

opentext™

# Access Manager CE 24.2 (v5.1) Security Guide

May 2024

## Legal Notice

### Copyright 2009 - 2024 Open Text.

The only warranties for products and services of Open Text and its affiliates and licensors ("Open Text") are as may be set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Open Text shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

For additional information, such as certification-related notices and trademarks, see <https://www.microfocus.com/en-us/> (<https://www.microfocus.com/en-us/>).

---

# Contents

<b>About this Book and the Library</b>	<b>9</b>
<b>1 Deployment Considerations</b>	<b>11</b>
1.1 Protecting Access Manager through Firewall	11
1.1.1 Access Gateway and Identity Server in DMZ	11
1.1.2 A Firewall Separating Access Manager Components from LDAP Servers	12
1.2 Protecting Access Manager Setup behind NAT	12
1.3 Protecting Identity Server behind Access Gateway	12
1.4 Configuring Identity Server to Listen on Port 443	13
<b>2 Securing Administration Console</b>	<b>15</b>
2.1 Restricting Administration Console Access to only Private Network	15
2.2	Managing Administration Console Session Timeout
2.3 Securing iManager Login Settings	16
2.4	Securing Administrator Accounts
2.4.1 Enforcing Case-Sensitive Universal Password For Access Manager	17
2.5	Protecting the Configuration Store
2.6 Securing Configuration Store Using TLS Port	19
2.7 Running the DHost HTTP Server on localhost	19
2.8 Preventing the SWEET32 Attack	20
2.9 Default Security Settings in Configuration Files	20
2.9.1 server.xml	20
2.9.2 web.xml	21
2.9.3 tomcat8.conf	21
<b>3 Securing Identity Server</b>	<b>23</b>
3.1 Disabling Unused Authentication Protocols	23
3.2 Configuring SSL Communication between Browsers and Identity Server	23
3.3 Configuring SSL Communication with Identity Server and a Service Provider	24
3.4 Securing Authentication by Using Strong and Multi-Factor Authentication Methods	24
3.5 Securing Federation	25
3.5.1 Setting Options	25
3.5.2 Configuring the Encryption Method for the SAML Assertion	26
3.6 Configuring a Whitelist of Target URL	26
3.6.1 Configuring a Global Whitelist of Target URL	27

3.6.2	Configuring a Whitelist of Intersite Transfer Service Target URL	27
3.6.3	Configuring a Whitelist of Assertion Consumer Service URL	27
3.6.4	Configuring a Whitelist of URLs for RelayState URL	28
3.7	Blocking Access to Identity Server Pages	29
3.8	Enabling Advanced Session Assurance	29
3.9	Securing Identity Server Web Service Interface	30
3.10	Enabling reCAPTCHA	30
3.11	Preventing the SWEET32 Attack	31
3.12	Detecting the Brute Force Attack	31
3.13	Restricting the Direct Access to Files in the nidp Folder	31
3.14	Preventing Cross-Site Request Forgery Attacks	32
3.15	Using the Device ID in the URN Cookie	34
3.16	Configuring the Cookie Secure Flag	34
3.17	Default Security Settings in Configuration Files	34
3.17.1	server.xml	34
3.17.2	web.xml	35
3.17.3	tomcat.conf	35

## 4 Securing Access Gateway 37

4.1	Enabling SSL Communication between Access Gateway and Identity Server	37
4.2	Enabling Secure Cookies	37
4.2.1	Securing the Embedded Service Provider Session Cookie	37
4.2.2	Securing the Proxy Session Cookie	38
4.3	Disabling Phishing	39
4.4	Disabling Weak Protocols between Access Gateway and Web Servers	39
4.5	Configuring Stronger Ciphers for SSL Communication between Access Gateway and Web Servers	40
4.6	Enabling Perfect Forward Secrecy	40
4.7	Preventing Error Messages to Show the Failure Reason on Browsers	41
4.8	Enabling Advanced Session Assurance	41
4.9	Preventing the Slowloris Attack	41
4.10	AJP Communication Setting for Access Gateway	42
4.11	Using the Device ID in the URN Cookie	43
4.12	Default Security Settings in Configuration Files	44
4.12.1	ESP web.xml	44
4.12.2	Access Gateway Advanced Options	44
4.12.3	httpd.conf	44
4.12.4	NovellAgSettings.conf	45

<b>5</b>	<b>Securing Analytics Server</b>	<b>47</b>
5.1	Customizing the Size of EDH Keys	47
5.2	Configuring SSL in Analytics Server	47
5.3	Disabling SSL Renegotiations	47
5.4	Securing Analytics Server Cluster Communication	48
5.5	Setting Analytics Dashboard Timeout	48
5.6	Default Security Settings in Configuration Files	48
5.6.1	server.xml	48
5.6.2	web.xml	49
<b>6</b>	<b>Hardening Appliance</b>	<b>51</b>
6.1	Disabling Weak MAC Algorithms on a Secure Shell Server	51
<b>7</b>	<b>Configuring Secure Communication</b>	<b>53</b>
7.1	Configuring SSL in Identity Server	54
7.1.1	Configuring a SSL Channel between Identity Server and LDAP Servers	54
7.1.2	Enabling SSL between Browsers and Identity Server	54
7.1.3	Enabling SSL between Identity Server and a Service Provider	55
7.2	Configuring SSL in Access Gateway	55
7.2.1	Enabling SSL between Browsers and Access Gateway	56
7.2.2	Enabling SSL between Access Gateway and Web Servers	57
7.3	Configuring SSL for Authentication between Identity Server and Access Gateway	58
7.4	Configuring SSL in Analytics Server	58
7.5	Using Trusted Certificates Authority	60
<b>8</b>	<b>Strengthening TLS/SSL Settings</b>	<b>61</b>
8.1	Disabling SSLv2 and SSLv3 Protocols	61
8.2	Optimizing SSL Configuration with Ciphers	62
8.3	Enabling Perfect Forward Secrecy	62
8.4	Adding HTTP Strict Transport Security	62
8.5	Disabling SSL Renegotiations	63
8.6	Customizing the Size of Ephemeral Diffie-Hellman Keys	63
<b>9</b>	<b>Strengthening Certificates</b>	<b>65</b>
9.1	Key Size and Signature Algorithm Considerations	65
9.2	Trusted Certificate Authorities	65
9.3	Certificate Renewal	66
<b>10</b>	<b>XSS, XFS, and Clickjacking Attacks</b>	<b>67</b>
10.1	Cross-site Scripting Attacks	67
10.1.1	Option 1: HTML Escaping	67
10.1.2	Option 2: Filtering	68
10.1.3	Option 3: Understanding Relaxed Query Parameters	70
10.2	Cross-Frame Scripting Attacks	70
10.3	Clickjacking Attacks	70

<b>11 Getting the Latest Security Patches</b>	<b>71</b>
<b>12 Securing Access Manager Components on Cloud</b>	<b>73</b>
12.1 Prerequisite	73
12.2 Protecting Administration Console on Cloud	73
<b>13 Restoring Previous Security Level After Upgrading Access Manager</b>	<b>75</b>
13.1 Restoring Previous Security Settings for Administration Console	76
13.1.1 Restoring the Previous Protocols Settings	76
13.1.2 Restoring the Previous Settings of Ciphers for SSL Communication	76
13.1.3 Disabling Perfect Forward Secrecy	77
13.1.4 Restoring the Previous Size of EDH Keys	77
13.1.5 Removing HTTP Strict Transport Security	77
13.2 Restoring Previous Security Settings for Identity Server	78
13.2.1 Restoring the Previous Protocols Settings	78
13.2.2 Restoring the Previous Settings of Ciphers for SSL Communication	78
13.2.3 Disabling Perfect Forward Secrecy	79
13.2.4 Restoring the Previous Settings of the Size of EDH Keys	79
13.2.5 Removing HTTP Strict Transport Security	79
13.2.6 Removing the Clickjacking Filter	80
13.3 Restoring Previous Security Settings for Access Gateway	80
13.3.1 Restoring the Previous Protocol Settings between Browsers and Access Gateway	80
13.3.2 Restoring the Previous Ciphers Settings between Browsers and Access Gateway	81
13.3.3 Removing the Clickjacking Filter	81
13.3.4 Removing HTTP Strict Transport Security	82
<b>A Appendix</b>	<b>83</b>
A.1 Default Ciphers for Identity Server	83
A.2 Default Ciphers for Administration Console	83
A.3 Default Ciphers for Analytics Server	84
A.4 Default Ciphers for Access Gateway	84

# About this Book and the Library

The *Security Guide* is intended for Access Manager administrators, designers, and implementers with several configuration guidelines. These guidelines can be used for enhancing the security of an Access Manager environment. The first half of the guide focuses on tasks for configuring the Access Manager components along with examples and references. The remaining part of the guide provides additional information about the important concepts described in prior sections.

It is recommended that the administrators frequently consult the product documentation (listed in “Other Information in the Library”), Access Manager TIDS, Cool Solutions, and keep up to date on patches and versions of both Access Manager and the host operating system.

## Intended Audience

This book is intended for Access Manager administrators. It is assumed that you have knowledge of evolving Internet protocols, such as:

- ◆ Extensible Markup Language (XML)
- ◆ Simple Object Access Protocol (SOAP)
- ◆ Security Assertion Markup Language (SAML)
- ◆ Public Key Infrastructure (PKI) digital signature concepts and Internet security
- ◆ Secure Socket Layer/Transport Layer Security (SSL/TLS)
- ◆ Hypertext Transfer Protocol (HTTP and HTTPS)
- ◆ Uniform Resource Identifiers (URIs)
- ◆ Domain Name System (DNS)
- ◆ Web Services Description Language (WSDL)

## Other Information in the Library

The library provides the following information resources:

- ◆ [Access Manager Product Documentation \(https://www.microfocus.com/documentation/access-manager/index.html\)](https://www.microfocus.com/documentation/access-manager/index.html)
- ◆ [Access Manager Developer Resources \(https://www.microfocus.com/documentation/access-manager/developer-documentation-5.0/\)](https://www.microfocus.com/documentation/access-manager/developer-documentation-5.0/)





# 1 Deployment Considerations

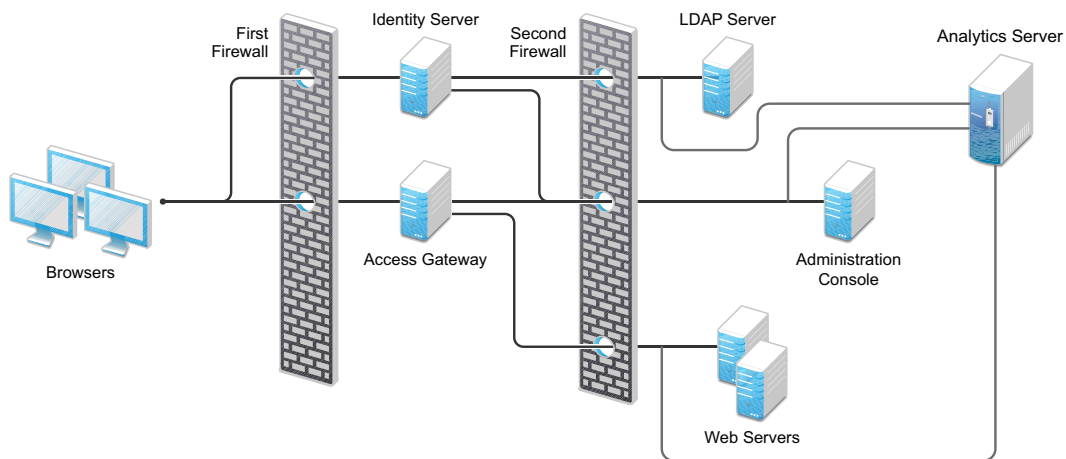
## In this Chapter

- ◆ Protecting Access Manager through Firewall
- ◆ Protecting Access Manager Setup behind NAT
- ◆ Protecting Identity Server behind Access Gateway
- ◆ Configuring Identity Server to Listen on Port 443

## 1.1 Protecting Access Manager through Firewall

Use Access Manager with firewalls. [Figure 1-1](#) illustrates a firewall setup for a basic Access Manager configuration of Identity Server, Access Gateway, Analytics Server, and Administration Console.

*Figure 1-1 Access Manager Components between Firewalls*



### 1.1.1 Access Gateway and Identity Server in DMZ

**First Firewall:** If you place a firewall between browsers and Access Gateway and Identity Server, you need to open ports so that browsers can communicate with Access Gateway and Identity Server and Identity Server can communicate with other identity providers.

For information about ports required to open in the first firewall, see “[First Firewall](#)” in the [NetIQ Access Manager CE 24.2 \(v5.1\) Installation and Upgrade Guide](#).

