SecureLogin 9.0

Security Guide

June, 2021

Legal Notice

Contents

About this Book and the Library

The *Security Guide* is intended to help the SecureLogin administrators with several configurations guidelines. These guidelines can be used to enhance the security of a SecureLogin deployment.

It is recommended that the administrators frequently consult the SecureLogin Documentation and keep up to date on patches and versions of both SecureLogin and the host operating system.

Intended Audience

This book is intended for SecureLogin administrators. It is assumed that you have knowledge of the following:

- Certificate Authority (CA)
- Microsoft Active Directory
- Microsoft Management Console (MMC)
- Microsoft Group Policy Object Management Console (GPMC)
- Microsoft Windows operating systems
- Lightweight Directory Access Protocol (LDAP)
- Secure Socket Layer/Transport Layer Security (SSL/TLS)

Additional Documentation

For the latest version of SecureLogin guides, see the SecureLogin Documentation portal.

Contact Information

We want to hear your comments and suggestions about this book and the other documentation included with this product. You can use the **comment on this topic** link at the bottom of each page of the online documentation, or send an email to Documentation-Feedback@microfocus.com.

For specific product issues, contact Micro Focus Customer Care at https://www.microfocus.com/support-and-services/.

1 Deployment Considerations

This section explains basic considerations to make the SecureLogin deployment more secure.

1.1 Installing SecureLogin In Network Firewall

SecureLogin can be installed within the network firewall as well as outside the network firewall.

1.2 Using AES for SSO Encryption

Select the AES for the SSO data encryption. AES is more advanced and secure than 3DES.

1.3 Enabling Passphrase

A SecureLogin passphrase is a question and response combination used as an alternative form of identity verification. Passphrase functionality protects SecureLogin credentials from unauthorized access and enables users to access SecureLogin in offline mode. Passphrases can also be used as a substitute authentication mode if, for example, a user forgets his or her password. Depending on your preferences, SecureLogin passphrase questions can be generated by the administrator and, or the user.

IMPORTANT: Enabling passphrase provides the following benefits:

1. It disables administrators to impersonate users and gain access to their secret data. Hence, the **Passphrase Hidden** must be used only when the LAN administrators are highly trusted, as getting access to users’ secret data may provide them access to other corporate systems.
2. It provides data encryption between the client and the LDAP server. Without the passphrase feature, the only protection available is ACL protection provided by the Directory server.

On initial login to SecureLogin, all users are requested to save a passphrase response. It is important that this response is easy to recall because it cannot be viewed by anyone.

WARNING: Remember the passphrase answer. You cannot access the answer if you forget it.

To set up a passphrase:

1. Specify a question in the **Enter a question** field.
2. Specify an answer in the **Enter the answer** field.

3. Specify the answer again in the **Confirm the answer** field.
4. Click **OK**. Your passphrase is saved and SecureLogin is installed on the workstation.

You can disable the passphrase policy by deselecting **Use Passphrase Policy** option in the **Advanced Settings** pane of the Administrative Management utility. If a passphrase has previously been configured, this dialog box does not display and the installation is complete.

2 Strengthening Certificates

This section provides information about how to install SecureLogin in the LDAP mode with and without root CA certificates.

- Section 2.1, “Installing SecureLogin In the LDAP Mode With Root CA Certificate,” on page 9
- Section 2.2, “Installing SecureLogin in the LDAP Mode Without Root CA Certificate,” on page 10

2.1 Installing SecureLogin In the LDAP Mode With Root CA Certificate

Perform the following steps to install SecureLogin in the LDAP mode with root CA certificate. For more information, see “Installing, Configuring, and Deploying in an LDAP Environment” in the *SecureLogin 9.0 Installation Guide*.

1. Log in to the workstation as an administrator.
2. Run the `NetIQSecureLogin.exe` file.
3. Accept the license agreement and click **Next**.
4. Select **NetIQ eDirectory with LDAP** as the datastore.
5. Click **Next**.
6. Click **Install**.
7. Click **Next**.
8. In the Custom Setup window, select the features you want to install.
9. Click **Next**.
10. In the **LDAP Server Information** window, specify the server address, port, and the root CA certificate path.

