

MiVoice MX-ONE

AD Authentication - Description

Release 7.4

November 2, 2021



Notice

The information contained in this document is believed to be accurate in all respects but is not warranted by **Mitel Networks™ Corporation (MITEL®)**. The information is subject to change without notice and should not be construed in any way as a commitment by Mitel or any of its affiliates or subsidiaries. Mitel and its affiliates and subsidiaries assume no responsibility for any errors or omissions in this document. Revisions of this document or new editions of it may be issued to incorporate such changes. No part of this document can be reproduced or transmitted in any form or by any means - electronic or mechanical - for any purpose without written permission from Mitel Networks Corporation.

Trademarks

The trademarks, service marks, logos and graphics (collectively "Trademarks") appearing on Mitel's Internet sites or in its publications are registered and unregistered trademarks of Mitel Networks Corporation (MNC) or its subsidiaries (collectively "Mitel") or others. Use of the Trademarks is prohibited without the express consent from Mitel. Please contact our legal department at legal@mitel.com for additional information. For a list of the worldwide Mitel Networks Corporation registered trademarks, please refer to the website: <http://www.mitel.com/trademarks>.

®, ™ Trademark of Mitel Networks Corporation
© Copyright 2021, Mitel Networks Corporation
All rights reserved

Contents

Chapter: 1	Introduction	1
	Description of AD Authentication	1
	Prerequisites	1
	Supported AD versions	2
Chapter: 2	Security	3
	Secure Socket Layer (SSL)	3
	Certificates	3
	Server Certificate	3
	Root Certificate	3
	Generating Certificate Signing Request (CSR) and Creating Keystore	4
	Creating Certificate by Signing Request File	4
	Creating the Key Store by using Signed Certificate	7
	Importing Certificates for PM and SNM	12
Chapter: 3	Users	13
Chapter: 4	Configuration	14
	Configure Web Protocol	14
	Configuring Web from HTTP to HTTPS with Self-Signed Certificate	14
	Configuring from HTTP to HTTPS with Uploaded Server Certificate	15
	Turning SSL off and back on again	15
	Configure SNM authentication method	15
	Certificate Management for AD Authentication	16
	Root Certificate or Signed Server Certificate	16
	Alternative to Root Certificates	17
	Search and Delete root certificates	17
	Configure AD Authentication	18
Chapter: 5	AD Authentication Maintenance	19
	Modifying AD Authentication Configuration	19

	Turning AD Authentication Off19
	Turning AD Authentication Back on19
Chapter: 6	AD Authentication Scenarios	20
	Scenario 1: PM Login22
	Scenario 2: SNM Login22
	Scenario 3: SNM Login over HTTP22
	Scenario 4: PM Login + use case 'Add Extension'23
	Scenario 5: In PM "click on subsystem"24
Chapter: 7	Fault Cases / Exceptions	26

Introduction

Description of AD Authentication

AD authentication refers to the possibility to configure Provisioning Manager (PM) to authenticate user passwords in Active Directory (AD) instead of in the PM user database.

The concept is not a true “*Single Sign On*”, but it will be possible to log in to PM and Service Node Manager (SNM) with the same user name/alias and password as when logging in to the corporate or department domain, as defined in AD.

When the PM server is configured for AD authentication, it will still be possible to log in with currently stored passwords in PM user database. This implies that if the AD server for some reason is out of service users who know the PM specific passwords will be able to continue working as normal.

AD authentication is only available when the PM web server is running in SSL mode. It is also required that SNM sub systems configured for PM authentication shall run in SSL mode.

Prerequisites

- The authentication towards AD is performed over protocol LDAPS. The AD server must therefore be enabled for LDAPS.
- AD authentication is implemented with LDAP authentication method ‘Simple authentication’ (see: <http://msdn.microsoft.com/en-us/library/cc223499.aspx>). As this requires that the user password is sent in clear text, AD authentication is only allowed when the entire web server PM is running on is configured for SSL (HTTPS).
- The AD server must be configured for SSL on the LDAP(S) interface.
- An SNM defined as subsystem for PM configured for AD authentication will also use the AD authentication feature when it is configured for PM authentication method (that is, not “Linux”). See [Scenario 2: SNM Login](#) for more details.
- Subsystems configured with PM authentication method must also be configured for SSL.
- Subsystems not configured for SSL must use Linux authentication method. That is, all users that need access must be provided with an own Linux account. Note that this is a high security risk if users get used to log in with AD credentials. See [Scenario 3: SNM Login over HTTP](#) for more details.
- Auto-login in SNM is disabled when you select **click on subsystem** in PM. Instead the user needs to be authenticated again, but as long as the SNM server is configured for SSL and PM authentication method, the same user name and password as for the domain (and PM login) will be used.
- All users that should be able to log in to PM and SNM must still be defined as users in the PM database. However, the password associated with each user is not in use as long as AD authentication is enabled.
- The user name or alias used when logging in to PM or SNM with AD authentication must match the one defined and stored in the PM database.

Example:

A user is defined as jdoe in PM and as jdoe@mydomain.com in AD.

The “Principal DNS Suffix” is configured as mydomain.com.

PM server will build the complete User-Principal-Name as `jdoe@mydomain.com` and it is possible to log in to PM with `jdoe`, only.

The login name or alias must have a perfect match in the PM database to read validate the users' privileges, even if the password is validated in AD.

See [Configure AD Authentication](#) for more details.

Supported AD Versions

- Windows Server 2008 R2 (64 bit)
- Windows Server 2012 (64 bit)
- Windows Server 2016 (64 bit)
- Windows Server 2019 (64 bit)

Security

Secure Socket Layer (SSL)

The server, on which PM is installed, needs to be configured for SSL (HTTPS). End users will be using HTTPS when accessing the application web page. The same applies for web services between PM and SNM. Also the AD communication with LDAP shall be encrypted over SSL. To configure SSL/HTTPS, use the `webserver_config` command. See [Configuration](#) for more details.

Certificates

The use of certificates is all about trust. We need to be able to trust the other party we connect to and communicate with. Due to this, two different kinds of certificates are required when running AD authentication:

- **Server Certificate:** As a server to run the web application over HTTPS
- **Root Certificate:** As a client when we connect to the AD server.

Server Certificate

The server certificate is used for the purpose of other (clients) to trust the PM when connecting over HTTPS. This could either be a self-signed certificate or a certificate signed by a Certificate Authority (CA). See [Configure Web Protocol](#) for more details.

Root Certificate

PM and SNM are running as Java driven applications under Jboss.

When connecting to an AD server over SSL, it is required that the Java application can verify the AD server as a trustful source. To make this functioning, a proper certificate must be imported and stored in the Java trust store.

The certificate could be the AD server's own server certificate that is distributed when connecting to it over SSL. This will work perfectly as long as the AD connection is defined as a fixed host name or IP address representing the AD server.

However, in a more complex environment, it is more common that the AD connection is made through a broadcast through the domain controller or by addressing a sub domain or even the entire domain. In this situation, it would be inconvenient to store the server certificates for all possible transfer servers the connection will need to verify as trustworthy. Instead, a proper root certificate that can verify the signature on each server's server certificate should be imported. See [Certificate Management for AD Authentication](#) for more details.