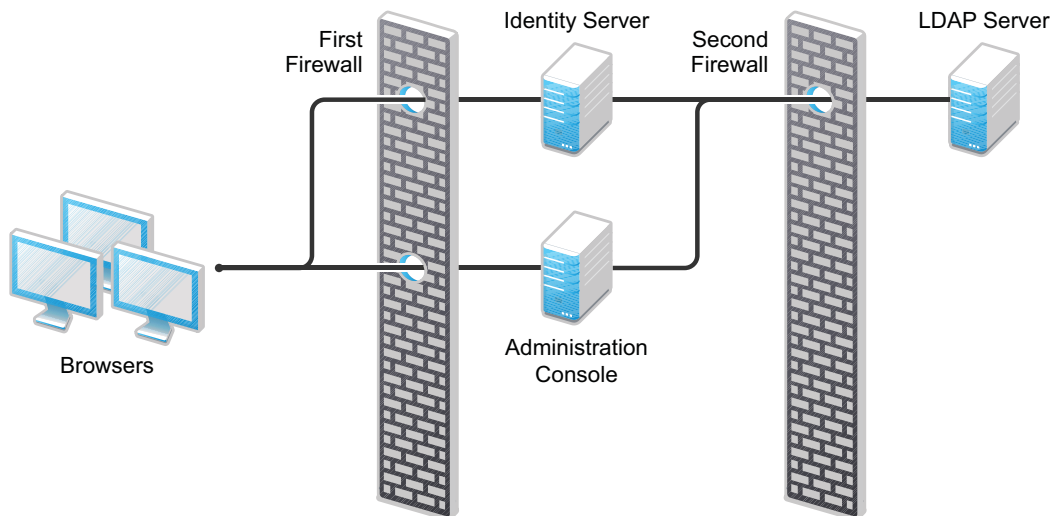
**Second Firewall:** The second firewall separates web servers, LDAP servers, Analytics Server, and Administration Console from Identity Server and Access Gateway.

For information about ports required to open in the second firewall, see “[Second Firewall](#)” in the [NetIQ Access Manager CE 24.2 \(v5.1\) Installation and Upgrade Guide](#).

## 1.1.2 A Firewall Separating Access Manager Components from LDAP Servers

You can configure your Access Manager components so that Administration Console is on the same side of the firewall as other Access Manager components and have a firewall between them and LDAP servers.

*Figure 1-2 A Firewall Separating Administration Console and the LDAP Server*



In this configuration, you need to open the required ports in the second firewall for Administration Console and Identity Server.

For information about all required ports, see [“Required Ports”](#) in the [NetIQ Access Manager CE 24.2 \(v5.1\) Installation and Upgrade Guide](#).

## 1.2 Protecting Access Manager Setup behind NAT

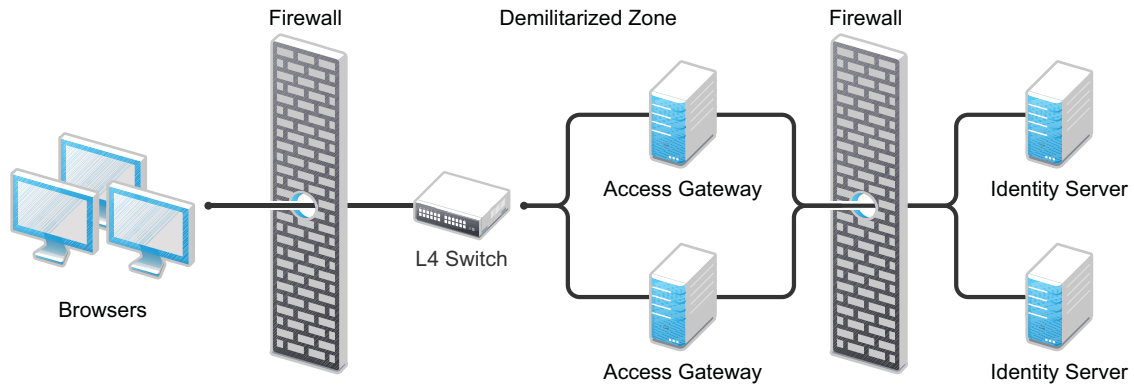
You can configure Access Manager by using Network Address Translation (NAT) to enable the communication between Administration Console from local network to other Access Manager devices such as Identity Server and Access Gateway. The devices can be in the external network or in another private network. The NAT address needs be to configured in router.

For information about how to configure Access Manager behind NAT, see [Installing Access Manager Components in NAT Environments](#).

## 1.3 Protecting Identity Server behind Access Gateway

You can configure Access Manager to protect Identity Server behind Access Gateway. This configuration reduces the number of ports you need to open between the outside world and your network. The following diagram illustrates such a configuration:

**Figure 1-3** Identity Servers behind an Access Gateway



With this configuration, you need an L4 switch to cluster Access Gateways. However, you do not need an L4 switch to cluster Identity Servers. When Identity Server is configured to be a protected resource of Access Gateway, Access Gateway uses its web server communication channel. Each Identity Server in the cluster must be added to the web server list, and Access Gateway uses its web server load balancing and failover policies for the clustered Identity Servers.

**Limitations:** The following features are not supported with this configuration:

- ◆ Identity Server cannot respond to identity provider introductions.
- ◆ Federation to an external service provider that requires the artifact profile with SOAP/Mutual SSL binding cannot be supported with this configuration.
- ◆ The proxy service that is protecting Identity Server cannot be configured to use mutual SSL. For example, X.509 authentication cannot be used for any proxy service. To perform X.509 authentication (which is a form of mutual SSL), a user's browser must have direct access to Identity Server.
- ◆ The proxy service that is protecting Identity Server cannot be configured to use NMAS.

For configuration details, see [Configuring a Protected Identity Server Through Access Gateways](#) in the [NetIQ Access Manager CE 24.2 \(v5.1\) Administration Guide](#).

## 1.4 Configuring Identity Server to Listen on Port 443

Identity Server by default listens on port 8443. It requires port 8443 to be opened in firewall for the communication between a browser and Identity Server. To avoid opening 8443 port in firewall, you can configure Identity Server to listen on standard port 443.

The Identity Server service (hosted on Tomcat) runs as a non-privileged user and cannot bind to ports below 1024. To allow requests to port 80/443 while Tomcat is listening on 8080/8443, use iptables to perform a port translation. Port translation allows the base URL of Identity Server to be configured for port 443 and to listen on this port. iptables translates it to port 8443 when communicating with Tomcat.

The following are two of many possible solutions:

- ♦ **Simple iptables Script:** If you have disabled the SUSE Linux Enterprise Server (SLES) firewall and do not have any other Access Manager components installed on the Identity Server machine, you can use a simple iptables script to translate the ports. See [“Configuring a Simple Redirect Script”](#) in the *NetIQ Access Manager CE 24.2 (v5.1) Installation and Upgrade Guide*.
- ♦ If you have configured the SLES firewall or have installed other Access Manager components on Identity Server, you use a custom rule script that allows for multiple port translations. See [“Configuring iptables for Multiple Components”](#) in the *NetIQ Access Manager CE 24.2 (v5.1) Installation and Upgrade Guide*.

# 2 Securing Administration Console

Administration Console contains configuration information for all Access Manager components. If you federate your users with other servers, it stores configuration information about these users. You need to protect Administration Console so that unauthorized users cannot change configuration settings or gain access to the information in the configuration store.

When you develop a security plan for Access Manager, consider the following considerations:

- ◆ [Restricting Administration Console Access to only Private Network](#)
- ◆ [Managing Administration Console Session Timeout](#)
- ◆ [Securing iManager Login Settings](#)
- ◆ [Securing Administrator Accounts](#)
- ◆ [Protecting the Configuration Store](#)
- ◆ [Securing Configuration Store Using TLS Port](#)
- ◆ [Running the DHost HTTP Server on localhost](#)
- ◆ [Preventing the SWEET32 Attack](#)
- ◆ [Default Security Settings in Configuration Files](#)

## 2.1 Restricting Administration Console Access to only Private Network

Sometimes you might need to install Administration Console with multiple IP address. For example, when you install Administration Console and Identity Server on the same machine. Identity Server must be accessible and the services provided by Access Manager must be available on the Internet. This might cause a security issue with Administration Console.

Perform the following steps to secure Administration Console in this scenario:

- 1 Open the Administration Console `server.xml` file.
- 2 Search for the context path `nps`.

```
<Context docBase="/opt/novell/nam/adminconsole/webapps/nps" path="/nps"
reloadable="false"
<Valve className="org.apache.catalina.valves.RemoteAddrValve"
allow="xxx.yyy.zzz.www"/>
```

- 3 If the context path exists, then modify or update the `allow` value with the host entries, if required.
- 4 If the context path does not exist, add the following lines before the end of the `</Host>` block:

```
<Context path="/nps">
    <Valve className="org.apache.catalina.valves.RemoteAddrValve"
        allow="xxx.yyy.zzz.www"/>
</Context>
```

The syntax for the allow directive, which can also be changed to a deny directive, is a comma-separated IP regular expressions list (Perl regex format). A simple example is as follows:

```
allow="192.168.10[1-3].[0-9]*"
```

This allows you to access Administration Console using the following IP addresses:

```
192.168.101.0/24, 192.168.102.0/24, 192.168.103.0/24.
```

If you write the syntax as follows:

```
deny=="192.168.10[1-3].[0-9]*"
```

Administration Console access is blocked from the following IP address:

```
192.168.101.0/24, 192.168.102.0/24, 192.168.103.0/24
```

For information about how to modify a file, see [“Modifying Configurations”](#) in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

---

**NOTE:** This file is specific to each cluster. Therefore, while applying the changes from this file, the keystore password is retained in each cluster.

---

If you write the syntax as follows:

```
allow="164.99.232.78|10.71.142.5"
```

Administration Console allows access to Analytics Dashboard through direct URL.

## 2.2 Managing Administration Console Session Timeout

The default Administration Console session timeout value is 30 minutes. You can modify this value in the `web.xml` file for a longer or a shorter period based on your security needs.

- 1 Open the Administration Console `web.xml` file.
- 2 Search for the `<session-timeout>` parameter.
- 3 Modify the value and save the file.

For information about how to edit a file, see [“Modifying Configurations”](#) in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

## 2.3 Securing iManager Login Settings

The default settings of Administration Console login by using iManager are changed in Access Manager 4.1 to ensure higher security. If you upgrade Access Manager from a previous version, you need to manually change the default iManager settings.

To change the default settings in Administration Console, perform the following steps:

- 1 Click **Administration Console > Configure > iManager Server > Configure iManager > Authentication**.
- 2 Make the following changes:
  - ◆ De-select **Remember login credentials (except password)**.
  - ◆ Select **Hide specific reason for login failure**.

## 2.4 Securing Administrator Accounts

The admin user you create while installing Administration Console has all rights to Access Manager components. We recommend that you secure this account through the following configuration:

- ♦ **Password Restrictions:** When the admin user is created, no password restrictions are set. To ensure that the password meets your minimum security requirements, configure the standard eDirectory password restrictions for this account. In Administration Console, select the **Roles and Tasks** view in the iManager header, then click **Users**. Browse to the admin user (found in the novell container), then click **Restrictions**.

The password is not case-sensitive by default. To make your password case-sensitive, see [Section 2.4.1, “Enforcing Case-Sensitive Universal Password For Access Manager,” on page 17.](#)

- ♦ **Intruder Detection:** The admin user is created in the novell container. You should set up an intruder detection policy for this container. In Administration Console, select the **Roles and Tasks** view in the iManager header, then click **Directory Administration > Modify Object**. Select **novell**, then click **OK**. Click **Intruder Detection**.
- ♦ **Backup Admin User Creation:** Only one admin user is created when you install Access Manager. If you forget the username or password, you cannot access Administration Console. It is recommended that you create a backup user who has the required privileges of an admin user. For more information, see [“Managing Administrators” in the NetIQ Access Manager CE 24.2 \(v5.1\) Administration Guide.](#)
- ♦ **Delegated Administrators:** If you create delegated administrators for policy containers, ensure that they have sufficient rights to implement a cross-site scripting attack using the Deny Message in an Access Gateway Authorization policy.

They are also granted rights to the LDAP server, which gives them sufficient rights to access the configuration datastore with an LDAP browser. Modifications done with an LDAP browser are not logged by Access Manager.

### 2.4.1 Enforcing Case-Sensitive Universal Password For Access Manager

Making the passwords case-sensitive adds to the security of the login to Access Manager. For example, if you have a password aBc that is case-sensitive, all the trials of login with the combinations like abc or Abc or ABC would fail.

#### Procedure for Setting Your Password Case-Sensitive

- 1 Log into Administration Console.
- 2 Click **username** on the top right corner of the page.
3. **Configuring iManager:**
  - 3a Click **Configure Console**.
  - 3b Click **iManager Server > Configure iManager**.
  - 3c Select **Plug-in Download** and select the **Query Download site for new netiq Plug-in Modules (NPM)**.
  - 3d Click **Save**.