NOTE: SecureLogin supports the following certificate formats:

- BASE64 (*.b64)
- PEM (*.pem)

IMPORTANT: It is mandatory to specify the root CA certificate path when installing SecureLogin in the LDAP mode. Specifying the root CA certificate is also mandatory when migrating to the LDAP mode using `slMigrationHelper.exe`. Although, it is not recommended, if you do not wish to specify the root CA certificate path, see Installing SecureLogin in the LDAP Mode Without Root CA Certificate to install SecureLogin without a certificate.

2.2 Installing SecureLogin in the LDAP Mode Without Root CA Certificate

WARNING: Installing SecureLogin without a root CA certificate makes SecureLogin and the LDAP server open to security threats. It is not recommended to install SecureLogin without the root CA certificate.

Perform one of the following actions to install SecureLogin in the LDAP mode without the root CA certificate:

- Installing SecureLogin in the LDAP Mode Without the Root CA Certificate Using Command Line
- Installing SecureLogin in the LDAP Mode Without the Root CA Certificate Using Response.ini (Silent Installation)

2.2.1 Installing SecureLogin in the LDAP Mode Without the Root CA Certificate Using Command Line

1. Log in as an administrator.
2. Launch the command prompt.
3. Navigate to the location where the SecureLogin installer package is saved.
4. Run the `NetIQSecureLogin.exe` installer file with the `INSTALLWITHOUTCACERT=Yes` parameter. For example:

   ```
   NetIQSecureLogin.exe INSTALLWITHOUTCACERT=Yes
   ```

   NOTE: You can use the `INSTALLWITHOUTCACERT=Yes` parameter and continue the remaining installation with the GUI installer. For example, NetIQSecureLogin.exe /install INSTALLWITHOUTCACERT=Yes.

5. Perform the following steps to modify the registries. The registry modification is necessary to prevent SecureLogin to check for the root CA certificate.

   5a Click **Start > Run** to open the **Run** dialog box.

   5b Specify `regedit` and click **OK** to open **Registry Editor**.

   5c Navigate to the **HKEY_LOCAL_MACHINE > SOFTWARE > Novell > Login > LDAP** key.

   5d Right click and click **New > DWORD**.

   5e Rename the **DWORD** to **CACertNotProvided**.

   5f Edit the `CACertNotProvided` value to `1`.

For more information, see “Installing through the Command Line” in the “*SecureLogin 9.0 Installation Guide*”.

2.2.2 Installing SecureLogin in the LDAP Mode Without the Root CA Certificate Using Response.ini (Silent Installation)

IMPORTANT: Upgrading SecureLogin using the `response.ini` file is not supported.

Perform the following steps to install the SecureLogin in the LDAP mode without the root CA certificate using the `response.ini` file:

1. Log in as an administrator.
2. Specify `INSTALLWITHOUTCACERT=YES` in the `response.ini` file.
3. Launch the command prompt.
4. Navigate to the location where the SecureLogin installer package is saved.
5. To install SecureLogin on all the target machines with the `response.ini` file, run the following command.

   ```
   NetIQSecureLogin.exe /install X_PRIMARYSTORE=LDAP
   PATHTOISS="c:\temp\response.ini" /quiet
   ```

6. Perform the following steps to modify the registries. The registry modification is necessary to prevent SecureLogin to check for the root CA certificate.

   6a Click **Start > Run** to open the **Run** dialog box.

   6b Specify `regedit` and click **OK** to open **Registry Editor**.

   6c Navigate to the **HKEY_LOCAL_MACHINE > SOFTWARE > Novell > Login > LDAP** key.

   6d Right click and click **New > DWORD**.

   6e Rename the **DWORD** to **CACertNotProvided**.

   6f Edit the `CACertNotProvided` value to `1`.