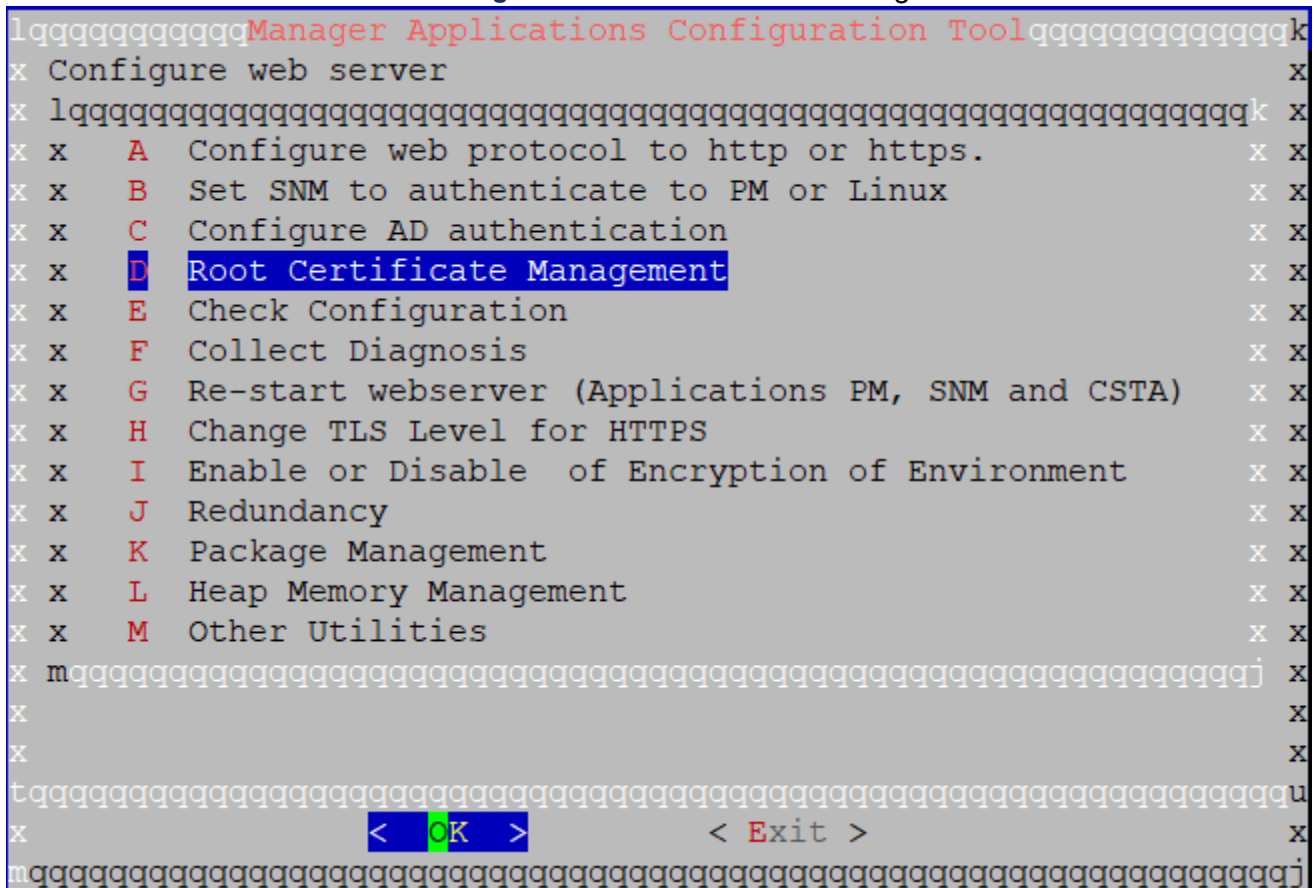
Generating Certificate Signing Request (CSR) and Creating Keystore

This section explains how to use the `webserver_config` utility to generate the Certificate Signing Request (CSR) and how to create the Keystore once you receive the Signed Certificate from the Certificate Authority.

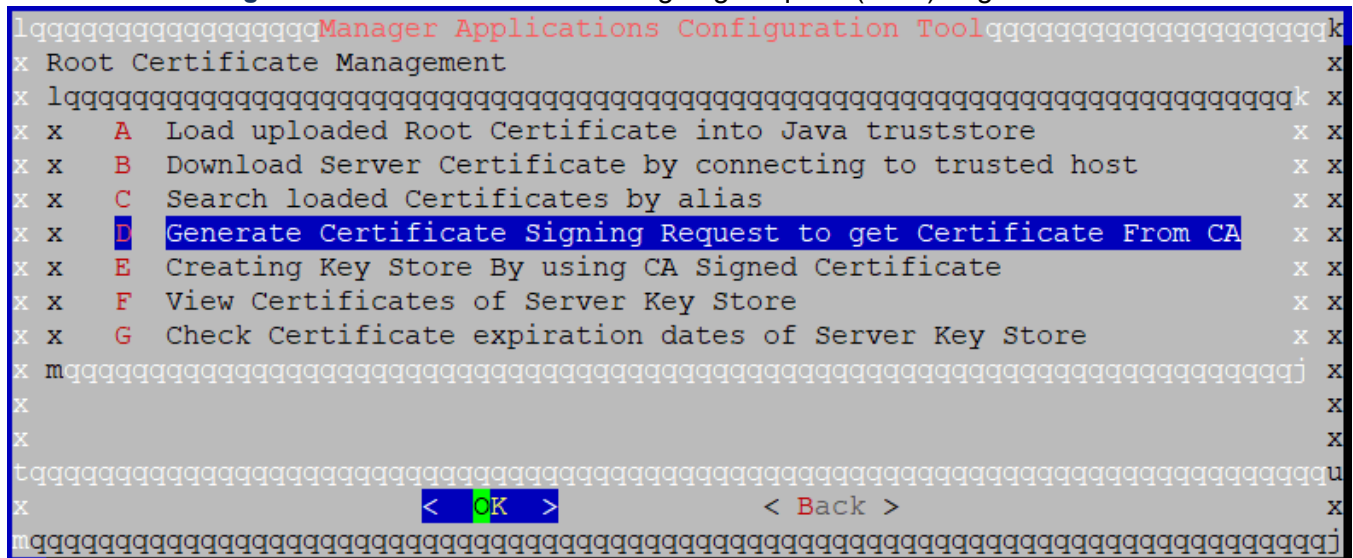
Creating Certificate by Signing Request File

1. Open the `webserver_config` utility. Select **Root Certificate Management**.

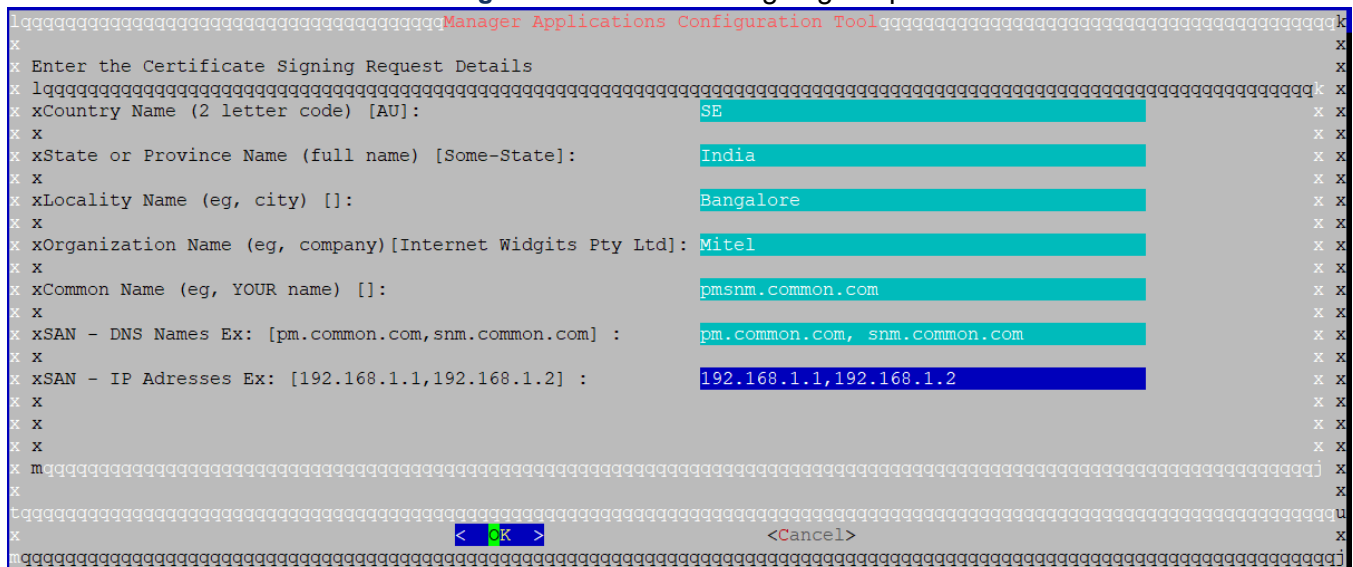
Figure 2.1: Root Certificate Management



2. Select **Generate Certificate Signing Request** to get Certificate from CA.

Figure 2.2: Generate Certificate Signing Request (CSR) to get Certificate From CA