#### 4. Installing Plug-in:

- 4a Click **Plug-in Installation > Available netiq Plug-in Modules**.
- 4b Select **netiq imanager Password Management**.
- 4c Click **Install**.
- 4d Click **I agree** and click **OK**.
- 4e Restart Administration Console after the installation is complete.
- 4f Log into Administration Console.
- 4g Click `<username>` from the top right corner of the page.

#### 5. Assigning a Policy to an Object:

- 5a Click **Manage Roles & Tasks**.
- 5b Select **Passwords > Password Policies**.
- 5c Click **None** from **Assignments** column entry.
- 5d Browse to select an object.
- 5e Click **Apply**. You can see that **novell** is reflecting in the Assignments columns now.

---

**NOTE:** When you log into Administration Console for the first time after setting the password policy and specify a new case-sensitive password, that password becomes your new password.

---

## 2.5 Protecting the Configuration Store

The configuration store is an embedded, modified version of eDirectory. It is backed up and restored with command line options, which back up and restore the Access Manager configuration objects in the `ou=accessManagerContainer.o=novell` object.

You should back up the configuration store on a regular schedule, and store the ZIP file created in a secure place. See “[Back Up and Restore](#)” in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

In addition to backing up the configuration store, you should also install at least two Administration Consoles (a primary and a secondary). If the primary console goes down, the secondary console can keep the communication channels open between the various components. You can install up to three Administration Consoles. See “[Installing Secondary Administration Console](#)” in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

It is not recommended to use the configuration store as a user store.



## 2.6 Securing Configuration Store Using TLS Port

By default the Access Manager config store has FIPS mode enabled and an RSA certificate associated with it. This disables SSLv3 and allows only TLS 1.0, 1.1 and 1.2 clients to connect.

To allow Administration Console to connect with config store on TLSv1.1 and TLSv1.2, perform the following steps:

- 1 Install the LDAP plug-in to list it in the default iManager page (**Roles and Tasks**).
- 2 Ensure FIPS mode is enabled.

Ensure the line `n4u.server.fips_tls=1` is in the `/etc/opt/novell/eDirectory/conf/nds.conf` file.

---

**NOTE:** After enabling FIPS mode, you must restart eDirectory (nds) daemon.

---

- 3 Click **Admin > Manage Roles and Tasks**.
- 4 Navigate to **LDAP > LDAP Options > View LDAP Servers**.

---

**NOTE:** To access LDAP Options, you have to use the iManager from a different eDirectory server of the standalone version. Install the edirectory90 Plugins from the iManager Plug-in repository.

---

- 5 Select the Access Manager server, then click the **Connections** tab.
- 6 Under the **SSL Configuration** section select only **TLSv1.1** and **TLSv1.2**. The settings for other sections on the page do not require any change.
- 7 Save the configuration and restart the LDAP server from Administration Console using the following commands:

```
ndstrace -c "unload nldap"
```

```
ndstrace -c "load nldap"
```

- 8 Restart Tomcat by running the `/etc/init.d/novell-ac restart` command.

## 2.7 Running the DHost HTTP Server on localhost

The DHost HTTP server running on HTTP port 8028 and HTTPS port 8030 does not set the X-Frame-Options HTTP Response Header. Therefore, it is prone to clickjacking attacks. To prevent the vulnerabilities, it is recommended to restrict the DHost HTTP Server to localhost.

Perform the following steps to configure the DHost server to run on localhost:

- 1 In Administration Console, open `/etc/opt/novell/eDirectory/conf/nds.conf`.
- 2 Search for the following lines and then replace the IP address (for example, 10.0.0.1) with 127.0.0.1:

```
http.server.interfaces=10.0.0.1@8028
```

```
https.server.interfaces=10.0.0.1@8030
```

After the change these lines will look as follows:

```
http.server.interfaces=127.0.0.1@8028
```

```
https.server.interfaces=127.0.0.1@8030
```

### 3 Restart the eDirectory services:

```
/etc/init.d/nds restart
```

## 2.8 Preventing the SWEET32 Attack

In the SWEET32 attack, a remote attacker can obtain sensitive information by recovering portions of the plaintext data when encrypted with 64-bit block ciphers (such as Triple-DES).

To prevent this attack, you need to modify the cipher list in the `server.xml` files of Administration Console and Identity Server as follows:

```
ECDHE-ECDSA-AES128-GCM-SHA256: ECDHE-RSA-AES128-GCM-SHA256: ECDHE-ECDSA-AES256-GCM-SHA384: ECDHE-RSA-AES256-GCM-SHA384: ECDHE-ECDSA-CHACHA20-POLY1305: ECDHE-RSA-CHACHA20-POLY1305: DHE-RSA-AES128-GCM-SHA256: DHE-RSA-AES256-GCM-SHA384
```

For information about how to modify a configuration file, see “[Modifying Configurations](#)” in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

## 2.9 Default Security Settings in Configuration Files

- ♦ [Section 2.9.1, “server.xml,” on page 20](#)
- ♦ [Section 2.9.2, “web.xml,” on page 21](#)
- ♦ [Section 2.9.3, “tomcat8.conf,” on page 21](#)

### 2.9.1 server.xml

These settings are configured in `NIDP_Name="devman"` and `NIDP_Name="connector"` attributes inside the `Connector` element.

For the list of all default ciphers supported by Access Manager Administration Console, see [Section A.2, “Default Ciphers for Administration Console,” on page 83](#)

```
<Connector NIDP_Name="connector" SSLEnabled="true" URIEncoding="utf-8"
"acceptCount="100" address="10.0.0.0"
ciphers="TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256, TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256, TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384, TLS_RSA_WITH_AES_256_CBC_SHA256,
TLS_DHE_RSA_WITH_AES_256_CBC_SHA256, TLS_DHE_DSS_WITH_AES_256_CBC_SHA256,
TLS_RSA_WITH_AES_128_CBC_SHA256, TLS_DHE_RSA_WITH_AES_128_CBC_SHA256,
TLS_DHE_DSS_WITH_AES_128_CBC_SHA256" clientAuth="false"
disableUploadTimeout="true" enableLookups="false" keystoreFile="/opt/novell/
devman/jcc/certs/idp/connector.keystore" keystorePass="xxxxxxxxxxxxxxxx"
maxThreads="200" minSpareThreads="5" port="8443" scheme="https" secure="true"
sslImplementationName="com.novell.socket.DevManSSLImplementation"
useServerCipherSuitesOrder="true" sslProtocol="TLSv1.2"
sslEnabledProtocols="SSLv2Hello,TLSv1.1,TLSv1.2" />
```

For more information about connector attributes, see [Apache Tomcat Configuration Reference](#).

You can modify this file using Advanced File Configurator. See “[Advanced File Configurator](#)” in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

---

**NOTE:** When you install Access Manager 5.0 and the value of `sslProtocol` is TLS instead of TLSv1.2, change the value from TLS to TLSv1.2.

---

**IMPORTANT:** Access Manager supports and recommends TLSv1.2. You can reconfigure to use the earlier versions of TLS if required. However, due to cipher block chaining (CBC), using TLSv1.0 or earlier makes the environment vulnerable to the **BEAST** attack. This attack exploits the vulnerability at the client-side. TLS 1.1, TLS 1.2, and other cipher suites that do not use the CBC mode are not vulnerable to this attack.

---

## 2.9.2 web.xml

```
<filter>
    <filter-name>
        httpHeaderSecurity
    </filter-name>

    <filter-class>
        org.apache.catalina.filters.HttpHeaderSecurityFilter
    </filter-class>

    <async-supported>
        true
    </async-supported>

    <init-param>
        <param-name>hstsMaxAgeSeconds</param-name>
    <param-value>31536000</param-value>
    </init-param>

    <init-param>
        <param-name>antiClickJackingOption</param-name>
        <param-value>SAMEORIGIN</param-value>
    </init-param>
</filter>
<filter-mapping>
    <filter-name>httpHeaderSecurity</filter-name>
    <url-pattern>/*</url-pattern>
    <dispatcher>REQUEST</dispatcher>
</filter-mapping>
```

You can modify this file using Advanced File Configurator. See [“Advanced File Configurator”](#) in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

---

**NOTE:** You can add these filters at any location in `web.xml` as long as it is not within any existing tag.

---

## 2.9.3 tomcat8.conf

```
JAVA_OPTS=" ${JAVA_OPTS} -Dsun.security.ssl.allowUnsafeRenegotiation=false"
JAVA_OPTS=" ${JAVA_OPTS} -Djdk.tls.rejectClientInitiatedRenegotiation=true"
JAVA_OPTS=" ${JAVA_OPTS} -Djdk.tls.ephemeralDHKeySize=2048"
```

You can modify this file using Advanced File Configurator. See [“Advanced File Configurator”](#) in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.



# 3 Securing Identity Server

## In this Chapter

- ◆ Disabling Unused Authentication Protocols
- ◆ Configuring SSL Communication between Browsers and Identity Server
- ◆ Configuring SSL Communication with Identity Server and a Service Provider
- ◆ Securing Authentication by Using Strong and Multi-Factor Authentication Methods
- ◆ Securing Federation
- ◆ Configuring a Whitelist of Target URL
- ◆ Blocking Access to Identity Server Pages
- ◆ Enabling Advanced Session Assurance
- ◆ Securing Identity Server Web Service Interface
- ◆ Enabling reCAPTCHA
- ◆ Preventing the SWEET32 Attack
- ◆ Detecting the Brute Force Attack
- ◆ Restricting the Direct Access to Files in the nidp Folder
- ◆ Preventing Cross-Site Request Forgery Attacks
- ◆ Using the Device ID in the URN Cookie
- ◆ Configuring the Cookie Secure Flag
- ◆ Default Security Settings in Configuration Files

## 3.1 Disabling Unused Authentication Protocols

You must disable any authentication protocol that is not in use. Enabling additional protocols increases the attack surface area.

Go to **Identity Servers > [cluster name] > Configuration > General** and ensure to deselect unused protocols in **Enabled Protocols**.

## 3.2 Configuring SSL Communication between Browsers and Identity Server

See [Section 7.1.2, “Enabling SSL between Browsers and Identity Server,”](#) on page 54.

## 3.3 Configuring SSL Communication with Identity Server and a Service Provider

See [Section 7.1.3, “Enabling SSL between Identity Server and a Service Provider,”](#) on page 55.

## 3.4 Securing Authentication by Using Strong and Multi-Factor Authentication Methods

One of the strengths of Access Manager is its wide range of support for various means of authentication that goes beyond simple and commonly used username/password methods including multi-factor and step-up scenarios. Access Manager includes many built-in preconfigured schemes via the combination of classes, methods, and contracts that can be used as is or can be configured to meet your needs. You can assign a contract directly to specific protected resources or federation partners. For more sophisticated security needs, the contract can also be dynamically chosen by Access Manager risk policies. Risk policies can allow access, ask for step-up authentication, or deny access based on the risk calculated at the time of the access request.

For more information about the Access Managers risk-based authentication feature, see [“Risk-based Authentication”](#) in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

The authentication contract, assigned directly or determined by risk policies, can come from a variety of sources. Many are included with Access Manager itself. An example of the third-party provider is RADIUS. If you need advance security or you want to focus on both security and mobile users convenience, a variety of single and multi-factor contracts of the Advanced Authentication solution integrated with Access Manager is an ideal option.

For more information about configuring authentication methods, see [“Configuring Authentication”](#) in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

---

**NOTE:** You must not use persistent authentication or social authentication for applications that require high security. When using persistent authentication, associate the persistent cookie with the client IP address.

For securing cookies to prevent session replay attacks, enable Advanced session Assurance. For information, see [“Setting Up Session Assurance”](#) in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

---

### Authentication Contracts

If you have set up Access Manager to require SSL connections among all of its components, delete the Name/Password - Form and Name/Password - Basic contracts. Deleting the contracts removes them from the list of available contracts to be assigned to protected resources. If these contracts are assigned, the user’s password can be sent across the wire in the clear text format. You can re-create these when required. To delete these contracts, go to Administration Console and click **Identity Servers** > *[cluster name]* > **Authentication** > **Contracts**.

If you use password-based authentication, you can make it more secure by using second-factor authentication methods such as TOTP method or Advanced Authentication methods in the contract.

You can configure advanced authentication by using Advanced Authentication methods. For information, see [Multi-Factor Authentication Using Advanced Authentication](#).

## 3.5 Securing Federation

You can secure your federation relationships in numerous ways. The methods available are defined within federation protocols themselves. The method you want to use must be agreed upon by both members of a federation relationship. Specifically, this agreement is required between the identity provider (most often Access Manager's role) and the service provider (for example, a SaaS service).

The most commonly used means of security includes using HTTPS for communication between parties secured by well-known CA certificates. For information about how to enable HTTPS in Access Manager Identity Server, see [Enabling SSL between Identity Server and a Service Provider](#).

Another way for SAML is the signing and/or encryption of assertions. For more information, see [Configuring the Encryption Method for the SAML Assertion](#).

SAML also has options for communicating the assertion data between parties known as protocol bindings. Protocol bindings include Post and Artifact. The Post binding is currently simplest and most popular among SaaS vendors and is typically secured using HTTPS, assertion signing, and encryption. The Artifact binding is considered more secure, but its level of security is not always required for a federation relationship.