For more information, see "Installing SecureLogin in the LDAP Mode Without the Root CA Certificate Using Responsefile.ini (Silent Installation)" in the "*SecureLogin 9.0 Installation Guide*".

3 Securing the Administration Access

You can make the SecureLogin deployment more secure by restricting the administrative rights of end-users. The following table list the default and the recommended preferences for a more secure deployment. To access these configurations, open SecureLogin and click **Preferences**.

Table 3-1 Preferences

Preferences	Default Value	Recommended
Add application prompts for WindowsAutomation (.Net) applications	Yes	No
Allow “Close” option via system tray	Yes	No
Allow “Refresh Cache” option via system tray	Yes	No
Allow “Log Off” option via system tray	Yes	No
Allow “Work Offline” option via system tray	Yes	No
Allow application definition to be modified by users	Yes	No
Allow application definition to be viewed by users	Yes	No
Allow credentials to be deleted by users through the GUI	Yes	No
Allow credentials to be modified by users through the GUI	Yes	No
Allow users to (de) activate SSO via system tray	Yes	No
Allow users to backup/restore	Yes	No
Allow users to change passphrase	Yes	No
Allow users to modify names of Applications and Logins	Yes	No
Allow users to view and change Preferences	Yes	No
Allow users to view and modify API preferences	Yes	No
Password protect the system tray icon	No	No
Provide API Access	Yes	No
Wizard mode	Yes	No
Add application prompts for Internet Explorer	Yes	No
Add application prompts for Mozilla Firefox	Yes	No
Add application prompts for Google Chrome	Yes	No
Add application prompts for web pages on mutation	No	No

4 Securing the Single Sign-on Data

4.1 Securing the Single Sign-On Data In the Local Cache

Securing the single sign-on (SSO) data in local cache has the following three aspects:

- The local cache file must have a sufficiently restrictive Access Control List (ACL).
- When SecureLogin is installed with the default cache file path, ACL must be set under %LOCALAPPDATA%.
- For a custom cache file path, the administrator must configure ACL for the custom path.

4.2 Securing the SSO Data in a Directory

SecureLogin stores the sensitive data under directory attributes of the user object. Sufficiently restrictive ACL must be configured at the Directory level.

5 Strengthening TLS/SSL Settings

It is recommended to use TLS v1.2 and later for the SecureLogin communication with the following components:

- Directory (Active Directory, eDirectory or LDAP compliant directory)
- Advanced Authentication server
- Privileged Account Manager server
- Syslog server

By default, SecureLogin initiates the SSL connection with the TLS v1.2 protocol. You must configure the same TLS version on the corresponding server.

6 Securing SecureLogin on Docker

In this Chapter

6.1 Securing Kubernetes Clusters

Ensure that you have secured the Kubernetes cluster from any accidental or malicious access. For more information, see Best practices for cluster security and upgrades in Azure Kubernetes Service.

6.2 YAML Best Practices

The values.yaml file contains some sensitive information as part of the installation process. Consider securing the file before storing it anywhere with wider accessibility, such as file share with version control.

The following are the recommendations:

- Do not share the file.
- You must clear the sensitive information, such as password, from values.yaml after the deployment is complete.

  When inputs (password) are sent by using the `--set command`, it is visible when you run the `helm get values <release name>` command.
- Change all default values in the values.yaml file.

6.3 Securing the Stored Data

- Allow access to the SecureLogin Postgres database to only database administrators.
- Ensure that you have secured the database from any accidental or malicious access. For more information, see Security.

6.4 Encrypting the SecureLogin Client Data

The SecureLogin data is encrypted and decrypted only on the client host. When using SecureLogin Advanced Edition, all stored data are encrypted with AES-256.