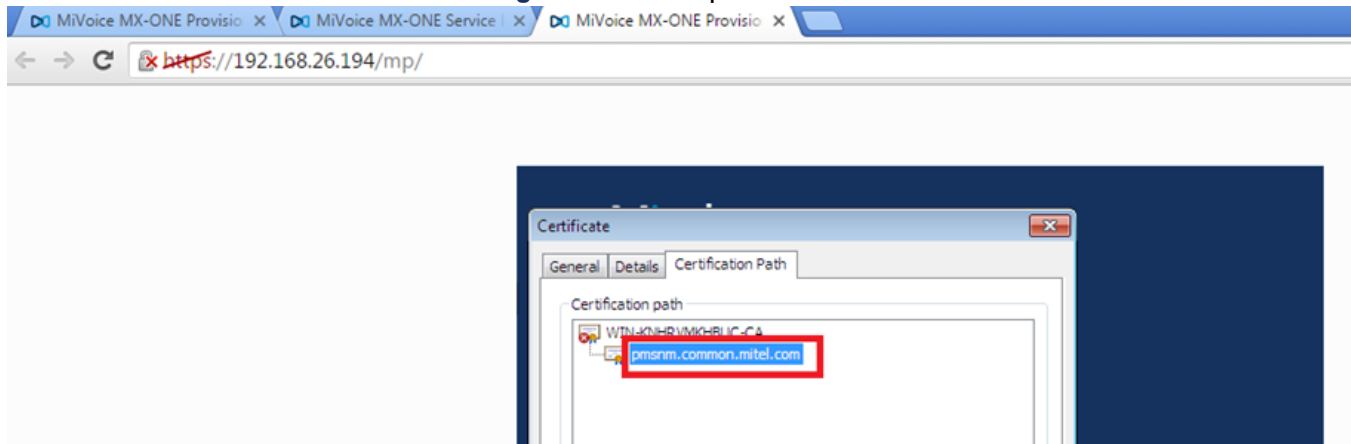
3. Enter all the CSR request details:

Figure 2.3: Certificate Signing Request Details

NOTE: In **Common Name**, Domain Name must be entered to get displayed in the Certificate.

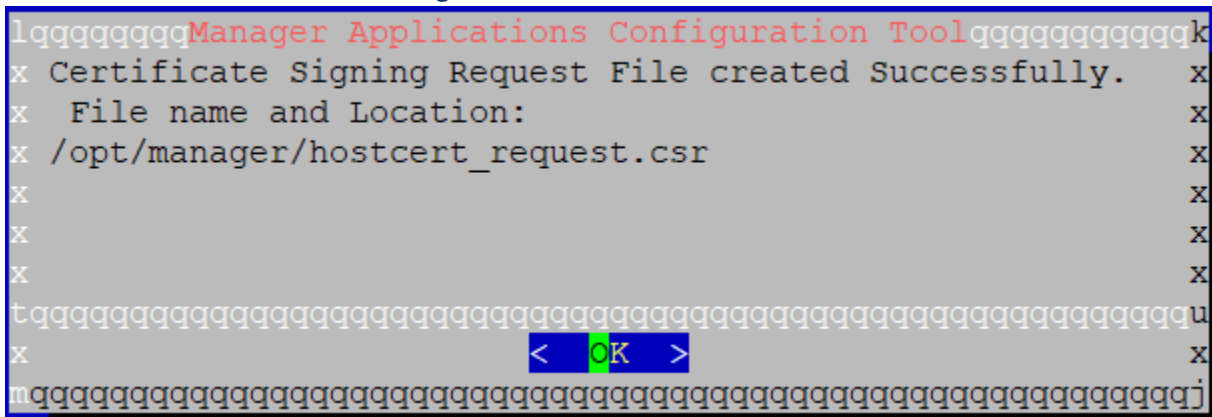
In the above screenshot, **pmsnm.common.mitel.com** is the common name, which can be anything based on the organization.

Figure 2.4: Example of Common Named



- a. SAN – DNS Names: Subject Alternate DNS Names
 - b. Ex: pm.common.com,snm.common.com
 - c. SAN – IP Addresses: Subject Alternate IP Addresses. Ex: 192.168.2.1, 192.168.2.3
 - d. Click **OK** after filling the required details. The system will generate the Certificate Signing Request file, located at /opt/manager/hostcert_request.csr.
4. You must copy this file to the local system and share it with Certificate Authority (CA) for to procure the Signed Certificate.

Figure 2.5: Successful creation of CSR file



Creating the Key Store by using Signed Certificate

Once you receive the Signed Certificate from the Certificate Authority, you must copy that certificate to the server where you want to enable the HTTPS.

1. Open the `webserver_config` utility.

Figure 2.6: Creating the Key Store

```

lqqqqqqqqqqqqqqqqqqqqManager Applications Configuration Toolqqqqqqqqqqqqqqqqqqqqk
x Root Certificate Management x
x lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk x
x x A Load uploaded Root Certificate into Java truststore x x
x x B Download Server Certificate by connecting to trusted host x x
x x C Search loaded Certificates by alias x x
x x D Generate Certificate Signing Request to get Certificate From CA x x
x x E Creating Key Store By using CA Signed Certificate x x
x x F View Certificates of Server Key Store x x
x x G Check Certificate expiration dates of Server Key Store x x
x mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj x
x x
x
tqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqu
x < OK > < Back > x
mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
  
```

2. Select **Creating Key Store By using CA Signed Certificate**.
3. select the path of the Signed Certificate in the system.
4. Press **TAB** to move between Directories, Files and Buttons sections.
5. Press **Space** and select the Directory and Files.

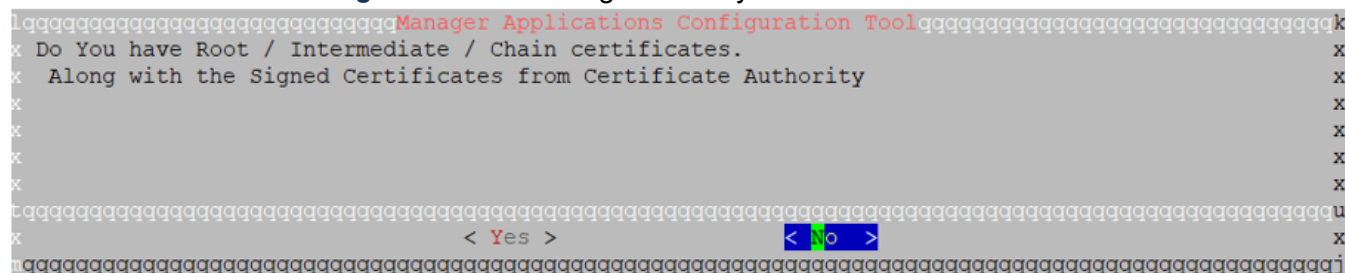
Figure 2.7: Selecting the directory and files

```

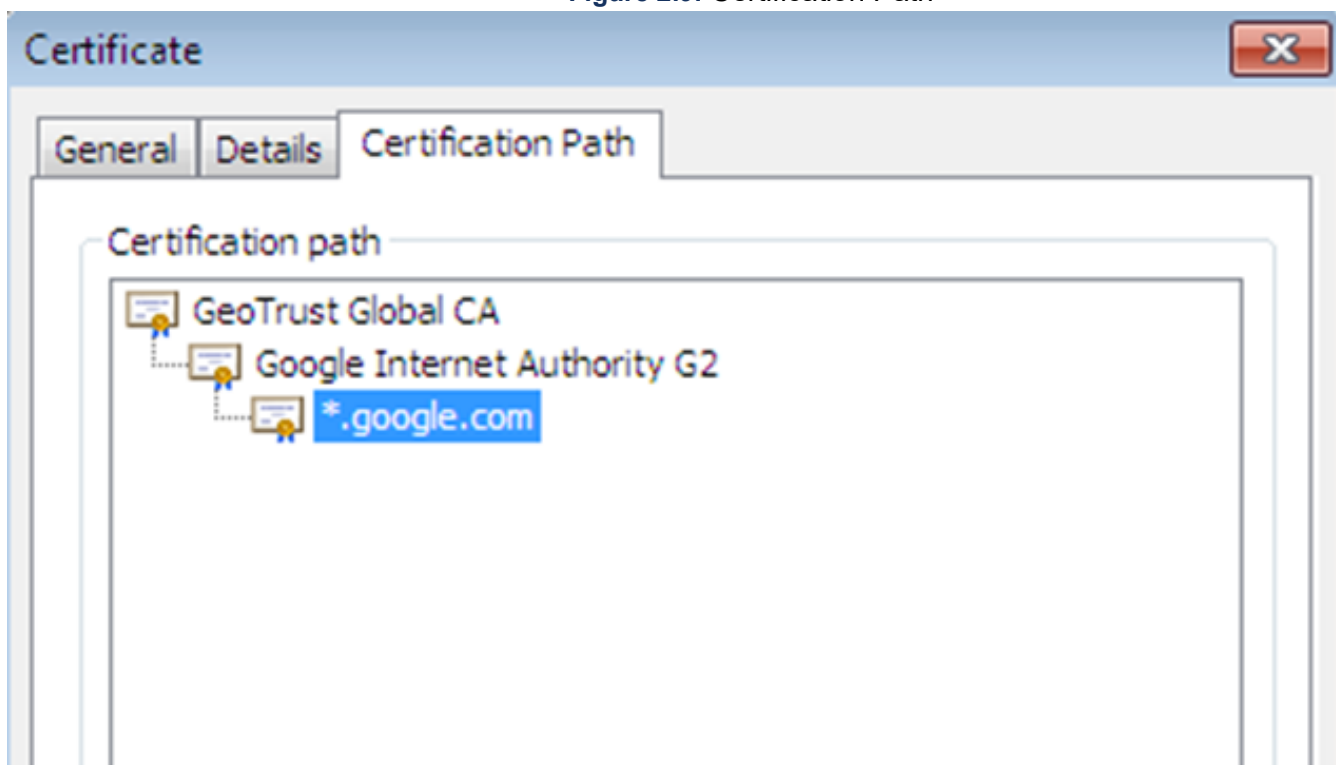
lqqqqqqqqqManager Applications Configuration Toolqqqqqqqqqqqqk
x Directories                               Files                               x
x lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk lqqqqqqqqq+ (-) qqqqqqqqqqqqqqqqqqqqk x
x x.                                           xx.profile                             x x
x x..                                          xx0                                    x x
x x.cassandra                                xx1                                    x x
x x.config                                   xxMX-ONE_7.3.sp0.hf2.rc4.binx x
x x.fonts                                    xxMX-ONE_7.3.sp1.hf0.rc26.bix x
x x.local                                    xxMX-ONE_7.3.sp1.hf0.rc29.bix x
x x.ssh                                       xxMYCSR.csr                            x x
x xbin                                        xxPRIVATEKEY.key                       x x
x xinit                                       xxUS_export_policy.jar                 x x
x xinstall_sw                               xxabc.key                             x x
x xmxo6576                                   xxabd.key                             x x
x xpmsn                                       xxcertnew.cer                         x x
x x                                           xxhostcert_request.csr                x x
x x                                           xxlocal_policy.jar                    x x
x x                                           xxmp.ear                             x x
x x                                           xxmp.ear.bac                         x x
x x                                           xxmpvo.jar                           x x
x x                                           xpass.txt                            x x
x x                                           xxroot.cer                           x x
x x                                           xxssl_temp.cnf                       x x
x mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq83%qqqqqj mqqqqqqqqq↓ (+) qqqqqqqqqqq73%qqqqqj x
x lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk x
x x/local/home/mxone_admin/certnew.cer x x
x mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj x
tqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqu
x < OK > <Cancel> x
mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj

