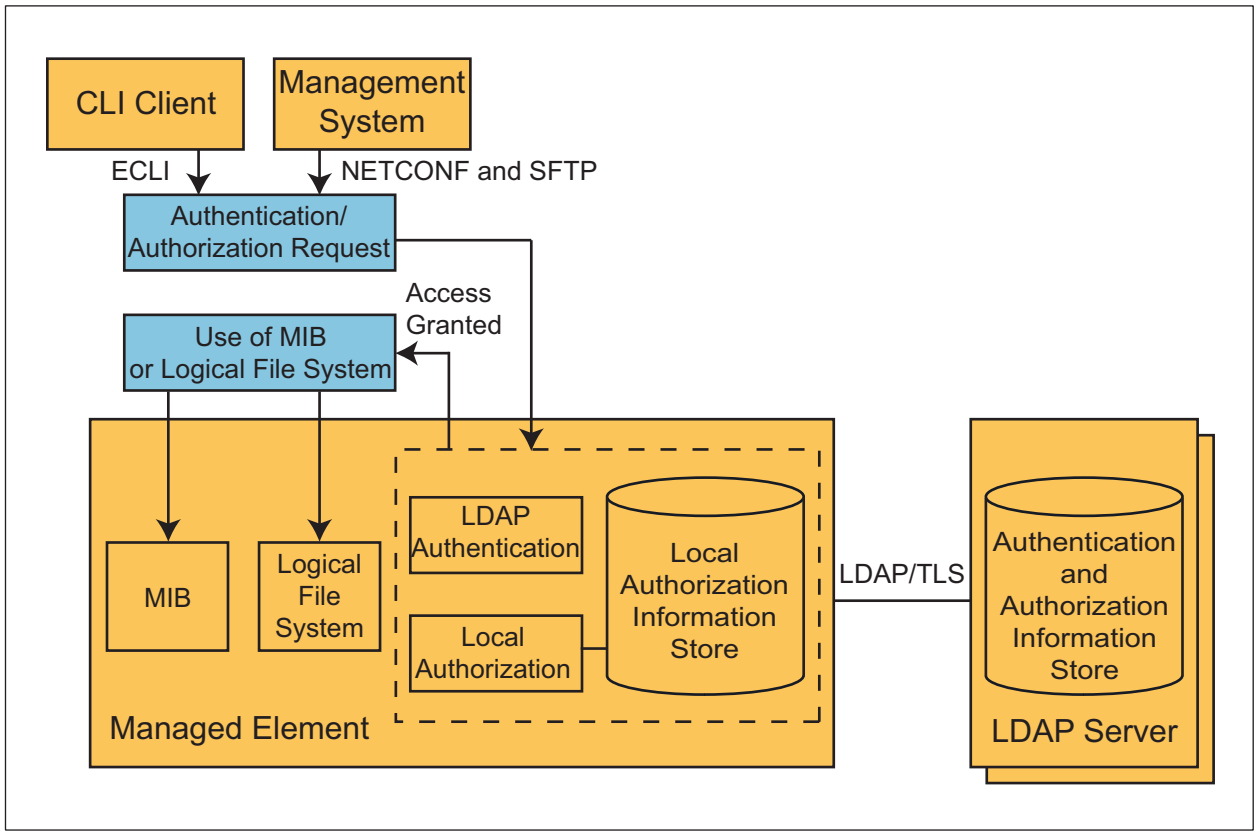


*Figure 1 Security Management Overview*

This document assumes that the ME has already been installed and initially configured. The initial configuration includes the necessary settings for the authentication of NBI users to an LDAP server and their authorization.

Authentication is used for querying user credentials and allowing user access. Role-Based Access Control (RBAC) and Target-Based Access Control (TBAC) authorization is used to ascertain the user access rights. Authentication and authorization are performed according to the organization authorization policy.

Concerning privacy, the communication between the NBI user and the ME is encrypted with SSH. The communication between the ME and the LDAP server can be encrypted using TLS.

For more information on the LDAP interface, refer to *LDAP-Based Authentication and Authorization Interface*.

## 2.1 NBI User Authentication

### Normal Access

By using username and password, the NBI user initiates a NETCONF, Ericsson Command-Line Interface (ECLI), or SFTP session over SSH to the ME and triggers a user LDAP authentication from the ME.