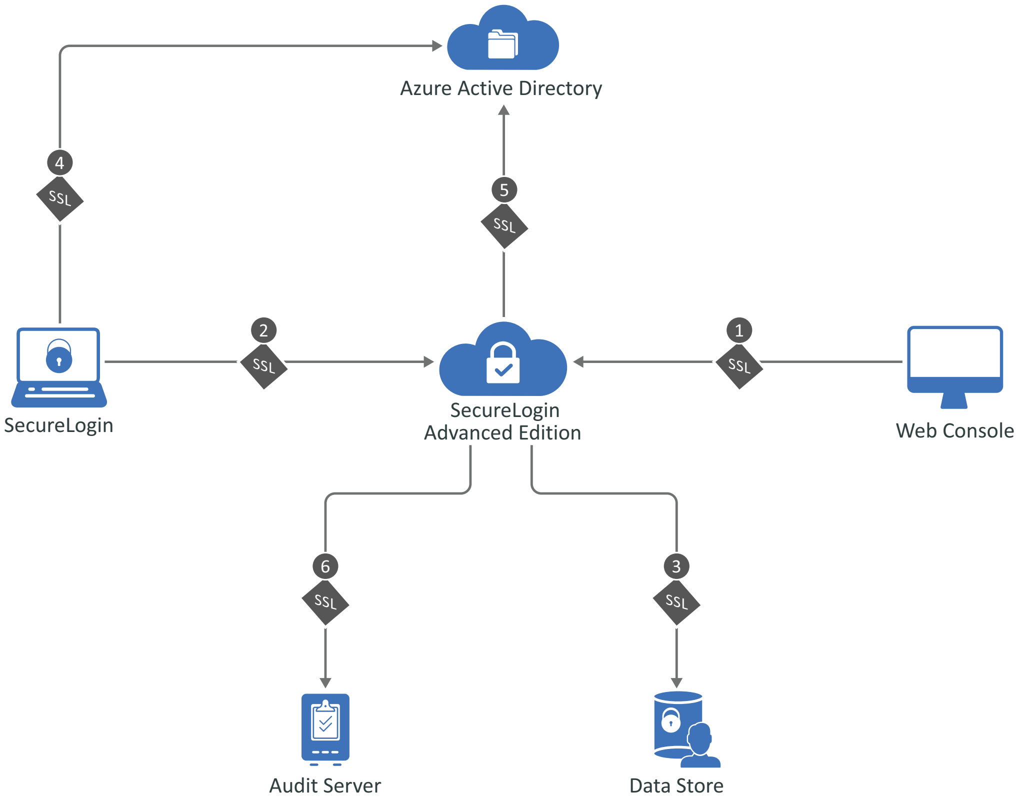
When you migrate the data from an existing setup to SecureLogin Advanced Edition, the imported information is re-encrypted with AES-256 as necessary. You do not need to change the encryption settings before migrating the data.

6.5 Securing Communication Channels

SecureLogin provides the following six communication channels:

1. Web Console to Advanced Edition
2. SecureLogin Client to Advanced Edition
3. Advanced Edition to Data Store
4. SecureLogin Client to Azure AD
5. Advanced Edition to Azure Active Directory (AD)
6. Advanced Edition to Audit Server

Figure 6-1 *SSL Communication Channels*



In this Section

- Protecting the Channel between Advanced Edition and the Web Console with SSL
- Protecting the Channel between Advanced Edition and the SecureLogin Client with SSL

- Protecting the Channel between Advanced Edition and Data Store with SSL
- Protecting the Channel between SecureLogin Client and Azure AD with SSL
- Protecting the Channel between Advanced Edition and Azure AD with SSL
- Protecting the Channel between Advanced Edition and Audit Server with SSL

6.5.1 Protecting the Channel between Advanced Edition and the Web Console with SSL

Channel 1 in Figure 6-1, “SSL Communication Channels,” on page 20.

While installing Advanced Edition, SecureLogin uses *Let's Encrypt* to generate the certificate. You can replace the default certificate with a third-party certificate authority (CA) issued certificate, such as Verisign.

For more information about how to replace the default certificate, see “Using Your CA Signed Certificate” in the *NetIQ SecureLogin 9.0 Administration Guide*.

6.5.2 Protecting the Channel between Advanced Edition and the SecureLogin Client with SSL

Channel 2 in Figure 6-1, “SSL Communication Channels,” on page 20.