```

6. Confirm whether you have any Root / Chain Certificates available with our CA Signed Certificate.

Figure 2.8: Confirming whether you have Root/Chain Certificates

What are Root / Intermediate / Chain Certificates?

Figure 2.9: Certification Path

When you open the certificate like above, here "GeoTrust Global CA" is a Root Certificate, "Google Internet Authority G2" is Chain / Intermediate Certificate, "*.google.com" is a Signed Certificate. These Root and Signed Certificates will be provided by the Certificate Authority or can be downloaded from their sites.

If there is only Root Certificate, you can use that file directly and ignore the below section otherwise merge all these Root and Chain Certificates into one file

When you open these Root / Chain Certificates in the notepad, the content might look like below:

-----BEGIN CERTIFICATE-----

-----END CERTIFICATE-----

7. Merge the content like below in the file First Root Certificate, then Chain Certificate:

-----BEGIN CERTIFICATE-----

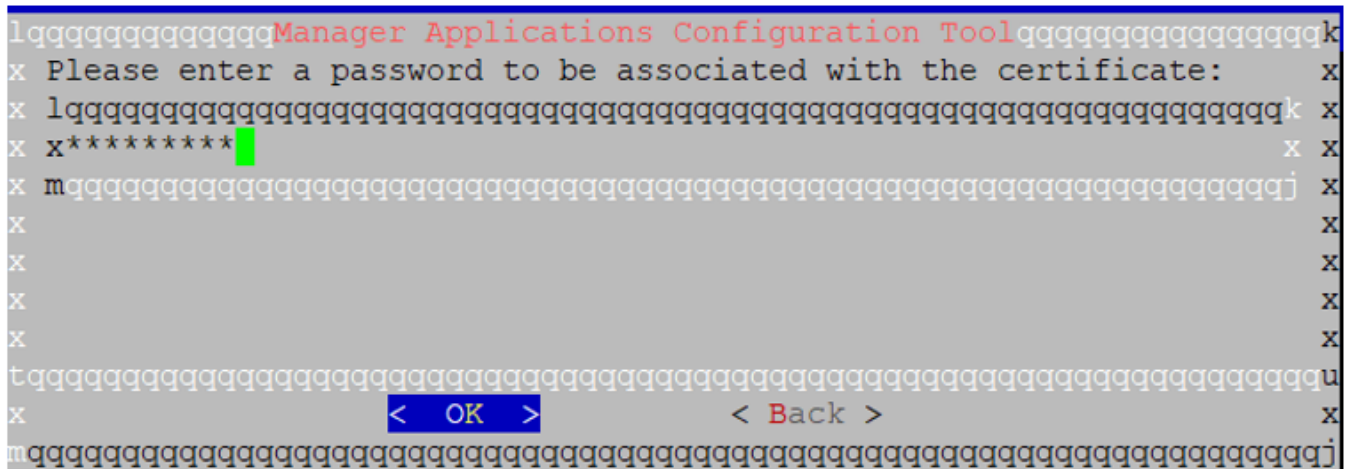
Root Certificate Information

-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----

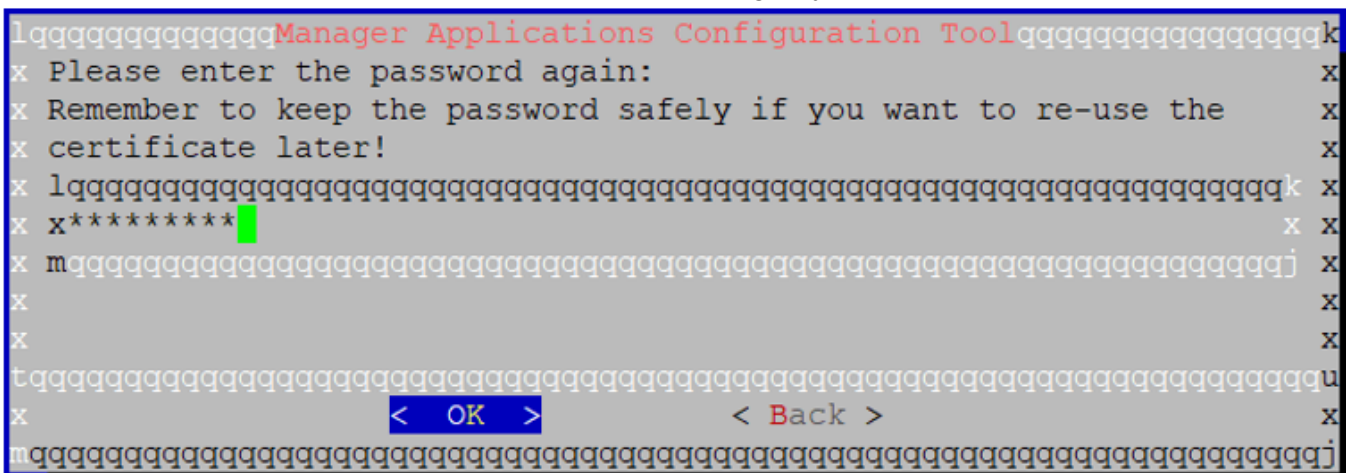
Chain Certificate Information

Figure 2.11: Creating the Key Store



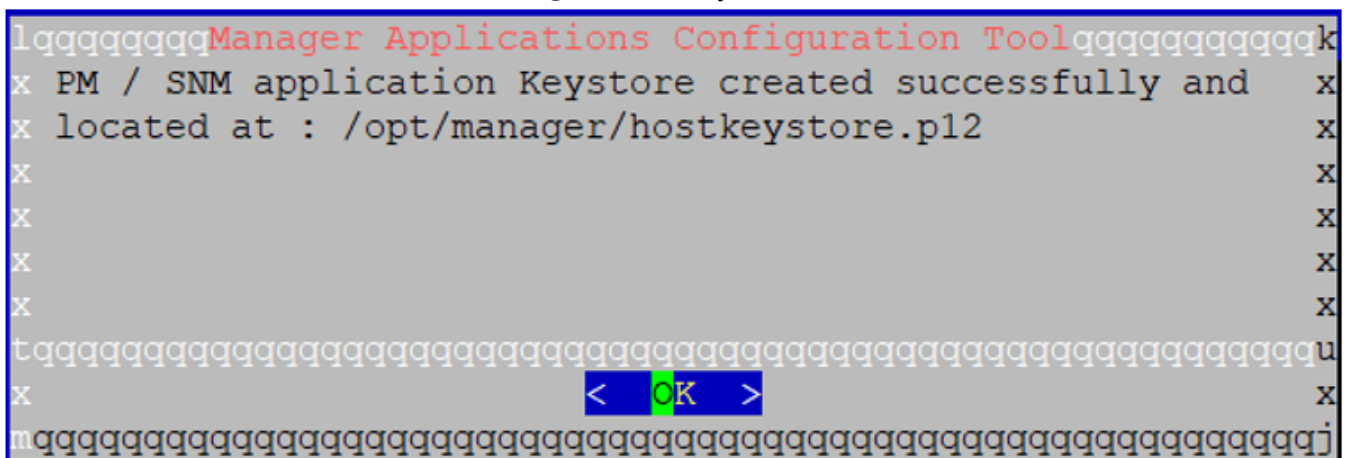
- d. Re-Enter the same password for Validation**

Figure 2.12: Validating Key Store creation



- e. Then System creates the Key Store File and locates at `/opt/manager/hostkeystore.p12`

Figure 2.13: Key Store File creation



- f. You can use this file and enable the HTTPS.

Webserver_config ? Configure web protocol to enable HTTP or HTTPS? Change to HTTPS / Keep HTTPS? Certificate is uploaded to file System? select the created Key Store file from /opt/manager/hostkeystore.p12.

Importing Certificates for PM and SNM

Follow the steps below to generate and import certificates for PM and SNM by using local Windows Server as Certificate Authority.

1. Generate certificate requests by using IIS for PM and SNM.
2. Upload the certificate request to “certsrv” application.
3. Issue the certificates by using Microsoft Management Console (MMC).
4. Download the Issued certificates from “certsrv” application.
5. Complete the certificate request by using IIS.
6. Generate the PFX files for PM and SNM with private key to use as a keystore.
7. Follow below steps to enable SSL for PM and SNM applications.
 - a. Open Webserver_config utility
 - b. Select **Configure web protocol to HTTPS**
 - c. Select **Change to HTTPS**
 - d. Select **Certificate is uploaded to file system**
 - e. Select the “PFX” file which was generated in Step 6.
8. Follow below steps to exchange the certificates between PM, SNM, AD and CMG
 - a. Open Webserver_config utility
 - b. Select **Root Certificate Management**
 - c. Select **Load Upload Root Certificate into Java truststore.**
 - d. Select the certificate of other system and provide the alias name. For example, if you are executing above steps in PM, then you must select the certificate of SNM that you have downloaded from Step 4
 - e. In case if the Other system is AD or CMG, then you must get the AD or CMG certificates from AD or CMG servers by using Microsoft Management Console (MMC) tool.
 - f. Importing the certificate of PM/SNM, AD and or CMG is applicable for Co-Existing system (PM/SNM are in same server) also.
 - g. You can import the certificate by connecting to other system directly by selecting **webserver_config > Root Certificate Management > Download Certificate by connecting to trusted host > Enter the Other Server Name / IP, Port and alias names.**
9. At AD or CMG side, have to import the server Certificate of PM by using Microsoft Management Console (MMC) tool to the “trusted people” section.
10. Import the Root Certificate of PM to AD or CMG by using Microsoft Management Console (MMC) tool to “trusted Certificate Authority” Section.

NOTE: Certificates that is generated in the steps above is a self-signed certificate using Microsoft IIS and there are also other ways of doing that. It is recommended to use the company’s certificate.

Users

There are two types of users that shall be considered in PM.

The first user type is “*end user*” which is the user that logs on to PM and/or SNM web page. This user may have different levels of privileges, from a “*plain*” end user in PM up to an administrator defined as “*Super User*”.