The LDAP authentication executes the following operations in succession:

1. An LDAP bind to identify the ME to the LDAP server with the configured bind Distinguished Name (DN) and bind password
2. An LDAP search, based on an LDAP filter to locate the user in the LDAP server
3. An LDAP bind to authenticate the user to the LDAP server with the username and user-provided password before returning the authentication result

Three different LDAP profile filters are supported for search; flexible filter, POSIX® groups filter, and Ericsson roles filter. The Ericsson roles profile filter is used together with the Ericsson Operations Support System (OSS) solution.

A primary and a secondary LDAP server are supported. The LDAP authentication first tries against the primary server and then against the secondary server.

All authentication attempts, whether successful or not, are recorded in the ME security log.

A successful NBI user authentication triggers an NBI user authorization.

When the Ericsson roles profile filter is used, the LDAP authentication can be selective based on the target type. In some networks, it can be required not to let all users have the right to log on to all MEs. For example, a network can span several countries and a user can be allowed to log on to MEs in only one country. This function is part of the TBAC functionality.

### Emergency Access

In situations where none of the LDAP servers is reachable or active, all LDAP authentications fail. However, local Linux users belonging to the

com-emergency Linux group can authenticate locally and get complete Management Information Base (MIB) access through the ECLI and NETCONF.

This emergency access is used only during troubleshooting.

## 2.2 NBI User Authorization

If authenticated locally by Linux, the NBI user obtains full access rights to the ME.

If authenticated by LDAP, the NBI user access rights depend on defined authorization rules that specify the permissions to a set of resources within the ME. The authorization rules are grouped into roles. A role is equivalent to the user occupation within an organization, for example, system administrator. A user can have one or more roles.

The roles are either retrieved from the LDAP server or are custom roles defined locally on the ME. The ME supports some predefined roles, see Section 2.2.2 Default Roles on page 6. Custom roles can only be configured over the NBI.

The authorization rules are all defined locally on the ME. Therefore the NBI user authorization is a local authorization. Custom rules corresponding to customer roles can be configured over the NBI.

Authorization rules provide different access levels to the MIB and the ECLI commands. Authorization rules are defined by permission types, see Section 2.2.1 Permission Types on page 5.

When the Ericsson roles profile filter is used, the authorization can be selective based on the target type. In some networks, it can be required to let a user have different management roles on different MEs. For example, the network can span several countries and it can be needed to let a user act as ‘‘admin’’ in one country, but only as ‘‘operator’’ in another. This function is part of the TBAC functionality.

### 2.2.1 Permission Types

Rules for access can be specified for Managed Objects (MOs), their attributes and actions. The execution of the ECLI commands and the NETCONF operations is not subject to authorization. However, the rules affect the result of the ECLI commands and the NETCONF operations that operate on MOs.

Permission types define different levels of access to the MIB according to Table 1.

*Table 1 Permission Types*

| Permission Type | Description |
|---|---|
| No access (NO_ACCESS) | The user has no read, write, or execute rights to the MOs, attributes, or actions |
| Execute (X) | The user can execute all actions in the MOM |
| Read (R) | The user can read MOs and get attribute values |
| Read and execute (RX) | The user can read MOs, get attribute values, and execute all actions in the MOM |
| Read and write (RW) | The user can create and delete MOs as well as get and set attribute values |
| Read, write, and execute (RWX) | The user can create and delete MOs, set, and get attribute values, as well as execute all actions in the MOM |

When a user with an authorization profile wants to access resources of the ME, the access request is authorized against matching security rules. The rules are checked in the following order:

1. All negative rules (with the NO_ACCESS permission) are evaluated. If a match is found, access is denied.

2. All positive rules (with X, R, RX, RW, and RWX permissions) are evaluated until a match is found; the corresponding access is granted. If no match is found, access is denied.

### 2.2.2 Default Roles

The ME supports the predefined default roles described in Table 2. These roles and the corresponding rules cannot be modified. The detailed permissions for each role are described in Section 3.1 Rules for Default Roles on page 12.

*Table 2 Default Roles*

| Default Role | Description |
|---|---|
| System Administrator | Responsible for the administration of all non-security-related attributes and capabilities of an ME, including features, configuration parameters, and monitoring |
| System Security Administrator | Responsible for the administration of all security-related attributes and capabilities of an ME, including user accounts and authorizations |
| Managed Function Application Administrator | Responsible for the administration of all non-security-related attributes and capabilities of the Managed Function, including features, configuration parameters, and monitoring |

*Table 2 Default Roles*

| Default Role | Description |
|---|---|
| Managed Function Application Security Administrator | Responsible for the administration of all security-related attributes and capabilities of the Managed Function, including user accounts and authorizations |
| Managed Function Application Operator | Can view some non-security-related attributes and capabilities of the Managed Function, including features, configuration parameters, and monitoring |

## 2.3 Transport Layer Security

TLS is used to secure the transport layer for the LDAP protocol.