The SecureLogin client communicates with Advanced Edition over a secure SSL port. This port is configured while installing the client in the Advanced Edition mode.

For higher security, it is recommended to install the root CA certificate on the SecureLogin client machines.

6.5.3 Protecting the Channel between Advanced Edition and Data Store with SSL

Channel 3 in Figure 6-1, “SSL Communication Channels,” on page 20.

For higher security, configure the data store root certificate in `SecureLogin-Server-x.x.x.x\values.yaml`.

Configuring the Datastore Root Certificate

1. Create a folder named `certs` inside the `SecureLogin-Server-x.x.x.x` folder of the helm charts.
2. Copy the datastore root certificate to the `certs` folder.
3. Open `SecureLogin-Server-x.x.x.x\values.yaml` and specify the following details in the **SSL** section:
   - **verifyDBCert**: Specify `true`. The server host name in the certificate is verified to ensure that it matches the host name specified in `values.yaml`.

- **DBCert**: Specify the name of the data store root certificate that you copied to the `certs` folder.
- **DBCertSecret**: To change the certificate for the first time, no need to change this value. However, the next time onward, you must change both **DBCert** and **DBCertSecret**.

4 Save the file.

5 Perform a helm install or upgrade using the following command:

- To install:

  ```
  helm install <name-of-the-release> <name-of-the-helm-chart> -n
  <name-of-the-namespace>
  ```

  For example, `helm install slserver001 server -n nsl-namespace`

- To upgrade:

  ```
  helm upgrade <release-name> <name-of-the-helm-chart> -n <name-of-
  the-namespace>
  ```

  For example, `helm upgrade slserver server -n my-ingress`

6.5.4 Protecting the Channel between SecureLogin Client and Azure AD with SSL

Channel 4 in Figure 6-1, "SSL Communication Channels," on page 20.

The SecureLogin client communicates to the identity provider over the TLS 1.2 protocol with the verified server certificate.

This channel is used for obtaining OAuth 2.0 tokens from the identity provider. The SecureLogin client uses the following methods:

- Initial request using Microsoft OAuth 2.0 Resource Owner Password Credentials (ROPC) grant. The ROPC flow is a single request. It sends the client identification and user's credentials to Azure AD for fetching the required token. For more information, see Microsoft identity platform and OAuth 2.0 Resource Owner Password Credentials.
- Refresh request using a refresh token: To improve the performance, refresh tokens can be enabled by adding the offline_access scope. For example, "api://$clientid/All offline_access"

  The offline_access scope is optional if you want users to receive a refresh token.

  For more information, see Microsoft identity platform and OAuth 2.0 Resource Owner Password Credentials and Resource owner password credentials policy.

6.5.5 Protecting the Channel between Advanced Edition and Azure AD with SSL

Channel 5 in Figure 6-1, "SSL Communication Channels," on page 20.

Advanced Edition communicates to the user and groups directory and identity provider over the TLS 1.2 protocol with the verified server certificate.

Advanced Edition uses the bearer access token to verify the authorization with the identity provider. It requests for an On Behalf Of (OBO) token using the MSAL library or using Graph REST API.

- Using MSAL library: With the OBO token, Advanced Edition requests the information about users and groups for collecting inherited settings or performing management tasks.
- Using Graph REST API: Graph API ensures the secure delivery of the data to Azure AD. For more information, see Microsoft Graph Documentation.

6.5.6 Protecting the Channel between Advanced Edition and Audit Server with SSL

Channel 6 in Figure 6-1, "SSL Communication Channels," on page 20.

For enhanced security, configure the audit server root certificate in `SecureLogin-Server-x.x.x.x\values.yaml`.

For information about how to configure the audit server root certificate in values.yaml, see "Configuring the Root Certificate in values.yaml" in the *SecureLogin 9 Advanced Edition Installation and Configuration Guide*.

Ensure that the audit server is configured to listen on a TLS port. For information about how to configure the port for the audit server, see "Configuring the Audit Server on the Web Console" in the *SecureLogin 9 Advanced Edition Installation and Configuration Guide*.