**Post method versus exchange artifacts:** When you set up a federation between an identity provider and a service provider, you can select either to exchange assertions with a post method or to exchange artifacts.

An assertion in a post method might contain the user's password or other sensitive data, which can make it less secure than an artifact when the assertion is sent to the browser.

An artifact is a randomly generated ID, it contains no sensitive data. Only the intended receiver can use it to retrieve assertion data.

If both providers support artifacts, you should select this method because it is more secure. For more details, see the **Response protocol binding** option in "[Configuring a SAML 2.0 Authentication Request](#)" and "[Configuring a SAML 2.0 Authentication Response](#)" in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

---

**NOTE:** To use exchange artifact, the service provider needs to establish a direct communication channel with the identity provider.

---

Additional SAML protocol options might also need to be configured and matched between the identity provider and service provider. Common options are covered later in this section.

### 3.5.1 Setting Options

Go to **Identity Servers > Servers > Edit > SAML 2.0 > Service Provider > Options** and set up the following options for a service provider:

- ♦ **SAML2 SIGN METHODDIGEST SHA256:** By default, this option is enabled. Assertions use the SHA 256 algorithm as a hashing algorithm for the service provider. You can disable this option by selecting false.

- ♦ **SAML2 POST SIGN RESPONSE TRUSTEDPROVIDERS:** Select true. The identity provider will sign the entire SAML 2.0 response for the service provider.
- ♦ **SAML2 AVOID AUDIENCE RESTRICTION:** Select true to avoid sending the audience restriction information with assertion to this service provider.
- ♦ **IS SAML2 POST SIGN RESPONSE:** Select true to enable the identity provider to send signed SAML 2.0 post responses to all its trusted providers.

---

**NOTE:** Configuring IS SAML2 POST SIGN RESPONSE is same as configuring the SignPost in web.xml. However, configuring it through Administration Console is recommended because it provides more options. You can combine these options with IS SAML2 POST SIGN RESPONSE to avoid Access Manager restarts.

---

## 3.5.2 Configuring the Encryption Method for the SAML Assertion

By default, AES128 (Advanced Standard Encryption, 128-bit) is used to encrypt SAML assertions. If you require a different encryption method, such as AES256 (Advanced Standard Encryption, 256-bit), you can modify the Tomcat web.xml file, and specify the required method.

- 1 Open the Identity Server [web.xml](#) file.
- 2 Add the following lines to the file:

```
<context-param>
    <param-name>EncryptionMethod</param-name>
    <param-value>AES256</param-value>
</context-param>
```

You can set the `<param-value>` element to AES128 or AES256. Because AES128 is the default, specifying this value in the web.xml file does not change any behavior.

For information about how to edit a file, see “[Modifying Configurations](#)” in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

The following algorithms for encryption method are supported:

```
<md:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc"/><md:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes256-cbc"/><md:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-oaep"/><md:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p"/>
```

## 3.6 Configuring a Whitelist of Target URL

URL redirection, which many applications and services require, inherently brings in security risks. While redirecting, the request can be tampered to redirect users to an external, malicious site. To prevent such issues, you can configure a list of permissible domains. Redirection is allowed only to the configured domains.

- ♦ [Section 3.6.1, “Configuring a Global Whitelist of Target URL,” on page 27](#)
- ♦ [Section 3.6.2, “Configuring a Whitelist of Intersite Transfer Service Target URL,” on page 27](#)
- ♦ [Section 3.6.3, “Configuring a Whitelist of Assertion Consumer Service URL,” on page 27](#)
- ♦ [Section 3.6.4, “Configuring a Whitelist of URLs for RelayState URL,” on page 28](#)



## 3.6.1 Configuring a Global Whitelist of Target URL

- 1 On the **Home** page, click **Identity Servers > Edit > Identity Providers**.
- 2 Under **Redirection White List**, click **New**.
- 3 Specify **Domain**.

You can specify a domain name with an asterisk wildcard character (\*) that represents the entire DNS subtree. For example, specifying \*.digitalairlines.com as a domain will allow redirection to all children domain under digitalairlines.com including digitalairlines.com. The www prefix is not required. You can specify the \* wildcard only at the lowest level of the subtree.

For example:

Valid domain name: \*.digitalairlines.com

Invalid domain name: innerweb.\*.com You must configure at least one domain to prevent open redirection.

**WS-Fed:** The wreply parameter is filtered. If the requested wreply is not in the white list, the Identity Server does not login. However, if wreply is same as the provider's single logout or single sign-on URL domain, the request is accepted.

**SAML 2.0:** For idpsend, the target parameter is filtered using this list. This list is not applicable for spsend.

## 3.6.2 Configuring a Whitelist of Intersite Transfer Service Target URL

- 1 On the **Home** page, click **Identity Servers > Edit > [SAML1.1, or SAML 2.0] > [Service Provider] > Intersite Transfer Service**.
- 2 In the **Domain List**, click **New**.
- 3 Specify the domain name.

The domain name must be a full domain name, such as www.digitalairlines.com.

Wildcard domain names, such as www.digitalairlines.\*.com, do not work.

## 3.6.3 Configuring a Whitelist of Assertion Consumer Service URL

When an authentication request from a service provider is not signed, Identity Server cannot validate the authenticity and integrity of the request. So, any intruder can intercept the request and change the Assertion Consumer Service URL in the request and make the Identity Server to send the assertion to malicious sites.

To secure and validate the authentication request from a service provider, you can use the following options in the service provider configuration of Identity Server:

- ♦ **SAML2\_ACS\_URL\_RESTRICT:** This option ensures that Identity Server must validate the Assertion Consumer Service URL in the request against the trusted metadata URL before sending the assertion. If the Assertion Consumer URL in the authentication request is tampered by any malicious user, Identity Server terminates the request and assertion is not sent.

- ♦ **SAML2\_ACS\_DOMAIN\_WHITELIST:** This option ensures that Identity Server must validate the Assertion Consumer URL in the request against a whitelist of domains. If the Assertion Consumer Service URL does not match with any of the domain URLs in the whitelist, Identity Server terminates the request.

You must define the `SAML2_ACS_DOMAIN_WHITELIST` along with `SAML2_ACS_URL_RESTRICT` for a service provider in Identity Server. `SAML2_ACS_DOMAIN_WHITELIST` does not work if `SAML2_ACS_URL_RESTRICT` is not enabled.

To define these options, perform the following steps:

- 1 On the **Home** page, click **Identity Servers** > <Cluster> > **Edit** > **SAML 2.0**.
- 2 Select the required service provider.
- 3 Click **Options** > **New**.
- 4 Select **OTHER** and specify the following properties:

Property Name	Property Value	Description
<code>SAML2_ACS_URL_RESTRICT</code>	True	Identity Server allows authentication only to the trusted ACS URLs.
<code>SAML2_ACS_DOMAIN_WHITELIST</code>	Domain names separated with semi-colon (;)	Identity Server performs additional validation of the authentication request with the ACS domain whitelist.

### 3.6.4 Configuring a Whitelist of URLs for RelayState URL

Enabling the **Use Introductions** option for an identity provider might result in the Open Redirect vulnerability. To prevent this vulnerability, perform the following steps:

- 1 Create the `relayStateWhiteList.conf` file and add the exact URLs for which you want to allow redirection.

For example, if you specify `https://www.<domainname>.com`, all sub-domains under this domain. Such as:

- ♦ `https://www.<domainname>.com/en-us/home`
- ♦ `https://www.<domainname>.com/en-us/digital-transformation/overview`
- ♦ `https://www.<domainname>.com`

If you specify `https://www.<domainname>.com/en-us/home`, then any child URL of this URL will be allowed. For example, `https://www.<domainname>.com/en-us/home/resources`. However, `https://www.<domainname>.com` will not be allowed.

- 2 Place `relayStateWhiteList.conf` to the `/opt/novell/nids/lib/webapp/WEB-INF` folder by using Advanced File Configurator.

For information about how to add a file, see “[Adding Configurations to a Cluster](#)” in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

- 3 Open the Identity Server `web.xml` file.

For information about how to edit a file, see “[Modifying Configurations](#)” in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

4 Uncomment the 'RelayStateWhiteListFilter' snippet:

```
<filter-name>RelayStateWhiteListFilter</filter-name>
<filter-class>com.mf.tomcat.custom.filter.RelayStateWhiteListFilter</
filter-class>
<Description>This filter is used to whitelist relaystate parameter for
introductions. </description>
<init-param>
    <param-name>configFile</param-name>
    <param-value>relayStateWhiteList.conf</param-value>
</init-param>
<init-param>
    <param-name>blackListRedirectPath</param-name>
    <param-value>/nidp/app/logout</param-value>
</init-param>
```

5 Save the file.

## 3.7 Blocking Access to Identity Server Pages

Authenticated users can access a few Identity Server pages. These pages contain information about the user and Identity Server that can cause security issues.

For information about how to block user access to these pages, see [“Blocking Access to the Legacy User Portal Page”](#) and [“Blocking Access to the WSDL Services Page”](#) in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

## 3.8 Enabling Advanced Session Assurance

Advanced Session Assurance enables you to prevent session replay attacks by adding an additional layer of security to your sessions. When a session is established, Access Manager creates a unique fingerprint of the device from which the session is established. During the session, at a configurable time interval, Access Manager validates the session to ensure that the fingerprint matches with that the device it originated from.

By default, in a fresh installation, Advanced Session Assurance is enabled for all clusters.

However, in an upgraded setup, it is disabled by default. You must upgrade all nodes in the cluster before enabling Advance Session Assurance.

For more information, see [“Setting Up Session Assurance”](#) in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

## 3.9 Securing Identity Server Web Service Interface

By default, the web service interface of Identity Server (`/nidp/services/IDSISCredentialProfile?wsdl`) is accessible by everyone. Identity Server and Access Gateway use this interface for updating credential profile information. An attacker can use this information to bring Identity Server down.

You can prevent such issues by configuring the `WSInterfaceFilter` filter in the `web.xml` file. You can modify filter's values depending on the requirement.

For information about how to edit a file, see [“Modifying Configurations”](#) in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

The following table lists parameters associated with the `WSInterfaceFilter` filter:

Parameter	Description
<code>activateWSFFirewall</code>	This activates the <code>WSFFirewall</code> filter. Specify <code>True</code> to activate the filter.
<code>shieldAllServices</code>	This specifies whether to shield all web services at <code>/nidp/services</code> or only selected services by using values <code>True</code> and <code>False</code> respectively.
<code>wsfAcceptedDevicesIPList</code>	This is a comma separated list of IP addresses that can access the <code>/nidp/services</code> interface. No white space is allowed.
<code>wsURLList</code>	This is a comma separated list of web services who can access to the web service when <code>shieldAllServices</code> is set to <code>False</code> . No whitespace is allowed.  For example, to filter requests for the <code>&lt;host&gt;/nidp/services/IDSISAuthenticationProfile</code> service, specify <code>IDSISAuthenticationProfile</code> as param-value for <code>wsURLList</code> . Both WSDL and the actual service will be placed behind the firewall.

**NOTE:** For certain web services, administrators can also specify a policy from Administration Console. If a policy is defined for a service that is in the `wsURLList` list, the policy is executed after passing this filter.

## 3.10 Enabling reCAPTCHA

Access Manager supports only the *invisible reCAPTCHA* version of reCAPTCHA. reCAPTCHA works on both Name/Password – Form and Secure Name/Password – Form authentication.

reCAPTCHA provides an additional layer of security by requesting users to confirm that they are not a robot. It displays images that users must select based on a matching criteria. If a response succeeds, Access Manager authenticates the user's authentication credentials. If a response fails, Access Manager does not authenticate the user credentials, and redirects to the login page.

**NOTE:** By default, reCAPTCHA is disabled. Before enabling this feature, verify your privacy statement. Some regions may consider it a privacy issue.

For more information, see [“Specifying Common Class Properties > Enabling reCAPTCHA”](#) in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

## 3.11 Preventing the SWEET32 Attack

See [Preventing the SWEET32 Attack](#).

## 3.12 Detecting the Brute Force Attack

When you enable the `RequestLogger` filter in the `web.xml` file, the debug information for each request, including the source IP address is recorded. You can check log messages and identify the IP address of the attacker and block these IP addresses.

This filter logs the client IP addresses of WS Trust, WS-Fed request. This filter is not required for logging client IP addresses of SAML requests.