The security algorithms used by TLS are determined as a result of a cipher suite handshake between the ME and the LDAP server. A cipher suite specifies a combination of authentication, encryption, key exchange, and message authentication algorithms. The cipher suite handshake selects the strongest common cipher suite between the ME and LDAP server. The selection is made from the ME-enabled cipher suite list.

The TLS can also be used with NETCONF.

## 2.4 Types of Operation

Security Management supports the following operations for an administrator with the System Security Administrator role:

**LDAP Authentication**

- View LDAP configuration

  The administrator can check the current LDAP configuration. The understanding of the LDAP configuration is a prerequisite for solving any authentication issues. The procedure in *View LDAP Configuration* provides further details on how to perform this operation.

- Unlock/lock LDAP authentication method

  In maintenance situations, the administrator can lock the LDAP authentication to prevent users from accessing the ME over the ECLI or NETCONF when it is not fully operational. When the LDAP authentication method is locked, only emergency access to the MIB is possible. The procedure in *Lock LDAP Authentication Method* provides further details on how to perform this operation.

  The administrator unlocks the LDAP authentication to enable user LDAP authentication when the ME is operational or to test the proper execution

of LDAP authentication. The procedure in *Unlock LDAP Authentication Method* provides further details on how to perform this operation.

- Change bind name and password for LDAP authentication

  The administrator can change the bind name and password required for password-based simple bind LDAP authentication. Such change can be triggered by the organization security policy. The procedure in *Configure LDAP Simple Bind* provides further details on how to perform this operation.

  The bind name and password are originally set at initial configuration.

- Change certificate settings for LDAP TLS

  The administrator needs to change the certificate settings for LDAP TLS in the following situations:

  – The Certificate Authority (CA) that has signed the LDAP server certificate has issued a new trusted certificate. This trusted certificate has been installed on the ME.

  – The ME node credential for LDAP TLS has been reinstalled.

  The procedure in *Configure TLS for LDAP* provides further details on how to perform this operation.

  **Note:** For information on trusted certificate and node credential installation, refer to *Certificate Management*.

  The certificate settings are originally set at initial configuration.

- Change Target-Based Access Control (TBAC)

  The administrator needs to change the TBAC settings when the current settings do no longer match the operator organization needs, for example, in the following situations:

  – The ME needs to become part of a different geographical domain.

  – The ME needs to become part of a different functional domain.

  – The ME needs to become part of a different competence domain.

  The procedure in *Configure Target-Based Access Control* provides further details on how to perform this operation.

**Local Authorization**

- View roles and rules

  The administrator can view the roles retrieved from the LDAP server and the rules defined in the ME. The understanding of the roles and rules is a

prerequisite for solving any authorization issues. The procedure in *View Roles and Rules* provides further details on how to perform this operation.

- Lock/unlock local authorization method

  The administrator locks the local authorization to give full access to all resources to all users authenticated by LDAP. Locking can be done in maintenance situations. The procedure in *Lock Local Authorization Method* provides further details on how to perform this operation.

  The administrator unlocks the local authorization to enable the local authorization based on defined rules and roles when the ME is operational or to test the proper execution of local authorization. The procedure in *Unlock Local Authorization Method* provides further details on how to perform this operation.

- Create, change, and delete custom roles and custom rules

  The administrator can create or change custom roles and custom rules when the predefined roles and rules do not match the needs of the organization authorization policy. The procedures in *Create Custom Role*, *Change Custom Role*, *Create Custom Rule*, and *Change Custom Rule* provide further details on how to perform these operations.

  The administrator can delete custom roles and custom rules when they are no longer needed by the organization authorization policy. The procedures in *Delete Custom Role* and *Delete Custom Rule* provide further details on how to perform these operations.