The second user type is “*service account*” which is the account used when PM and SNM communicate in the background through web services. This account is set up as any “*end user*” account and should be provided with enough privileges to serve all actions needed to be performed (independently of who is performing them).

The “*service account*” is the user defined as “User ID in Subsystem” when adding SNM as subsystem in PM. Whenever an action that requires access to SNM is performed in PM this account’s privileges are checked.

See example of how ‘end user’ versus ‘service account’ is used: [Scenario 4: PM Login + use case ‘Add Extension’](#).

Configuration

All configuration of AD authentication is done through the configuration utility for the web server, which is provided with each installation of PM or SNM. It is only when PM is installed that the configuration of AD authentication is available.

All necessary configuration scripts are available through the following command:

```
webserver_config.
```

Configure Web Protocol

To set the server in SSL mode, choose to configure the web server protocol and then choose HTTPS.

In SSL mode the PM server must have its own server certificate. This may be a self-signed root certificate, or a properly signed certificate from a Certification Authority (CA). The latter will be considered as 'safe' for visitors of the PM web as long as the CA's root certificate is included in the clients' browser. All modern web browsers are provided with a subset of root certificates from trusted CA's.

A self-signed certificate is from a technical point of view like a signed certificate from a CA. The main, but important, difference is that the self-signed certificate cannot be verified as trustworthy by other servers or clients. The clients web browser will show a warning when the page is visited.

If a server certificate is to be imported to the PM server, it must first be uploaded to the Linux file system. Use any kind of SFTP (Secure File Transfer Protocol) client, like e.g. WinSCP to connect and upload the certificate file. The certificate should be in the format PKCS#12.

Configuring Web from HTTP to HTTPS with Self-Signed Certificate

Follow the procedure below to configure the Web portal from HTTP to HTTPS with Self-Signed Certificate:

1. As root, or with sudo privileges, run the `webserver_config` command.
2. Select **Configure web protocol to http or https**.
3. Select **[CHANGE TO/KEEP] HTTPS**.
4. Confirm the next two windows by pressing the **Type** key.
5. Select **Create a self-signed certificate**.
6. Type the password to be associated with the certificate.

NOTE: This step is required in case the certificate shall be re-used later on.

7. Re-type the password.
8. Accept a restart of the web server. The configuration will not take effect before a restart of the web server is done.

Configuring from HTTP to HTTPS with Uploaded Server Certificate

Follow the procedure below to configure the Web portal from HTTP to HTTPS with uploaded server Certificate:

1. As root, or with sudo privileges, run the `webserver_config` command.
2. Select **Configure web protocol to http or https**.
3. Select **[CHANGE TO/KEEP] HTTPS**.
4. Confirm the next two dialogs by pressing the **Type** key.
5. Select **Certificate is uploaded to file system**.
6. Use the Up/Down key, and Enter on the keyboard, to navigate to your certificate file, mark it and press **Enter**.
7. Accept that it is the correct file to be used.
8. Provide the password associated with the certificate.

NOTE: If an incorrect password is used, the certificate will not function.

9. Accept a restart of the web server. The configuration will not take effect before a restart of the web server is done.

Turning SSL off and back on again

When the server has been configured in SSL mode, it means a server certificate already is available. In case the SSL is turned off and should be turned on again, the configuration script will identify the previously used certificate which could be re-used.

The procedure looks exactly like the one for an uploaded server certificate, with the difference that you already at step 5 can choose to re-use the found certificate instead.

As for the case with the uploaded server certificate, it is an absolute must that the password matches the one associated with the certificate.

Configure SNM authentication method

The SNM authentication method refers to in which way a user that logs in to SNM is authenticated. It could be either of *“Linux authentication”* or *“PM authentication”*.