Perform the following steps to enable the `RequestLogger` filter:

- 1 Modify Identity Server's [web.xml](#).
- 2 Uncomment the following snippet:

```
<filter>
  <filter-name>RequestLogger</filter-name>
  <filter-class>com.novell.nidp.servlets.filters.debug.RequestLogger</
filter-class>
</filter>
<filter-mapping>
  <filter-name>RequestLogger</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

For information about how to modify a configuration file, see “[Modifying Configurations](#)” in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

## 3.13 Restricting the Direct Access to Files in the nidp Folder

For security purposes, direct access to `application.xml` and `extern/dist/lib/` files available in the `nidp` folder is restricted by default. You can remove the restriction by commenting the `<security-constraint>` tag in the `web.xml` file.

If you want to restrict access to any other file in the `nidp` folder, perform the following steps:

- 1 Open the Identity Server [web.xml](#) file.  
For information about how to edit a file, see “[Modifying Configurations](#)” in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.
- 2 Under the `<security-constraint>` tag, add `<url-pattern>` or `<path of the file>` that you want to hide from the direct access.

The following is an example snippet:

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>Include files</web-resource-name>
    <description>No direct access to include files.</description>
    <url-pattern>/application.xml</url-pattern>
    <url-pattern>/extern/dist/lib/*</url-pattern>
    <http-method>POST</http-method>
    <http-method>GET</http-method>
  </web-resource-collection>
  <auth-constraint />
</security-constraint>
```

- 3 Save the file.

## 3.14 Preventing Cross-Site Request Forgery Attacks

The `CSRFDetectionFilter` filter verifies all requests to detect and mitigate any Cross-Site Request Forgery (CSRF) attempts. By default, this filter is disabled.

This filter verifies for a session-wide anti-CSRF token that is expected in each request as a form parameter or a query parameter. If the token is not found, the filter matches the referrer header in the request with the server host and any configured exclusions.

When CSRF is detected in the request, Identity Server throws the HTTP 400 Bad Request error. When logging is enabled, you can check the reasons in the log file (`catalina.out`).

Perform the following steps to enable this filter:

- 1 Open the Identity Server [web.xml](#) file.

For information about how to edit a file, see “[Modifying Configurations](#)” in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

- 2 Locate and uncomment the following snippet:

```
<!--<filter>
  <filter-name>CSRFDetectionFilter</filter-name>
  <filter-class>com.novell.nidp.servlets.filters.csrf.CSRFDetectionFilter</
filter-class>
  <description> This filter is used to detect CSRF attacks in NIDS, for an
authenticated session</description>
  <init-param>
    <param-name>active</param-name>
    <param-value>False</param-value>
  </init-param>
  <init-param>
    <param-name>exclude</param-name>
    <param-value>metadata</param-value>
  </init-param>
```

```

    <init-param>
      <param-name>RefererWhitelist</param-name>
      <param-value></param-value>
    </init-param>
    <init-param>
      <param-name>RequestWhitelist</param-name>
      <param-value>GET</param-value>
    </init-param>
  </filter>
  <filter-mapping>
    <filter-name>CSRFDetectionFilter</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>
-->

```

The following table lists parameters required for this filter:

Parameter	Description
active	You must set this to True to enable the filter. By default, this is set to False.
exclude	Specify the coma-separated paths you want to exclude in the verification by the CSRFDetectionFilter filter.  For example,  <pre> &lt;init-param&gt;   &lt;param-name&gt;exclude&lt;/param-name&gt;   &lt;param-value&gt;metadata,saml2&lt;/param-value&gt; &lt;/init-param&gt; </pre>
RefererWhitelist	Specify the coma-separated domain or hostname that you want to exclude in the verification by the CSRFDetectionFilter filter.  For example,  <pre> &lt;init-param&gt;   &lt;param-name&gt;RefererWhitelist &lt;/param-name&gt;   &lt;param-value&gt;www.test.com,www.idp.com&lt;/param-value&gt; &lt;/init-param&gt; </pre>
RequestWhitelist	Specify the coma-separated REST methods that you want to exclude in the verification by the CSRFDetectionFilter filter. By default, GET is included in the filter.  For example,  <pre> &lt;init-param&gt;   &lt;param-name&gt;RequestWhitelist &lt;/param-name&gt;   &lt;param-value&gt;GET,POST&lt;/param-value&gt; &lt;/init-param&gt; </pre>

3 Save the file.

### To Enable the CSRF Verification for the Password Class and TOTP Class

Access Manager provides an option **LOGIN CSRF CHECK** to verify CSRF attempts for the Password class and TOTP class. This is applicable for Access Manager default pages. If you have modified any page, you must add the anti-CSRF token to the page.

For information about how to enable this property, see “[Managing a Cluster of Identity Servers > Configuring Identity Server Global Properties](#)” in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

## 3.15 Using the Device ID in the URN Cookie

In an Access Manager environment with multiple Identity Servers and Access Gateways, a cluster cookie (UrnNovellNidpClusterMemberId) is automatically set for the serving node of the cluster. When requests come to Identity Server or Embedded Service Provider (ESP), this cookie is used by all nodes of the cluster to perform the proxying, if necessary.

For higher security, it is recommended to enable the **USE DEVICE ID IN URN COOKIE** property in Identity Server and Access Gateway. When this property is enabled, instead of obfuscation, hashing is used for the cookie value.

For information about how to enable this property, see [“Managing a Cluster of Identity Servers > Configuring Identity Server Global Properties”](#) and [“Managing Reverse Proxies and Authentication > Configuring ESP Global Options”](#) in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

## 3.16 Configuring the Cookie Secure Flag

Identity Server sets the cookie Secure flag by default. In some cases, the cookie Secure flag is not set because of which the cookie can be transmitted over an insecure connection. This leads to a risk where others can access the cookie information. This scenario occurs when Tomcat automatically sets the JSESSIONID cookie multiple times.

To mitigate the risk, you must configure the cookie Secure flag. Configuring the Secure flag ensures that the cookie is transmitted over a secure HTTPS connection.

- 1 Open the Identity Server [web.xml](#) file.

For information about how to edit a file, see [“Modifying Configurations”](#) in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

- 2 Add the following entry:

```
<session-config>
  <cookie-config>
    <secure>true</secure>
  </cookie-config>
</session-config>
```

## 3.17 Default Security Settings in Configuration Files

- ♦ [Section 3.17.1, “server.xml,”](#) on page 34
- ♦ [Section 3.17.2, “web.xml,”](#) on page 35
- ♦ [Section 3.17.3, “tomcat.conf,”](#) on page 35

### 3.17.1 server.xml

These settings are configured in NIDP\_Name="devman" and NIDP\_Name="connector" attributes inside the Connector element.

For the list of all default ciphers supported by Access Manager Identity Server, see [Section A.1, “Default Ciphers for Identity Server,”](#) on page 83



You can modify this file using Advanced File Configurator. See “[Advanced File Configurator](#)” in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

```
<Connector NIDP_Name="connector" SSLEnabled="true" URIEncoding="utf-8"
acceptCount="100" address="10.0.0.0"
ciphers="TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,TLS_RSA_WITH_AES_256_CBC_SHA256,
TLS_DHE_RSA_WITH_AES_256_CBC_SHA256,TLS_DHE_DSS_WITH_AES_256_CBC_SHA256,
TLS_RSA_WITH_AES_128_CBC_SHA256,TLS_DHE_RSA_WITH_AES_128_CBC_SHA256,
TLS_DHE_DSS_WITH_AES_128_CBC_SHA256" clientAuth="false"
disableUploadTimeout="true" enableLookups="false" keystoreFile="/opt/novell/
devman/jcc/certs/idp/connector.keystore" keystorePass="xxxxxxxxxxxxxxxx"
maxThreads="600" minSpareThreads="5" port="8443" scheme="https" secure="true"
sslImplementationName="com.example.nidp.common.util.net.server.NIDPSSLImplementati
on" useServerCipherSuitesOrder="true" sslProtocol="TLSv1.2"
sslEnabledProtocols="SSLv2Hello,TLSv1.1,TLSv1.2" />
```

For information about connector attributes, see [Apache Tomcat Configuration Reference](#).

## 3.17.2 web.xml

```
<filter>
  <filter-name>
    httpHeaderSecurity
  </filter-name>
  <filter-class>
    org.apache.catalina.filters.HttpHeaderSecurityFilter
  </filter-class>
  <async-supported>
    true
  </async-supported>
  <init-param>
    <param-name>hstsMaxAgeSeconds</param-name>
    <param-value>31536000</param-value>
  </init-param>
  <init-param>
    <param-name>antiClickJackingOption</param-name>
    <param-value>SAMEORIGIN</param-value>
  </init-param>
</filter>
<filter-mapping>
  <filter-name>httpHeaderSecurity</filter-name>
  <url-pattern>/*</url-pattern>
  <dispatcher>REQUEST</dispatcher>
</filter-mapping>
```

---

**NOTE:** You can add these filters at any location in the `web.xml` if it is not within any existing tag.

---

You can modify this file using Advanced File Configurator. See “[Advanced File Configurator](#)” in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

## 3.17.3 tomcat.conf

```
JAVA_OPTS="{JAVA_OPTS} -Dsun.security.ssl.allowUnsafeRenegotiation=false"
JAVA_OPTS="{JAVA_OPTS} -Djdk.tls.rejectClientInitiatedRenegotiation=true"
JAVA_OPTS="{JAVA_OPTS} -Djdk.tls.ephemeralDHKeySize=2048"
```

You can modify this file using Advanced File Configurator. See [“Advanced File Configurator”](#) in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

# 4 Securing Access Gateway

## In this Chapter

- ◆ [Enabling SSL Communication between Access Gateway and Identity Server](#)
- ◆ [Enabling Secure Cookies](#)
- ◆ [Disabling Phishing](#)
- ◆ [Disabling Weak Protocols between Access Gateway and Web Servers](#)
- ◆ [Configuring Stronger Ciphers for SSL Communication between Access Gateway and Web Servers](#)
- ◆ [Enabling Perfect Forward Secrecy](#)
- ◆ [Preventing Error Messages to Show the Failure Reason on Browsers](#)
- ◆ [Enabling Advanced Session Assurance](#)
- ◆ [Preventing the Slowloris Attack](#)
- ◆ [AJP Communication Setting for Access Gateway](#)
- ◆ [Using the Device ID in the URN Cookie](#)
- ◆ [Default Security Settings in Configuration Files](#)

## 4.1 Enabling SSL Communication between Access Gateway and Identity Server

See [Configuring SSL for Authentication between Identity Server and Access Gateway](#).

## 4.2 Enabling Secure Cookies

Access Gateway and Embedded Service Provider (ESP) of Access Gateway both use session cookies in their communication with the browser. You must protect these session cookies to prevent from being intercepted by hackers.

- ◆ [Securing the Embedded Service Provider Session Cookie](#)
- ◆ [Securing the Proxy Session Cookie](#)

---

**NOTE:** You can enable secure Access Gateway session cookies when only SSL resources exist. If a mix of HTTP and HTTPS proxy services exist, you cannot enable it as it is a global setting.

---

### 4.2.1 Securing the Embedded Service Provider Session Cookie

An attacker can spoof a non-secure browser into sending a JSESSION cookie that contains a valid user session. This might happen because Access Gateway communicates with its ESP on port 9009, which is a non-secure connection. Because ESP does not know whether Access Gateway is using SSL

to communicate with the browsers, ESP does not mark the JSESSION cookie as secure when it creates the cookie. Access Gateway receives the Set-Cookie header from ESP and passes it to the browser as a non-secure clear-text cookie. If an attacker spoofs the domain of Access Gateway, the browser sends the non-secure JSESSION cookie over a non-secure channel where the cookie might be sniffed.

To stop this, you must first configure Access Gateway to use SSL. See [Section 7.2.1, “Enabling SSL between Browsers and Access Gateway,”](#) on page 56.

After configuring SSL, you must perform the following steps to configure Tomcat to secure the cookie:

- 1 Open the Access Gateway `server.xml` file.
- 2 Search for the connector on port 9009.
- 3 Add the following parameter within the `Connector` element:

```
secure="true"
```

For information about how to edit a file, see “[Modifying Configurations](#)” in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

---

**NOTE:** This file is specific to each cluster. Therefore, while applying the changes from this file, the keystore password is retained in each cluster.

---

#### Preventing Automatically Changing Session ID

1. On the Home page, click [Access Gateways > Edit > Reverse Proxy / Authentication > ESP Global Options](#).
2. Set `RENAME_SESSIONID` to false. By default, this is set to true.
3. Restart Tomcat on each Identity Server in the cluster.

## 4.2.2 Securing the Proxy Session Cookie

Proxy session cookies store authentication information and other information in the temporary memory that is shared between the browser and the proxy. These cookies are deleted when the browser is closed. However if these cookies are sent through a non-secure channel, hackers might intercept the cookies and impersonate a user on websites. you can use the following configuration options:

- ♦ [Section 4.2.2.1, “Setting an Authentication Cookie with a Secure Keyword for HTTP,”](#) on page 38
- ♦ [Section 4.2.2.2, “Preventing Cross-Site Scripting Vulnerabilities,”](#) on page 39

### 4.2.2.1 Setting an Authentication Cookie with a Secure Keyword for HTTP

You can configure Access Gateway to force the HTTP services to authenticate the cookie set with the keyword secure.