# 3 Managed Object Model

The Security Management managed area is represented in the *Managed Object Model (MOM)* as follows:

```
ManagedElement
   +-SystemFunctions
      +-SecM
         +-Tls
         +-UserManagement
            +-LdapAuthenticationMethod
               +-Ldap
                  +-EricssonFilter
                  +-Filter
            +-LocalAuthorizationMethod
               +-CustomRole
               +-CustomRule
               +-Role
                  +-Rule
```

For general information about the MOM, MOCs, MOs, cardinality, and related concepts, refer to *Managed Object Model User Guide*.

The Security Management MOCs are described in Table 3.

*Table 3 Security Management Managed Object Class Descriptions*

| Managed Object Class | Description |
|---|---|
| *SecM* | The root of the Security Management model. |
| *Tls* | Handles the TLS properties. |
| *UserManagement* | Describes the ME target types for TBAC. |
| *LdapAuthenticationMethod* | Handles the authentication method used to verify user credentials when attempting to log on to an ME. |
| *Ldap* | Handles the primary and secondary LDAP servers. |
| *EricssonFilter* | Defines the configuration used for the Ericsson filter (applicable when the value of *profileFilter* is `ERICSSON_FILTER`). |
| *Filter* | Defines the configuration used for the flexible filter (applicable when the value of *profileFilter* is `FLEXIBLE`). |
| *LocalAuthorizationMethod* | Handles the local authorization method used to verify the user access to the ME resources. |

*Table 3 Security Management Managed Object Class Descriptions*

| Managed Object Class | Description |
|---|---|
| *CustomRole* | Handles the authorization roles that can be assigned to users. |
| *CustomRule* | Handles the rules that define the user access control of MOs. |
| *Role* | Describes the authorization roles that can be assigned to users. |
| *Rule* | Describes the authorization rules that define the user access control to MOs. |

## 3.1 Rules for Default Roles

The detailed permissions for the default roles are described in Table 4 and Table 5.

For example, in Table 4, the rule named ‘‘FaultManagement_1’’ is part of the System Administrator role and defines read/write/execute permissions for the Fault Management (FM) MO, its attributes, actions, and its child MOs.

‘‘Deny’’ indicates the default behavior when no permission rule is defined. In such case, scope information is not applicable.

*Table 4 System Administrator Permissions for Default Roles*

| MOM Fragment | Permission | Scope | Rule Id |
|---|---|---|---|
| Managed Element (ME) | RWX | ME [(1)] | Top_1 |

*Table 4 System Administrator Permissions for Default Roles*

| MOM Fragment | Permission | Scope | Rule Id |
|---|---|---|---|
| System Functions (SF) | RWX | SF (1) | Top_2 |
| Back up and Restore Management (BRM) | RWX | BRM,* (2) | BackupAndRestoreManagement_1 |
| Fault Management (FM) | RWX | FM,* (2) | FaultManagement_1 |
| File Management | Deny | | |
| License Management (LM) | RWX | LM,* (2) | License_Management_1 |
| Performance Management (PM) | RWX | PM,* (2) | PerformanceManagement_1 |
| Security Management (SecM) | R | SecM (3) | SecurityManagement_2 |
| Certificate Management (CertM) | R | CertM,* (2) | CertificateManagement_1 |
| Software Inventory Management (SwIM) | RW | SwIM,* (2) | SoftwareInventory_1 |
| Software Management (SWM) | RWX | SWM,* (2) | SoftwareManagement_1 |
| System Management (SysM) | RWX | SysM,* (2) | SystemManagement_1 |
| Transport | RWX | Transport (1) | Top_3 |
| Equipment | Deny | | |

*(1) The MO, its attributes, and actions.*
*(2) The MO, its attributes, actions, and child MOs.*
*(3) Only the MO but not the attributes (enables navigation in the ECLI).*

*Table 5 System Security Administrator Permissions for Default Roles*

| MOM Fragment | Permission | Scope | Rule Id |
|---|---|---|---|
| Managed Element (ME) | R | ME (1) | Top_4 |

*Table 5 System Security Administrator Permissions for Default Roles*

| MOM Fragment | Permission | Scope | Rule Id |
|---|---|---|---|
| System Functions (SF) | R | SF (1) | Top_5 |
| Back up and Restore Management (BRM) | Deny | | |
| Fault Management (FM) | R | FM,* (2) | FaultManagement_2 |
| File Management | Deny | | |
| License Management (LM) | Deny | | |
| Performance Management (PM) | Deny | | |
| Security Management (SecM) | RWX | SecM (2) | SecurityManagement_1 |
| Certificate Management (CertM) | | | |
| Software Inventory Management (SwIM) | R | SwIM,* (2) | SoftwareInventory_2 |
| Software Management (SWM) | Deny | | |
| System Management (SysM) | Deny | | |
| Transport | Deny | | |
| Equipment | Deny | | |

*(1) Only the MO but not the attributes (enables navigation in the ECLI).*
*(2) The MO, its attributes, actions, and child MOs.*

# 4 Configuration Management

Security Management is accessed using NETCONF or the ECLI to manipulate the MIB.