6.6 Protecting SecureLogin Secrets on Kubernetes

SecureLogin uses helm charts to create the secrets. The data of the secrets are Base64 encoded. These are sensitive and critical data. To protect these data, access to the Kubernetes environment should be limited to only trusted users, such as administrators.

6.7 Securing Web Console Access

Access to the SecureLogin Advanced Edition web console and REST APIs are limited to the admin user as specified in the helm chart.

The following are possible configuration options:

- A single admin user name and password configured while deploying Advanced Edition.
- The access token timeout and unique encryption secret. This also allows the administrative API to be also shared across the pods.

Configuring the Access Token

When a user tries to access the web console and provides valid credentials, system returns a JWT token and the access is granted. This token is encrypted using the JWT token secret. You can configure this secret and the expiration time of the token in `SecureLogin-Server-x.x.x.x\values.yaml`.

For information about how to change the expiration time, see "Modifying the Life Span of a JWT Token" in the *SecureLogin 9 Advanced Edition Installation and Configuration Guide*.

To secure the web console access, consider the following best practices for the JWT token secret:

- The value must contain alphanumeric characters and symbols.
- The length must 64 characters.
- The value must be changed while setting up the server for the first time and must be changed periodically later.

Changing the Secret of the JWT Token

1. Open `SecureLogin-Server-x.x.x.x\values.yaml`.
2. Change the value of **secret** in the **JWTToken** section.
3. Perform a helm install or upgrade using the following command:
   - To install:

     ```
     helm install <name-of-the-release> <name-of-the-helm-chart> -n
     <name-of-the-namespace>
     ```

     For example, `helm install slserver001 server -n nsl-namespace`
   - To upgrade:

     ```
     helm upgrade <release-name> server -n <name-of-the-namespace>
     ```

     For example, `helm upgrade slserver server -n my-ingress`

6.8 Securing Access to Advanced Edition

You can configure to allow access to the administration REST API only for the requests coming from a machine with the specified IP addresses. This configuration prevents unauthorized and malicious access attempts.

Perform the following steps to restrict access based on the IP address:

1. Log in to the Azure portal.
2. Select **Resource groups**.
3. In the resource groups list, find the relevant resource group in the following format:

   ```
   MC_<your-resource-group-name>_<aks-cluster-name>_<geo location>
   ```
4. In the selected resource group, select **Network Security Group**. A list of inbound security rules is displayed.
5. Edit the security rule with port 443.
6. Change the **Source** to **IP addresses**.
7. Specify the comma-delimited list of IP addresses or IP range in **Source IP addresses**. For example:

   `192.168.0.101`

   `192.168.0.101, 192.168.0.156` (two IP addresses)
8. Click **Save**. Changes might take 1 or 2 minutes to take effect.

7 Restoring Previous Security Level After Upgrade

All protocols, ciphers, and configurations in all components are highly secure by default in SecureLogin 8.7 and later. If your SecureLogin deployment is configured with less secure settings, upgrading it to higher and then downgrading it back is not supported. The following are a few example scenarios.

- Downgrading the SSO data from AES to 3DES is not supported. When the SSO data is encrypted using AES, downgrading it to 3DES might lead to data loss.
- From SecureLogin 8.7, SHA1 is replaced with SHA256 as the default hashing algorithm. SHA256 is more secure and trustworthy. SHA1 to SHA256 is a seamless migration and is available only if you were already using the default AES encryption.

  IMPORTANT: If you are using 3DES encryption, then upgrading to SecureLogin 8.7 will not encrypt single sign-on data to SHA256. It remains in SHA1.

- After you install SecureLogin 8.7 that includes SHA256, you cannot downgrade to lower SecureLogin version which includes SHA1. If you downgrade from SecureLogin 8.7 to a previous version, SecureLogin will stop working. This issue occurs because the lower versions of SecureLogin can only process the SHA1 encryption, it does not process the SHA256 encrypted single sign-on data.