To enable this option, perform the following steps:

- 1 On the Home page, click [Access Gateways > Edit > Reverse Proxy / Authentication](#).
- 2 Select [Enable Secure Cookies](#).

This option is used to secure a cookie when Access Gateway is placed behind an SSL accelerator, such as the Cisco SSL accelerator, and Access Gateway is configured to communicate by using only HTTP.

### 4.2.2.2 Preventing Cross-Site Scripting Vulnerabilities

Cross-site scripting vulnerabilities in web browsers allow malicious sites to grab cookies from a vulnerable site. Intruders might perform session fixation or impersonate the valid user. You can configure Access Gateway to set its authentication cookie with the `HttpOnly` keyword to prevent scripts from accessing the cookie.

To enable this option, perform the following steps:

- 1 On the Home page, click **Access Gateways > Edit > Reverse Proxy / Authentication**.
- 2 Select **Force HTTP-Only Cookies**.

## 4.3 Disabling Phishing

You can configure Access Gateway ESP to disable the ESP phishing by implementing a context parameter in the `web.xml` file for ESP.

- 1 Open the ESP `web.xml` file.
- 2 Add the following entry:

```
<context-param>
  <param-name>phishingCheck</param-name>
  <param-value>standard</param-value>
</context-param>
```

You can set the parameter value as `standard` or `off`.

For information about how to edit a file, see “[Modifying Configurations](#)” in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

## 4.4 Disabling Weak Protocols between Access Gateway and Web Servers

See the overview of [Strengthening TLS/SSL Settings](#) for information about weak protocols.

To restrict Access Gateway to communicate with backend web servers only using TLS 1.1 and TLS 1.2 protocols, set the following advanced options:

- ♦ On the Home page, click **Access Gateways > Edit > [Name of Reverse Proxy] > [Name of Proxy Service] > Advanced Options** and set `SSLProxyProtocol TLSv1.1 +TLSv1.2`

While setting the protocol, ensure that the web server supports the configured protocol. For example, if Access Manager supports TLS1.1, but the web server does not support that, the connection will fail.

- ♦ On the Home page, click **Access Gateways > Edit > Advanced Options**, and set `SSLProtocol -SSLV2 -SSLV3 -TLSv1 -TLSv1.1 +TLSv1.2`

For more information about SSLProxyProtocol directives, see [SSLProxyProtocol Directive documentation](#).

## 4.5 Configuring Stronger Ciphers for SSL Communication between Access Gateway and Web Servers

See the overview of [Strengthening TLS/SSL Settings](#) for information about strong ciphers.

1 On the **Home** page, click **Access Gateways > Edit > Advanced Options**.

2 Set the following advanced options:

- ◆ `SSLHonorCipherOrder` on
- ◆ `SSLCipherSuite` `ECDHE-RSA-AES256-SHA384:AES256-SHA256:HIGH:MEDIUM:!LOW:!EXP:!SSLv2:!aNULL:!EDH:!ECDH:!ECDSA:!AESGCM:!eNULL:!NULL`

While setting the cipher suite, ensure that the web server supports the cipher suite. For example, if Access Manager supports ECDH ciphers, but the web server does not support it, the connection fails.

You can configure `SSLCipherSuite` option as follows to get the **A+ rating** while validating with SSL Labs. However, this setting might affect performance.

```
SSLCipherSuite ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384
```

For the list of all default ciphers supported by Access Manager Access Gateway, see [Section A.4, "Default Ciphers for Access Gateway,"](#) on page 84

## 4.6 Enabling Perfect Forward Secrecy

Apache simplifies the process with the [SSLHonorCipherOrder directive](#). This directive indicates that Apache must respect the sequence of the encryption processes in `SSLCipherSuite` that is the first match found must be used. With the `SSLCipherSuite` list above and the `SSLHonorCipherOrder` on directive in place, Perfect Forward Secrecy (PFS) is enabled.

On the **Home** page, click **Access Gateways > Edit > Advanced Options** and set the following advanced options:

```
SSLHonorCipherOrder On
SSLCipherSuite
ECDH+AESGCM:ECDH+AES256:ECDH+AES128:ECDH+3DES:RSA+AESGCM:RSA+AES:!aNULL:!DES:!MD5:
!DSS
```

For information about PFS and prerequisites for enabling it, see [Enabling Perfect Forward Secrecy](#).

## 4.7 Preventing Error Messages to Show the Failure Reason on Browsers

Whenever Identity Server reports a 500 internal error due to an invalid input, the reason for failure is included in the response and visible on the browser.

This might cause a security issue as intruders can use this information to attack against Identity Server and ESP.

Configure the `web.xml` file for ESP as follows:

- 1 Open the `web.xml` file.
- 2 Update the file as follows:

```
<welcome-file-list>
  <welcome-file>index.html</welcome-file>
</welcome-file-list>
<error-page>
  <error-code>500</error-code>
  <location>/index.html</location>
</error-page>
```

`index.html` can be any custom page. Similarly, you can configure `web.xml` for error-code 404 by adding one more `<error-page>` tag. For information about how to edit a file, see “[Modifying Configurations](#)” in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

## 4.8 Enabling Advanced Session Assurance

Advanced Session Assurance enables you to prevent session replay attacks by adding an additional layer of security to your sessions. When a session is established, Access Manager creates a unique fingerprint of the device from which the session is established. During the session, at a configurable time interval, Access Manager validates the session to ensure that the fingerprint matches with that the device it originated from.

By default, in a fresh installation, Advanced Session Assurance is enabled for all clusters.

However, in an upgraded setup, it is disabled by default. You must upgrade all nodes in the cluster to version 4.3 or later before enabling Advance Session Assurance. Enable Advanced Session Assurance on the need basis. See “[Best Practices for Enabling Session Assurance at the Proxy Service Resource Level](#)” in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

For more information about Advanced Session Assurance and how to enable it, see “[Setting Up Session Assurance](#)” in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

## 4.9 Preventing the Slowloris Attack

To secure your environment from the Slowloris attack, you can configure the `RequestReadTimeout` option at the global and proxy service levels. This option sets timeout values for the following actions:

- ♦ Completing the TLS handshake
- ♦ Receiving the request headers
- ♦ Receiving the request body

Perform the following steps to configure the RequestReadTimeout option:

- 1 (Conditional) Modify the `httpd.conf` file using the Advanced File Configurator and add the following in the LoadModule section:

```
LoadModule reqtimeout_module libexec/mod_reqtimeout.so
```

For more information about how to manage configuration files using Advanced File Configurator, see [Managing Configuration Files](#) in the [NetIQ Access Manager CE 24.2 \(v5.1\) Administration Guide](#)

- 2 To configure the option at the global level, on the **Home** page, click **Access Gateways > Edit > Advanced Options**.
- 3 To configure the option for a proxy service, on the **Home** page, click **Access Gateways > Edit > [Name of Reverse Proxy] > [Name of Proxy Service] > Advanced Options**.
- 4 Add the option in the following format:

```
RequestReadTimeout [handshake=timeout[-maxtimeout]][,MinRate=rate]  
[header=timeout[-maxtimeout]][,MinRate=rate] [body=timeout[-  
maxtimeout]][,MinRate=rate]
```

For example, configure the option as follows to allow for 10 seconds to complete the TLS handshake, 15 seconds to receive the request headers, and 30 seconds for receiving the request body:

```
RequestReadTimeout handshake=10 header=15 body=30
```

- 5 Click **OK**.

For more information about this option, see [RequestReadTimeout Directive](#).

---

**NOTE:** This option is not supported for path-based multi-homing proxy services.

---

## 4.10 AJP Communication Setting for Access Gateway

By default, a constant passcode is used to communicate between HTTPD and Tomcat. For higher security, you can change this passcode. You must change the passcode in the `httpd` proxy service configuration file and the secret in `/opt/novell/nam/mag/conf/server1.xml`.

To change the passcode in the HTTPD configuration file, perform the following steps:

- 1 On the **Home** page, click **Access Gateways > Edit > Advanced Options**.
- 2 Set the `AJPToken <passcode>` option and specify the passcode. For example,

```
AJPToken <new-passcode>
```

The following list includes the supported special characters:

```
&  
[  
]  
|  
{  
}  
^  
\
```



```
`  
"  
<  
>
```

3 Click **OK**.

To change the secret in the server.xml file, perform the following steps:

- 1 In `/opt/novell/nam/mag/conf/server1.xml`, modify the value of secret in the AJP protocol section.

The value must be the same as the passcode specified in the `AJPToken <new-passcode>` option that you specified in [Step 2](#).

```
<Connector port="9009" enableLookups="false" redirectPort="8443"  
protocol="AJP/1.3" address="127.0.0.1" minSpareThreads="25" maxThreads="600"  
backlog="0" connectionTimeout="20000" packetSize="65536" maxPostSize="65536"  
secret="value" />
```

---

**NOTE:** Due to the tomcat restriction, special characters need to be specified in a specific format. For example, to include `&`, specify `&amp;`;

---

To specify a special character in the value, refer to the following list:

To Use	Specify
<code>&amp;</code>	<code>&amp;amp;</code>
<code>[</code>	<code>&amp;#x5B;</code>
<code>]</code>	<code>&amp;#x5D;</code>
<code> </code>	<code>&amp;#x7C;</code>
<code>{</code>	<code>&amp;#x7B;</code>
<code>}</code>	<code>&amp;#x7D;</code>
<code>^</code>	<code>&amp;#x5E;</code>
<code>\</code>	<code>&amp;#x5C;</code>
<code>`</code>	<code>&amp;#x60;</code>
<code>"</code>	<code>&amp;#x22;</code>
<code>&lt;</code>	<code>&amp;#x3C;</code>
<code>&gt;</code>	<code>&amp;#x3E;</code>

- 2 Save the file.
- 3 Restart Access Gateway by running the following command:  

```
/etc/init.d/novell-mag restart
```

## 4.11 Using the Device ID in the URN Cookie

See [Section 3.15, "Using the Device ID in the URN Cookie,"](#) on page 34.

## 4.12 Default Security Settings in Configuration Files

- ◆ Section 4.12.1, “ESP web.xml,” on page 44
- ◆ Section 4.12.2, “Access Gateway Advanced Options,” on page 44
- ◆ Section 4.12.3, “httpd.conf,” on page 44
- ◆ Section 4.12.4, “NovellAgSettings.conf,” on page 45

### 4.12.1 ESP web.xml

```
<context-param>
    <param-name>phishingCheck</param-name>
    <param-value>standard</param-value>
</context-param>

<welcome-file-list>
    <welcome-file>index.html</welcome-file>
</welcome-file-list>

<error-page>
    <error-code>500</error-code>
    <location>/index.html</location>
</error-page>

<filter>
    <filter-name>TomcatSameOriginFilter</filter-name>
    <filter-class>org.apache.catalina.filters.HttpHeaderSecurityFilter
</filter-class>

<init-param>
    <param-name>antiClickJackingOption</param-name>
    <param-value>SAMEORIGIN</param-value>
</init-param>
</filter>

<filter-mapping>
    <filter-name>TomcatSameOriginFilter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

You can modify this file using Advanced File Configurator. See “[Advanced File Configurator](#)” in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

### 4.12.2 Access Gateway Advanced Options

```
SSLProtocol TLSv1.1 +TLSv1.2
SSLCipherSuite !aNULL:!eNULL:!EXPORT:!DSS:!DES:!RC4:ALL:!EDH
```

### 4.12.3 httpd.conf

The mod\_headers library is enabled.