For Linux authentication it is required that the user has a Linux account. This is to be created according to standard Linux procedures via command or using `mxone_maintenance` utility. Log-in as user `mxone_admin`, and key the command `sudo -H /opt/mxone_install/bin/mxone_maintenance` and select option user and follow the instructions on screen.

For PM authentication, SNM will instead send a (SOAP) request to the configured PM server to verify user credentials and privileges (authenticate and authorize).

NOTE: If the SNM server is not running SSL (HTTPS) and PM is configured for AD authentication, the SNM authentication method must be set to Linux authentication. This means that administrators that need to log in to SNM must be provided with a separate Linux account. This account will not be authenticated

towards AD. Note that this is a security risk if users get used to log in with AD credentials. See further [Scenario 3: SNM Login over HTTP](#).

If PM and SNM are running on the same server, that is, MX-ONE Service Node 1 (LIM 1), the SNM authentication method will automatically be set to PM authentication method as soon as AD authentication is enabled.

To set PM authentication method when PM is running on a standalone server, log in to the SNM server and do the following:

1. As root, or with sudo privileges, run the `webserver_config` command.
2. Select **Set SNM to authenticate to PM or Linux**.
3. Select **[CHANGE TO/KEEP] PM authentication**.
4. Enter the required information in fields in the window.
5. Accept a restart of the web server. The configuration will not take effect before a restart of the web server is done.

Certificate Management for AD Authentication

This topic provides information on how to obtain and manage root certificates for AD authentication.

Root Certificate or Signed Server Certificate

To import a root certificate or a server certificate from another server that should be trusted.

Do as follows:

1. Connect to the Linux server with a SFTP client such as WinSCP and upload the root certificate to the file system.
2. As root, or with sudo privileges, run command `webserver_config`.
3. Select **Root certificate management**.
4. Select **Load uploaded Root Certificate into Java trust store**.
5. Use the Up/Down on the keyboard, to navigate through the file system and find the uploaded certificate.
6. Select the certificate file and press **Enter**.
7. Select **Yes**, and press **Enter**.
8. Assign an alias (as identifier) to the certificate. This will be the only way to easily search and – when applicable – delete the certificate from trust store.
9. Press **Enter**
10. Confirm (on command line) with yes or y, to trust the certificate.

Alternative to Root Certificates

It is for practical reasons not usual to have properly signed root certificates in lab environments, but it might still be necessary to verify AD authentication functionality. As long as the AD server can be directly accessed through either IP address or host name the following procedure can be used.

Do as follows:

1. As root, or with sudo privileges, run the `webserver_config` command.
2. Select **Root certificate management**.
3. Select **Download Server Certificate by connecting to trusted host**.
4. Type server IP address or host name.
5. Type SSL port (commonly used is 636 for LDAPS).
6. Type an alias (as identifier) to the certificate. This will be the only way to easily search and – when applicable – delete the certificate from trust store.
7. Press **Enter**.
8. Follow the progress on the screen. If the connection takes very long time, e.g. because of faulty inserted IP address, you may have to press ctrl-c to interrupt. In that case the process needs to be started over again.
9. Repeat the steps from Step 3 to Step 8 for PM/SNM machine also, to download the certificate of PM/SNM local certificate to Java Trust Store.
10. Confirm (on command line) with yes or y to accept the certificate.

Search and Delete root certificates

There might be good reasons for finding root certificates in the Java trust store. If e.g. a certificate was trusted added by mistake or if in general find out what's already stored. All certificates in trust store are identified through a unique alias.

To search Java trust store and optionally delete a certificate, perform the following steps:

1. As root, or with sudo privileges, run the `webserver_config` command.
2. Select **Root certificate management**
3. Select **Search loaded Certificates by alias**.
4. Type the full or partial alias – or all to view all stored certificates. All aliases are in lower case.
5. If any hits a result list appears.
6. To view details, select **View** and press **Type**.
7. To go back, press **Type**.
8. To delete a certificate from trust store, select **Delete** and press **Type**.
9. To confirm deletion, select **Yes** and press **Type**, or select **No**.

Configure AD Authentication

The following must be considered before AD authentication is configured:

- How shall the AD server be addressed:
 - IP address
 - Host name
 - Domain name
- Which port number is applicable for LDAPS on the AD server (default is 636)
- Is there a need to define the Base Context DN (Distinguished Name) for AD? This is equal to defining in which part of the AD the user shall be sought. Example:
`CN=Users,DC=mysubdomain,DC=mydomain,DC=org`
To search the entire AD, leave the field blank.
NOTE: Base Context DN must have a perfect match, and it cannot be verified by any means by the PM configuration tool.
- How the user credentials shall be typed:
 - a useralias only
 - useralias@domain (also commonly known as User-Principal-Name)
- All users belong to the same domain, ourdomain.com.
- The AD server uses User-Principal-Name when authorizing users, i.e. by default the user e-mail address, e.g. jdoe@ourdomain.com.
- In the AD authentication configuration, define the Principal DN suffix as ourdomain.com.
- The users will now use the alias only, e.g. jdoe when logging in to PM.
NOTE: If a user tries to login with jdoe@ourdomain.com, it will still be suffixed and the entire string jdoe@ourdomain.com@ourdomain.com will be sent to AD. The login will fail. The same applies if there are users with different domain names in their User-Principal-Name.
- Type the Principal DN suffix.
- Do never Type the character @. It will automatically be appended.

Do as follows:

- As root, or with sudo privileges, run the `webserver_config` command.
- Select **Configure AD authentication**.
- If AD authentication already is configured, you will be asked if you'd like to turn it off, select **No**.
- Type host name, IP address or domain name to address the AD server.
- Type the port for LDAPS on the AD server (default 636).
- Type the Base Context DN, if desired (leave blank if not).
- Type the Principal DN suffix.
- Acknowledge restart of the web server. The configuration will not take effect before a restart of the web server is done.
NOTE: If the server not is configured for SSL/HTTPS, the AD configuration wizard will be interrupted.
NOTE: For AD authentication, PM internally uses `userid@UserPrincipalSuffix` as *User/ID* and it should match the `SAMAccountName`; otherwise, AD authentication may not work.

AD Authentication Maintenance

This topic provides information on the managing the AD authentication settings.

Modifying AD Authentication Configuration

To modify any settings in the AD authentication configuration, such as AD server host name and port or Principal DN suffix, the procedure is basically the same as when activating it the first time.

Do as follows:

1. As root, or with sudo privileges, run the `webserver_config` command.
2. Select **Configure AD authentication**.
3. On question Do you want to turn AD authentication off?, select **No**.
4. Modify applicable settings.
5. Accept restart of the web server. The modified configuration will not take effect before a restart of the web server is finished.

Turning AD Authentication Off

The AD authentication service can be temporarily or permanently turned off. In any case the previous settings are saved in case the service should be turned on again later.

Do as follows:

1. As root, or with sudo privileges, run the `webserver_config` command.
2. Select **Configure AD authentication**.
3. On question Do you want to turn AD authentication off? select **Yes**.
4. Accept restart of the web server. The modified configuration will not take effect before a restart of the web server is done.

Turning AD Authentication Back on

An already existing configuration of AD authentication service can be turned back on. The procedure to do this is exactly the same as when configuring AD authentication the first time, with the difference that the fields are pre-filled with previous settings. If applicable, modify the settings.

The modified configuration will not take effect before a restart of the web server is finished.

AD Authentication Scenarios

AD authentication is performed on the server where PM is installed. If no PM is in use in the MX-ONE system, only Linux accounts can be used to access SNM.

For each login attempt it is checked whether the user is:

1. Authenticated - a match between user alias and Typed password is found.
2. Authorized - the authenticated user has enough privileges.

At each login attempt it is first checked if the user can be authenticated in the PM user database. If not, an AD authentication attempt is made and if this is successful the user is authorized from PM user database.

Through this sequence it will always be possible to log in with an administrative service account in PM also if for some reason the connectivity with the AD server is lost. It is also possible to log in with a password stored in PM user database or a password connected to the user account in AD.