The following operations, described in Operating Instructions using the ECLI, can be performed by an administrator with the System Security Administrator role:

**LDAP Authentication**

- *View LDAP Configuration*
- *Unlock LDAP Authentication Method*
- *Lock LDAP Authentication Method*
- *Configure LDAP Simple Bind*
- *Configure TLS for LDAP*
- *Configure Target-Based Access Control*

**Local Authorization**

- *View Roles and Rules*
- *Unlock Local Authorization Method*
- *Lock Local Authorization Method*
- *Create Custom Role*
- *Change Custom Role*
- *Delete Custom Role*
- *Create Custom Rule*
- *Change Custom Rule*
- *Delete Custom Rule*

# 5 MTAS Roles and Rules

*Table 6 MTAS Roles and Rules*

| **Managed Area** | **Mtas_Application_ Administrator Role** | **Mtas_Application_ Security_Administrator Role** | **Mtas_Application_ Operator Role** |
|---|---|---|---|
| Backup and Restore Management | **Mtas_AA_BrM**<br>'RWX' permission on the 'ManagedElement, SystemFunctions, BrM,*' resource | | |
| Fault Management | **Mtas_AA_Fm**<br>'R' permission on the 'ManagedElement, SystemFunctions, Fm,*' resource | **Mtas_ASA_Fm**<br>'R' permission on the 'ManagedElement, SystemFunctions,Fm,*' resource | **Mtas_AO_Fm**<br>'R' permission on the 'ManagedElement, SystemFunctions,Fm,*' resource |

*Table 6 MTAS Roles and Rules*

| Managed Area | Mtas_Application_ Administrator Role | Mtas_Application_ Security_Administrator Role | Mtas_Application_ Operator Role |
|---|---|---|---|
| File Management | **Mtas_AA_FileM_FGP**<br>'RWX' permission on the 'ManagedElement, SystemFunctions, FileM,FileGroupPolicy, *' resource | **Mtas_ASA_FileM_FGP**<br>'RWX' permission on the 'ManagedElement, SystemFunctions, FileM,FileGroupPolicy, *' resource | **Mtas_AO_FileM_FGP**<br>'R' permission on the 'ManagedElement, SystemFunctions, FileM,FileGroupPolicy, *' resource |
| | **Mtas_AA_FileM_FG_Alarm**<br>'RWX' permission on the 'ManagedElement=1, SystemFunctions=1, FileM=1,LogicalFs=1, FileGroup=AlarmLogs, *' resource | **Mtas_ASA_FileM_FG_Alarm**<br>'R' permission on the 'ManagedElement=1, SystemFunctions=1, FileM=1,LogicalFs=1, FileGroup=AlarmLogs, *' resource | **Mtas_AO_FileM_FG_Alarm**<br>'R' permission on the 'ManagedElement=1, SystemFunctions=1, FileM=1,LogicalFs=1, FileGroup=AlarmLogs, *' resource |
| | **Mtas_AA_FileM_FG_Alert**<br>'RWX' permission on the 'ManagedElement=1, SystemFunctions=1, FileM=1,LogicalFs=1, FileGroup=AlertLogs, *' resource | **Mtas_ASA_FileM_FG_Alert**<br>'R' permission on the 'ManagedElement=1, SystemFunctions=1, FileM=1,LogicalFs=1, FileGroup=AlertLogs, *' resource | **Mtas_AO_FileM_FG_Alert**<br>'R' permission on the 'ManagedElement=1, SystemFunctions=1, FileM=1,LogicalFs=1, FileGroup=AlertLogs, *' resource |
| | **Mtas_AA_FileM_FG_DC**<br>'RWX' permission on the 'ManagedElement=1, SystemFunctions=1, FileM=1,LogicalFs=1, FileGroup=DataCollection, *' resource | | **Mtas_AO_FileM_FG_DC**<br>'R' permission on the 'ManagedElement=1, SystemFunctions=1, FileM=1,LogicalFs=1, FileGroup=DataCollection, *' resource |
| | **Mtas_AA_FileM_FG_Mtas**<br>'RWX' permission on the 'ManagedElement=1, SystemFunctions=1, FileM=1,LogicalFs=1, FileGroup=Mtas, *' resource | | **Mtas_AO_FileM_FG_Mtas**<br>'R' permission on the 'ManagedElement=1, SystemFunctions=1, FileM=1,LogicalFs=1, FileGroup=Mtas, *' resource |
| | **Mtas_AA_FileM_FG_PM**<br>'RWX' permission on the 'ManagedElement=1, SystemFunctions=1, FileM=1,LogicalFs=1, FileGroup=PerformanceManagementReportFiles, *' resource | | |
| Performance Management | **Mtas_AA_PM**<br>'RWX' permission on the 'ManagedElement, SystemFunctions, Pm,*' resource | | **Mtas_AO_FileM_FG_PM**<br>'R' permission on the 'ManagedElement=1, SystemFunctions=1, FileM=1,LogicalFs=1, FileGroup=PerformanceManagementReportFiles, *' resource |