Figure 6.1: User authenticated and authorized in PM user database

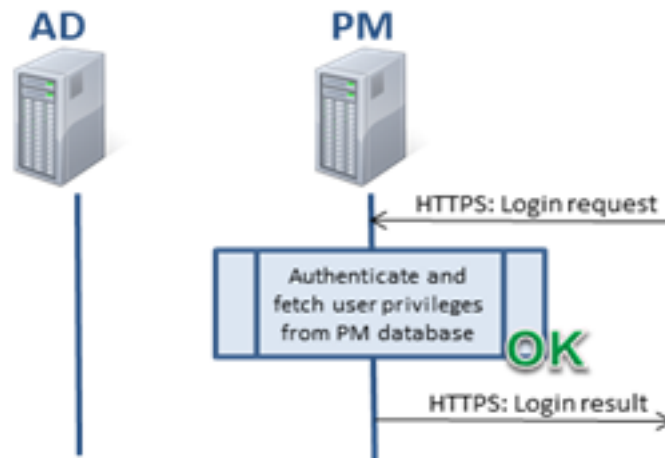
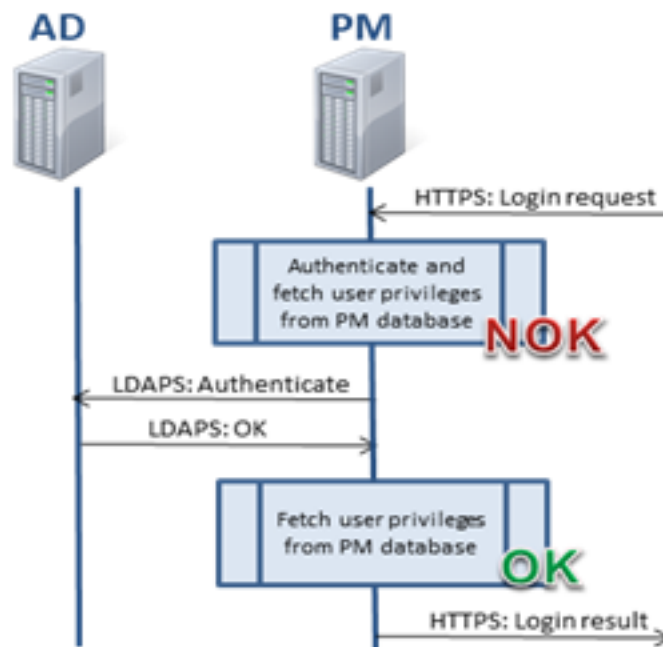
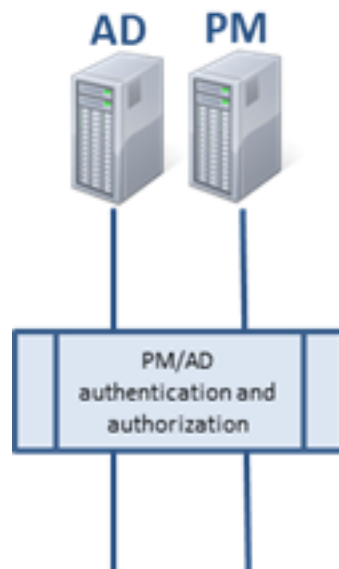
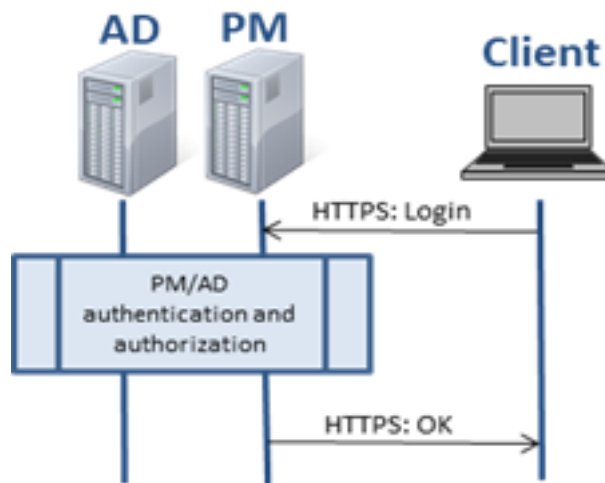


Figure 6.2: User authenticated in AD and authorized in PM user database**Figure 6.3:** The examples above are in the following scenarios referred to as

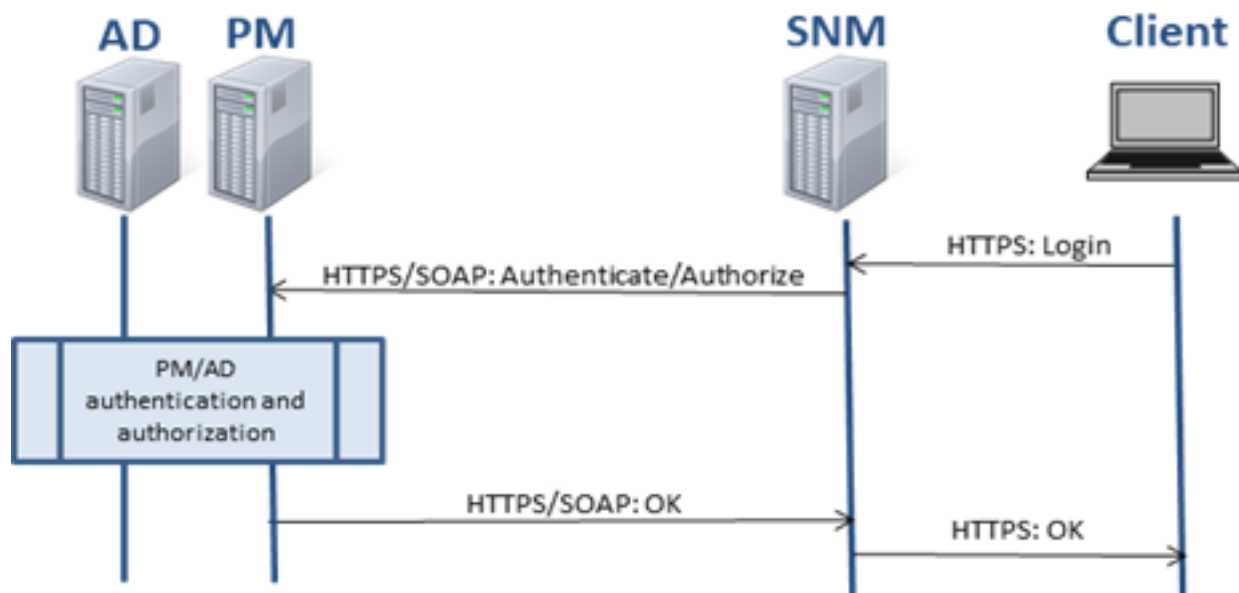
Scenario 1: PM Login

Figure 6.4: Scenario 1: PM Login



Scenario 2: SNM Login

Figure 6.5: Scenario 2: SNM Login



Scenario 3: SNM Login over HTTP

Warning! This scenario shall be avoided!

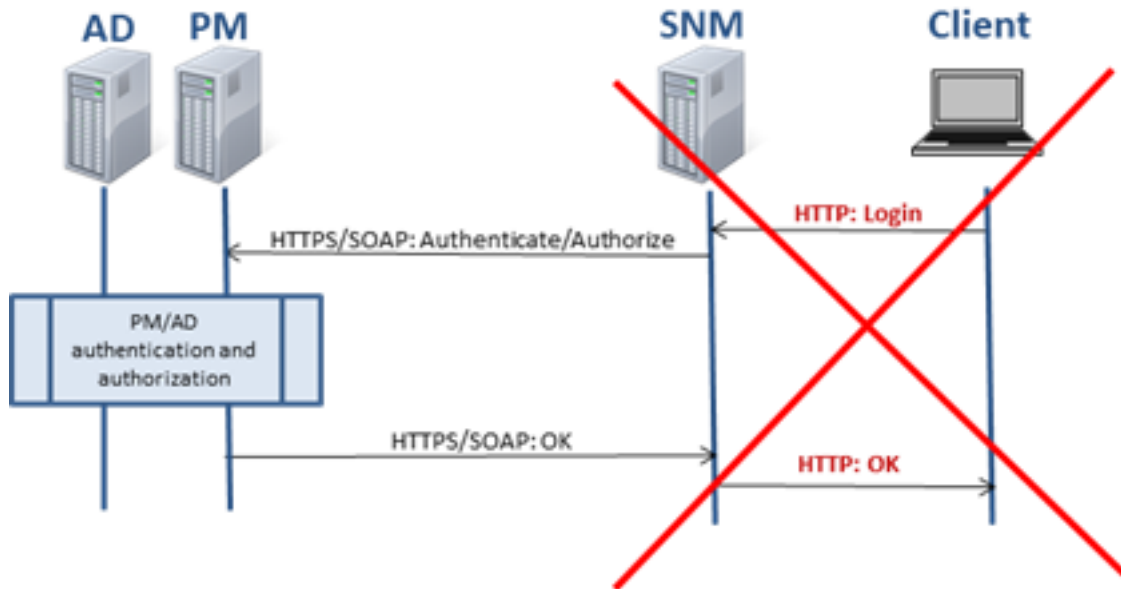
1. PM server is configured for AD authentication.
2. SNM server is configured for PM authentication.