*Table 6 MTAS Roles and Rules*

| Managed Area | Mtas_Application_ Administrator Role | Mtas_Application_ Security_Administrator Role | Mtas_Application_ Operator Role |
|---|---|---|---|
| Security Management | **Mtas_AA_CertM**<br>'R' permission on the 'ManagedElement, SystemFunctions, SecM, CertM,*' resource | **Mtas_ASA_CertM**<br>'R' permission on the 'ManagedElement, SystemFunctions, SecM,CertM,*' resource | **Mtas_AO_CertM**<br>'R' permission on the 'ManagedElement, SystemFunctions, SecM,CertM,*' resource |
| | | **Mtas_ASA_MAA_Role_Rule**<br>'RWX' permission on the 'ManagedElement, SystemFunctions, SecM,UserManagement, LocalAuthorization Method,Role=Mtas_ Application_Administrator, *' resource | |
| | | **Mtas_ASA_MASA_Role_Rule**<br>'RWX' permission on the 'ManagedElement, SystemFunctions, SecM,UserManagement, LocalAuthorizationMethod, Role=Mtas_ Application_Security_Administrator, *' resource | |
| | | **Mtas_ASA_UserManagement**<br>'R' permission on the 'ManagedElement, SystemFunctions, SecM,UserManagement, *' resource | |
| Software Inventory Management | **Mtas_AA_SwIM**<br>'RW' permission on the 'ManagedElement, SystemFunctions, SwInventory, *' resource | **Mtas_ASA_SwIM**<br>'R' permission on the 'ManagedElement, SystemFunctions, SwInventory, *' resource | **Mtas_AO_SwIM**<br>'R' permission on the 'ManagedElement, SystemFunctions, SwInventory, *' resource |
| Software Management | **Mtas_AA_SwM**<br>'RWX' permission on the 'ManagedElement, SystemFunctions, SwM,*' resource | | |
| | **Mtas_AA_SwM_UP**<br>'RWX' permission on the 'ManagedElement, SystemFunctions, SwM, UpgradePackage.*' resource | | |

*Table 6 MTAS Roles and Rules*

| Managed Area | Mtas_Application_ Administrator Role | Mtas_Application_ Security_Administrator Role | Mtas_Application_ Operator Role |
|---|---|---|---|
| Application | **Mtas_AA_MtasFunction**<br>'RWX' permission on the 'ManagedElement, MtasFunction,*' resource | **Mtas_ASA_MtasFunction**<br>'R' permission on the 'ManagedElement, MtasFunction,*' resource | **Mtas_AO_MtasFunction**<br>'RWX' permission on the 'ManagedElement, MtasFunction,*' resource |
| | **Mtas_AA_MtasXdmsCai3gUser**<br>'NO_ACCESS' permission on the 'managed element, MtasFunction, MtasXdms, MtasXdmsCai3gUser, *' resource | **Mtas_ASA_MtasXdmsCai3gUser**<br>'RWX' permission on the 'ManagedElement, MtasFunction, MtasXdms ,MtasXdmsCai3gUser, *' resource | **Mtas_AO_MtasXdmsCai3gUser**<br>'NO_ACCESS' permission on the 'ManagedElement, MtasFunction, MtasXdms, MtasXdmsCai3gUser, *' resource |