3. SNM server is not configured for SSL.

There is no built in mechanisms to control that SNM systems have been configured to use an PM as authentication server. Therefore it is important that the system administrator makes sure to configure all involved components in the network homogeneously.

The risk is that users get used to log in with AD credentials which basically requires SSL. If the user now connects directly to an SNM system without SSL, it is highly possible that the user still types AD credentials and the user alias and password may be exposed in the network.

Figure 6.6: Scenario 3: SNM Login over HTTP



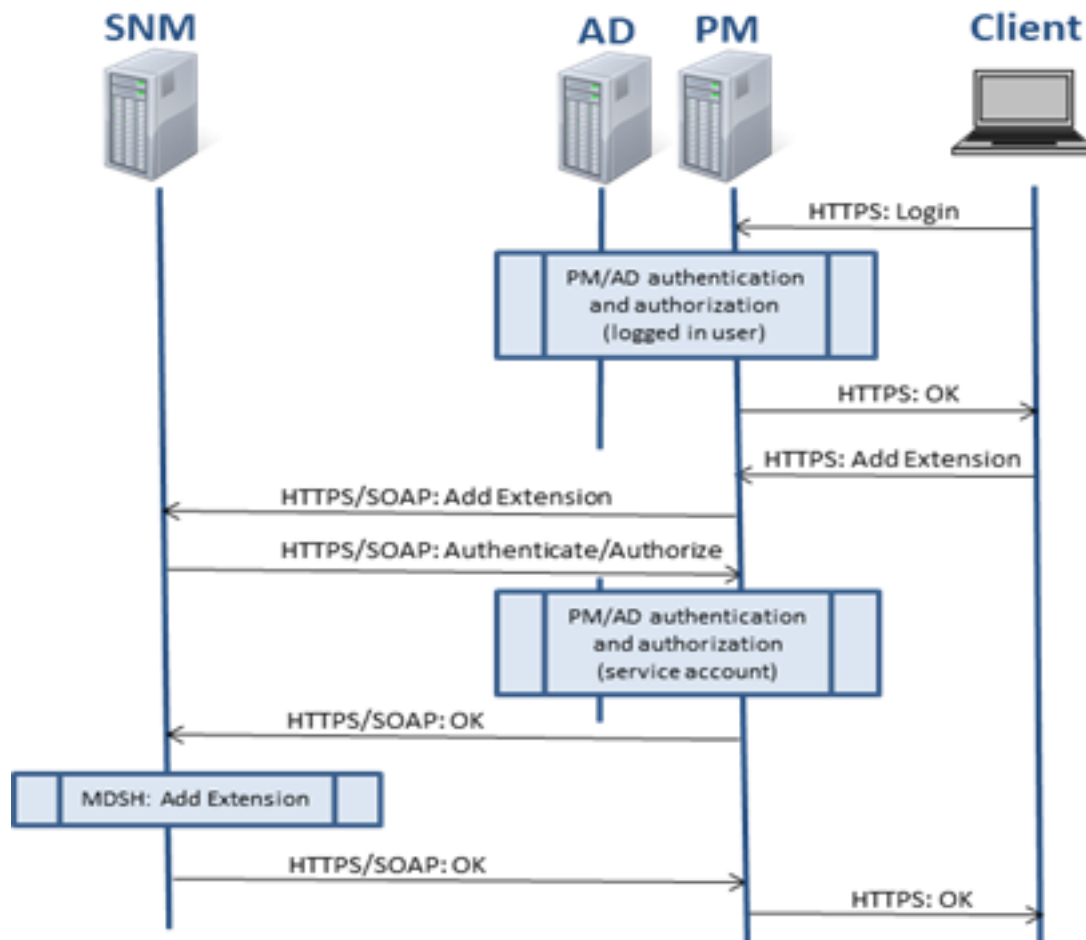
Scenario 4: PM Login + use case 'Add Extension'

A user that logs in to PM will get the privileges he/she is entitled according to the configuration. However, when executing things that requires access to the connected subsystem (SNM server) another account will be used for authentication in the background web services involved. This is referred to as the subsystem "service account".

The service account is the one defined when the subsystem was created. It is a separate account and should not be mixed up with normal accounts. The privileges for the service account must be set "high enough" to serve all end users and administrators.

- **NOTE:** Despite that the "service account" is configured with privileges on a high level, it will not enable any extra features for the logged in PM user.
- It is recommended that the "service account" is created for PM authentication only, and not as an AD user account. Implicit actions in the PM application that require use of the subsystem "service account" will render a lot of authentication requests. It is therefore more efficient to avoid AD authentication.

Figure 6.7: Scenario 4: PM Login + use case ‘Add Extension’

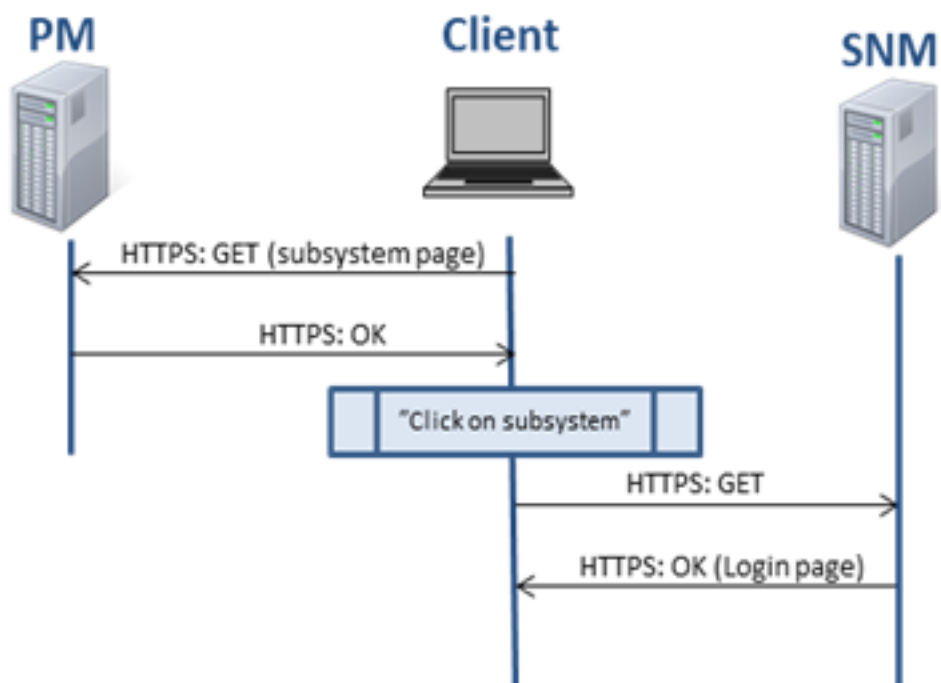


Scenario 5: In PM “click on subsystem”

PM holds a list of connected “subsystems”. It is possible to click on the hyper links to log in to each subsystem. This may result in an automatic log in depending on the configuration.

The behavior for click on subsystem of type SNM is changed when using AD authentication. The automatic login is disabled. Instead the login page for the SNM will be launched. If the SNM server is configured for PM authentication, the same credentials can be used as when logging in to PM.

Figure 6.8: Scenario 5: In PM "click on subsystem"



Fault Cases / Exceptions

Table 7.1: Fault Cases / Exceptions

Fault / Exception	Resulting in	Comments
Server is configured for HTTP (no SSL)	AD authentication will be automatically disabled	Administrator will be notified during the configuration
Wrong password is used in existing or uploaded server certificate	SSL traffic will not work Not possible to even load the web page	The reason will be seen in Jboss server.log as: Error starting endpointjava.io.IOException: failed to decrypt safe contents entry
Root certificate for AD server or (when applicable) domain is missing	All communication over LDAPS with AD server will fail Not possible to log in	The reason will be seen in Jboss server.log as: Message: Exception caught during login: Password Incorrect/Password Required