```
LoadModule headers_module libexec/mod_headers.so
```

You can modify this file using Advanced File Configurator. See “[Advanced File Configurator](#)” in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

## 4.12.4 NovellAgSettings.conf

The header set directive for the HSTS header is added at the bottom of the file:

```
Header always set Strict-Transport-Security "max-age=31536000; includeSubDomains"
```

You can modify this file using Advanced File Configurator. See “[Advanced File Configurator](#)” in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.



# 5 Securing Analytics Server

- ◆ Section 5.1, “Customizing the Size of EDH Keys,” on page 47
- ◆ Section 5.2, “Configuring SSL in Analytics Server,” on page 47
- ◆ Section 5.3, “Disabling SSL Renegotiations,” on page 47
- ◆ Section 5.4, “Securing Analytics Server Cluster Communication,” on page 48
- ◆ Section 5.5, “Setting Analytics Dashboard Timeout,” on page 48
- ◆ Section 5.6, “Default Security Settings in Configuration Files,” on page 48

## 5.1 Customizing the Size of EDH Keys

For information about why to customize the EDH key size, see [Customizing the Size of Ephemeral Diffie-Hellman Keys](#).

- 1 Open the `/opt/novell/nam/dashboard/conf/tomcat.conf` file.
- 2 Ensure that the following line exists:

```
JAVA_OPTS="{JAVA_OPTS} -Djdk.tls.ephemeralDHKeySize=2048"
```

## 5.2 Configuring SSL in Analytics Server

See [Section 7.4, “Configuring SSL in Analytics Server,” on page 58](#).

## 5.3 Disabling SSL Renegotiations

Disable SSL renegotiation as it is vulnerable to the man-in-the-middle attacks.

Perform the following steps to disable SSL renegotiations in Analytics Server:

- 1 Open the `/opt/novell/nam/dashboard/conf/tomcat.conf` file.
- 2 Ensure that the following lines exist:

```
JAVA_OPTS="{JAVA_OPTS} -  
Dsun.security.ssl.allowUnsafeRenegotiation=false "  
JAVA_OPTS="{JAVA_OPTS} -  
Djdk.tls.rejectClientInitiatedRenegotiation=true "
```

## 5.4 Securing Analytics Server Cluster Communication

To access Analytics Dashboard, ensure that port 8445 is open. For information about other ports that you need to open for internal communication, see [“Prerequisites for Installing Analytics Server”](#) in the *NetIQ Access Manager CE 24.2 (v5.1) Installation and Upgrade Guide*.

You can use IP tables to restrict cluster communication. Allow communication between only Analytics Server cluster nodes and Access Manager devices.

The following is a sample configuration of IP tables:

```
Iptables -P INPUT DROP    ## By default drop all
iptables -A INPUT -s 164.99.184.0/23 -j ACCEPT ## You can allow traffic only
between Analytics Dashboard cluster nodes and Access Manager devices instead of the
entire network.
iptables -A INPUT -i lo -j ACCEPT    ## Enable Loop back communication
iptables -A INPUT -p tcp --dport 8445 -j ACCEPT    ## Enable 8445 for public access
iptables-save
```

## 5.5 Setting Analytics Dashboard Timeout

By default, no timeout is set for the Analytics Dashboard session. However, for security reasons, it is recommended to set an appropriate timeout value.

Perform the following steps to set a timeout value:

- 1 In the `/opt/netiq/common/tomcat/conf/web.xml` file, add the following snippet within the `<session-config>` element:

```
<cookie-config>
  <max-age>3600</max-age>  <!-- 3600 is in seconds -->
</cookie-config>
```

- 2 Restart Analytics Dashboard.

## 5.6 Default Security Settings in Configuration Files

- ♦ [Section 5.6.1, “server.xml,”](#) on page 48
- ♦ [Section 5.6.2, “web.xml,”](#) on page 49

### 5.6.1 server.xml

Path: `/opt/novell/nam/dashboard/conf/server.xml`

For the list of all default ciphers supported by Access Manager Analytics Server, see [Section A.3, “Default Ciphers for Analytics Server,”](#) on page 84

```

<Connector NIDP_Name="connector" SSLEnabled="true" URIEncoding="utf-8"
acceptCount="100" ciphers="TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA
256,
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA
384,
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,TLS_RSA_WITH_AES_256_CBC_SHA256,TLS_
DHE_RSA_WITH_AES_256_CBC_SHA256,TLS_DHE_DSS_WITH_AES_256_CBC_SHA256,TLS_RS
A_WITH_AES_128_CBC_SHA256,TLS_DHE_RSA_WITH_AES_128_CBC_SHA256,
TLS_DHE_DSS_WITH_AES_128_CBC_SHA256" clientAuth="false"
disableUploadTimeout="true" enableLookups="false" keystoreFile="/opt/
novell/devman/jcc/certs/ra/connector.keystore"
keystorePass="xxxxxxxxxxxxxxxx" maxThreads="150" minSpareThreads="5"
port="8445" protocol="org.apache.coyote.http11.Http11NioProtocol"
scheme="https" secure="true"
sslEnabledProtocols="SSLv2Hello,TLSv1.1,TLSv1.2" sslProtocol="TLSv1.2"/>

```

## 5.6.2 web.xml

Path: /opt/novell/nam/dashboard/webapps/kibana/WEB-INF/web.xml

```

<filter>
  <filter-name>httpHeaderSecurity</filter-name>
  <filter-class>org.apache.catalina.filters.HttpHeaderSecurityFilter
</filter-class>
  <async-supported>true</async-supported>
</filter>

<init-param>
  <param-name>hstsMaxAgeSeconds</param-name>
  <param-value>31536000</param-value>
</init-param>

<filter-mapping>
  <filter-name>httpHeaderSecurity</filter-name>
  <url-pattern>*</url-pattern>
  <dispatcher>REQUEST</dispatcher>
</filter-mapping>

<filter>
  <filter-name>TomcatSameOriginFilter</filter-name>
  <filter-class>org.apache.catalina.filters.HttpHeaderSecurityFilter
</filter-class>
  <init-param>
    <param-name>antiClickJackingOption</param-name>
    <param-value>SAMEORIGIN</param-value>
  </init-param>
</filter>

<filter-mapping>
  <filter-name>TomcatSameOriginFilter</filter-name>
  <url-pattern>*</url-pattern>
</filter-mapping>

```





# 6 Hardening Appliance

## In this Chapter

- ♦ [Section 6.1, “Disabling Weak MAC Algorithms on a Secure Shell Server,”](#) on page 51

## 6.1 Disabling Weak MAC Algorithms on a Secure Shell Server

After installing or upgrading Analytics Server, reconfigure SSH server to use the strong MAC algorithms.

You can identify the available MAC algorithms by using the `sudo sshd -T |grep mac` command. The MD5 or 96-bit MAC algorithms are considered as weak algorithms. Hence, you must remove the weak algorithms.

To remove the weak MAC algorithms, perform the following:

- 1 Log in to Analytics Server with `root` credentials.
- 2 Open the `/etc/ssh/sshd_config` file and search for `macs`.
- 3 Remove the weak MAC algorithms that are mentioned in the file.

The entry will be similar to the following line and can include additional strong MAC algorithms:

```
macs hmac-sha1,hmac-sha2-256,hmac-sha2-512,hmac-ripemd160,hmac-ripemd160@openssh.com,umac-64@openssh.com,umac-128@openssh.com,hmac-sha1-etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-ripemd160-etm@openssh.com,umac-64-etm@openssh.com,umac-128-etm@openssh.com
```

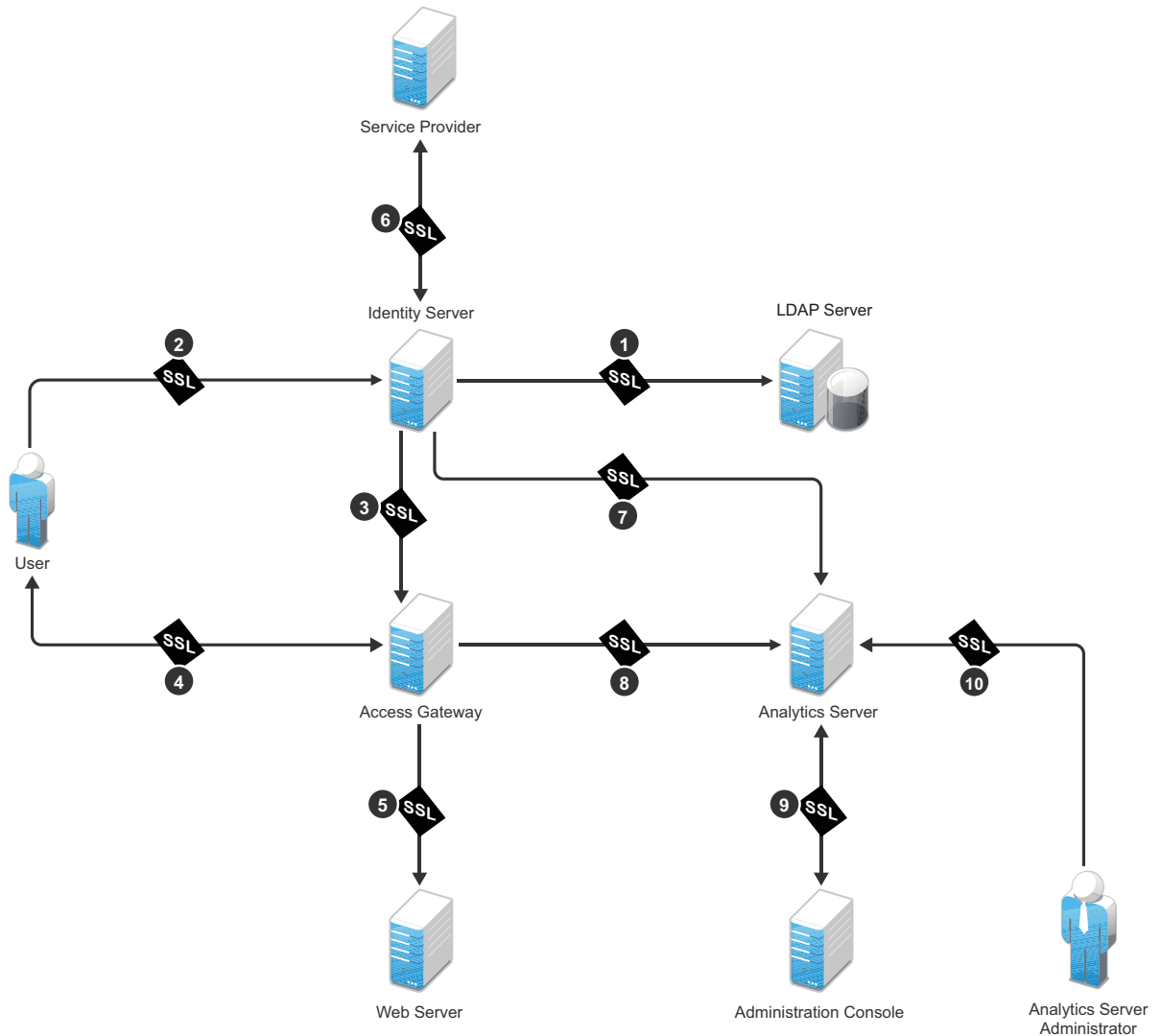
- 4 Save the file.
- 5 Restart the `sshd` service by using the `service sshd restart` command.
- 6 Launch a new SSH session before closing the existing session.  
This verifies the connection and you can log in to the server with the `root` account.
- 7 (Conditional) If the connection to server fails, revert the changes to the `sshd_config` file.



# 7 Configuring Secure Communication

Access Manager has six communication channels that you can configure for SSL. The following diagram illustrates potential SSL Communication channels:

*Figure 7-1 SSL Communication Channels*



The first channel is set between Identity Server and LDAP servers when you configure user stores. The other channels are configured according to their numeric values. SSL must be configured between Identity Server and browsers before you configure the channel between Access Gateway and Identity Server for SSL.

This section discusses the following topics:

- ◆ [Configuring SSL in Identity Server](#)
- ◆ [Configuring SSL in Access Gateway](#)
- ◆ [Configuring SSL for Authentication between Identity Server and Access Gateway](#)
- ◆ [Configuring SSL in Analytics Server](#)
- ◆ [Using Trusted Certificates Authority](#)

## 7.1 Configuring SSL in Identity Server

An attacker can spoof a non-secure browser and send a JSESSION cookie that contains a valid user session. You can prevent this by configuring Identity Server to use a SSL channel for communications.

### In this Section

- ◆ [Configuring a SSL Channel between Identity Server and LDAP Servers](#)
- ◆ [Enabling SSL between Browsers and Identity Server](#)
- ◆ [Enabling SSL between Identity Server and a Service Provider](#)

### 7.1.1 Configuring a SSL Channel between Identity Server and LDAP Servers

Channel 1 in [Figure 7-1, “SSL Communication Channels,” on page 53](#).

You can set a SSL channel between Identity Server and LDAP servers while configuring user stores. Select the **Use secure LDAP connections** option to change the port from 389 to the secure LDAP port 636.

---

**IMPORTANT:** If you use port 389, user names and passwords are sent in the clear text that is vulnerable to security issues.

---

To enable the **Use Secure LDAP Connections** option, perform the following steps:

- 1 Go to **Identity Servers** > *[cluster name]* > **User Stores**.
- 2 Click *[name of the user store]* > *[name of the replica]*.
- 3 Turn on **Use Secure LDAP Connections**.

### 7.1.2 Enabling SSL between Browsers and Identity Server

Channel 2 in [Figure 7-1, “SSL Communication Channels,” on page 53](#).

- 1 On the **Home** page, click **Identity Servers** > *[cluster name]* > **General**.
- 2 Change the protocol to HTTPS. The system changes the port to 8443.
- 3 In the **SSL Certificate** line, click **Browse** and select the Identity Server certificate.
- 4 Restart Tomcat.

If your Identity Server and Administration Console are on the same machine, log in to Administration Console again.

5 After the Identity Server health turns green, on the **Home** page, click **Access Gateways > Edit > Service Provider Certificates > Trusted Roots**.

6 Click **Add** to select the trusted root certificate of the certificate authority that signed Identity Server certificate.

(Conditional) If you imported intermediate certificates for the CA, select them also.

---

**IMPORTANT:** If the external certificate authority writes the DN in reverse order (the cn element is displayed first), you receive an error message that the certificate names do not match. You can ignore this warning, if the order of the DN elements is the cause.

---

7 Update Access Gateway.

## 7.1.3 Enabling SSL between Identity Server and a Service Provider

Channel 6 in [Figure 7-1, “SSL Communication Channels,”](#) on page 53.

To make the communication between Identity Server and a service provider more secure, you must consider the following settings:

**Identity Provider Signing Certificate:** Select a certificate from the keystore and assign it to the service provider.

**Identity Provider Encryption Certificate:** Select a certificate from the keystore and assign it to the service provider.

**Signing certificate per service provider:** When you assign custom certificates to each service provider while configuring Identity Server, ensure that you export these certificates and custom metadata to the service provider. To retrieve the metadata, click the metadata link (available in the note on the Trust page).

For more information, see [“Configuring Communication Security for a SAML 2.0 Service Provider”](#) *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

---

**NOTE:** These security considerations are also valid when Identity Server acts as a service provider.

---

## 7.2 Configuring SSL in Access Gateway

You can configure Access Gateway to use SSL in its connections to Embedded Service Provider (ESP), browsers, and its web servers.

**Enable SSL with ESP:** To encrypt the data exchanged for authentication (a communication channel between Identity Server and Access Gateway). This option is available only for the reverse proxy that has been assigned to perform authentication.

If you enable SSL between browsers and Access Gateway, this option is automatically selected. You can enable SSL with the ESP without enabling SSL between Access Gateway and browsers. This allows the authentication and identity information that Access Gateway and Identity Server

exchange to use a secure channel. However, it allows the data, that Access Gateways retrieves from the back-end web servers and sends to users, to use a non-secure channel. This saves processing overhead if the data on web servers is not sensitive.

**Enable SSL between Browser and Access Gateway:** To configure SSL connections between your clients and Access Gateway. SSL must be configured between browsers and Access Gateway before you can configure SSL between Access Gateway and web servers.

**Redirect Requests from Non-Secure Port to Secure Port:** To determine whether browsers are redirected to a secure port and allowed to establish an SSL connection. If this option is not selected, browsers that connect to the non-secure port are denied service.

This option is only available if you have selected **Enable SSL with Embedded Service Provider**.

For information about how to enable SSL between SSL with ESP and how to redirect requests from a non-secure port to a secure port, see [Enabling SSL between Browsers and Access Gateway](#).

## 7.2.1 Enabling SSL between Browsers and Access Gateway

This section explains how to enable SSL communication between Access Gateway and browsers (channel 4 in [Figure 7-1 on page 53](#)).

- 1 On the **Home** page, click **Access Gateways > Edit > [Name of Reverse Proxy]**.
- 2 Select the following options based on your requirement:
  - ◆ **Enable SSL with Embedded Service Provider**
  - ◆ **Enable SSL between Browser and Access Gateway**
  - ◆ **Redirect Requests from Non-Secure Port to Secure Port**
- 3 Select the certificate to use for SSL between Access Gateway and browsers.
- 4 (Conditional) If you selected a certificate in [Step 3](#) that was created by an external CA, click **Auto-Import Embedded Service Provider Trusted Root**, and specify an alias name.

This option imports the public key from ESP into the trust store of Identity Servers of the selected Identity Server configuration. This sets up a trusted SSL relationship between Identity Server and ESP.

If you are using certificates signed by the Access Manager CA, the public key is automatically added to this trust store.

- 5 Configure the ports for SSL:

**Non-Secure Port:** Indicates a specific port to listen to HTTP requests. The default port for HTTP is 80.

- ◆ If you selected the **Redirect Requests from Non-Secure Port to Secure Port** option, requests sent to this port are redirected to the secure port. If the browser can establish an SSL connection, the session continues on the secure port. If the browser cannot establish an SSL connection, the session is terminated.
- ◆ If you do not select the **Redirect Requests from Non-Secure Port to Secure Port** option, this port is not used when SSL is enabled.

**Secure Port:** Indicates a specific port to listen to HTTPS requests (usually 443). This port needs to match the configuration for SSL. If SSL is enabled, this port is used for all communication with the browsers. The listening address and port combination must not match any combination you have configured for another reverse proxy or tunnel.

- 6 Click **OK** > **Reverse Proxy / Authentication**.
- 7 (Conditional) If you are using an externally signed certificate for Identity Server cluster, click **Auto-Import Identity Server Trusted Root** to import the public key of the CA.

## 7.2.2 Enabling SSL between Access Gateway and Web Servers

Channel 5 in [Figure 7-1, “SSL Communication Channels,”](#) on page 53.

SSL must be enabled between Access Gateway and browsers before you can enable it between Access Gateway and its web servers. See [Enabling SSL between Browsers and Access Gateway](#).

- 1 On the **Home** page, click **Access Gateways > Edit [Name of Reverse Proxy] > [Name of Proxy Service] > Web Servers**.
- 2 Select **Connect Using SSL**.
- 3 Configure how you want the proxy service to verify the web server certificate:
  - 3a Select one of the following options in **Web Server Trusted Root**:
    - Do not verify**: Use this option when you want the information between Access Gateway and the web server encrypted, but you do not need the added security of verifying the web server certificate.

Continue with [Step 4](#).

**Any in Reverse Proxy Trust Store**: Use this option to verify the certificate authority of the web server certificate. When this option is selected, the public certificate of the certificate authority must be added to the proxy trust store.

---

**IMPORTANT:** For an Access Gateway Service, this is a global option. If you select this option for one proxy service, all proxy services on an Access Gateway Service are flagged to verify the public certificate. This verification is done even when other proxy services are set to **Do not verify**.

If the web server certificate is part of a chain of certificates, select **SSLProxyVerifyDepth** and specify how many certificates are in the chain.

The SSL connection between Access Gateway and a web server may fail if a self-signed certificate is used. To prevent this, import the web server certificates to the proxy trust store and then use the following advanced option:

```
SSLProxyCACertificateFile /opt/novell/apache2/cacerts/myserver.pem.
```

This is a service level advanced option.

---

- 3b Click **Manage Reverse Proxy Trust Store**.
- 3c Ensure that the IP address of the web server and the port match your web server configuration and then click **OK**.
  - If the whole chain is not displayed, import what is displayed. You then need to manually import the missing parents in the chain. A parent is missing if the chain does not include a certificate where the Subject and the Issuer have the same CN.
- 3d Specify an alias.
  - All the displayed certificates are added to the trust store.
- 4 (Optional) Set up mutual authentication so that the web server can verify the proxy service certificate. Click **Select Certificate** to select the certificate you created for the reverse proxy.

Import the trusted root certificate of the CA that signed the proxy service's certificate to the web servers assigned to this proxy service. For instructions, see the web server documentation.

- 5 In **Connect Port**, specify the port that your web server uses for SSL communication.

## 7.3 Configuring SSL for Authentication between Identity Server and Access Gateway

This section explains how to enable SSL communication between Access Gateway and Identity Server (channel 3 [Figure 7-1, "SSL Communication Channels,"](#) on page 53).

- 1 On the **Home** page, click **Access Gateways > Edit > [Name of Reverse Proxy]**.
- 2 Select **Enable SSL with Embedded Service Provider** and **Enable SSL between Browser and Access Gateway**.
- 3 In the **Server Certificate** line, click the **Browse** icon to select the Access Gateway certificate.

---

**IMPORTANT:** If the external certificate authority writes the DN in reverse order (the cn element comes first rather than last), you receive an error message that the subject name does not contain the cn of the device. You can ignore this warning, if the order of the DN elements is the cause.

---

- 4 Click **Auto-Import Embedded Service Provider Trusted Root**.  
This adds the trusted root of the Access Gateway certificate to the trusted root store of Identity Server.
- 5 Specify an **Alias** for the certificate.
- 6 On the Server Configuration page, click **Reverse Proxy / Authentication**.
- 7 In the **Embedded Service Provider** section, click **Auto-Import Identity Server Configuration Trusted Root** and follow the prompts.  
This imports the trusted root certificate of Identity Server into the trusted root store of the embedded service provider.
- 8 Update Access Gateway and Identity Server on respective pages.

## 7.4 Configuring SSL in Analytics Server

Channel 7, 8, and 9 in [Figure 7-1, "SSL Communication Channels,"](#) on page 53.

- 1 Generate the Certificate Authority (CA) Certificate.

- 1a Create a private key.

```
certtool --generate-privkey --outfile ca-key.pem
```

- 1b Create the self-signed certificate.

```
certtool --generate-self-signed --load-privkey ca-key.pem --outfile ca.pem
```

- 2 Generate a certificate for the local syslog client (private key).

- 2a Create a private key for the syslog agent.

```
certtool --generate-privkey --outfile rslclient-key.pem --bits 2048
```



**2b** Generate a certificate request for the syslog client.

```
certtool --generate-request --load-privkey rslclient-key.pem --
outfile request.pem
```

**2c** Generate a certificate and sign it with the CA private key.

```
certtool --generate-certificate --load-request request.pem --
outfile rslclient-cert.pem --load-ca-certificate ca.pem --load-ca-
privkey ca-key.pem
```

**3** Generate a certificate for the remote syslog server (private key)

**3a** Remove the previously generated request.pem.

**3b** Create a private key for the syslog server.

```
certtool --generate-privkey --outfile rslserver-key.pem --bits 2048
```

**3c** Generate a certificate request for the rsyslog server.

```
certtool --generate-request --load-privkey rslserver-key.pem --
outfile request.pem
```

**3d** Generate a machine certificate and sign it with the CA private key.

**3e** Copy certificates from CA to rsyslog server and to rsyslog client.

```
certtool --generate-certificate --load-request request.pem --
outfile rslserver-cert.pem --load-ca-certificate ca.pem --load-ca-
privkey ca-key.pem
```

**3f** Configure the syslog server for TLS.

Sample nam.conf:

```
# Increase the amount of open files rsyslog is allowed, which
includes open tcp sockets
# This is important if there are many clients.
# http://www.rsyslog.com/doc/rsconf1_maxopenfiles.html
$MaxOpenFiles 2048

# make gtls driver the default $DefaultNetstreamDriver gtls

# certificate files generated on RHEL6 and stored in /root
$DefaultNetstreamDriverCAFile /etc/pki/rsyslog/ca.pem
$DefaultNetstreamDriverCertFile /etc/pki/rsyslog/rslserver-cert.pem
$DefaultNetstreamDriverKeyFile /etc/pki/rsyslog/rslserver-key.pem
```

**3g** Configure the syslog agent for TLS.

Sample /etc/rsyslog.d/nam.conf:

```
# make gtls driver the default $DefaultNetstreamDriver gtls

# certificate files $DefaultNetstreamDriverCAFile /etc/pki/rsyslog/
ca.pem $DefaultNetstreamDriverCertFile /etc/pki/rsyslog/rslclient-
cert.pem $DefaultNetstreamDriverKeyFile /etc/pki/rsyslog/rslclient-
key.pem

#### GLOBAL DIRECTIVES #### $ActionSendStreamDriverAuthMode x509/
name $ActionSendStreamDriverMode 1 # run driver in TLS-only mode
```

---

**NOTE:** Making any changes on the Auditing UI overwrites the manual changes made in the `nam.conf` file. The changes must be manually done in each component.

---

## 7.5 Using Trusted Certificates Authority

When Identity Server is configured to use an SSL certificate that is signed externally, the trusted store of the embedded service provider for each component must be configured to trust this new CA. Browsers that are used to authenticate to Identity Server must be configured to trust the CA that created the certificate for Identity Server. Most browsers are already configured to trust certificates from well-known CAs.

To use certificates signed by an external CA, perform the following activities:

1. Obtain externally signed certificates.

For more information, see the *“Obtaining Externally Signed Certificates”* section in [“Using Externally Signed Certificates”](#).

2. Configure Identity Server to use externally signed certificates.

For more information, see the *“Configuring Identity Server to Use an Externally Signed Certificate”* section in [“Using Externally Signed Certificates”](#).

3. Configure Access Gateway to use externally signed certificates.

For more information, see the *“Configuring Access Gateway to Use an Externally Signed Certificate”* section in [“Using Externally Signed Certificates”](#).

# 8

## Strengthening TLS/SSL Settings

Securing TLS/SSL settings have the following three aspects:

- ♦ **Protocol:** SSL v2, SSL v3, and TLS1.0 contain known vulnerabilities. Starting with JDK 8u31, SSL v3 has been deactivated and is not available by default. If SSLv3 is required, you can reactivate the protocol at the JRE level. For more information, see [The SunJSSE Provider](#).  
By default, Access Manager is configured with only TLS1.1 and TLS1.2.
- ♦ **Encryption:** In the encryption algorithms, you need to look at two aspects:
  - ♦ **Key Exchange Algorithm:** In these algorithms, DH is vulnerable. By default, Access Manager 4.3 and later include only RSA, DHE, or ECDHE.
  - ♦ **Bulk Encryption Algorithm:** In these algorithms, cipher suites that contain NULL, DES, 3DES, and RC4 encryptions are vulnerable. By default, Access Manager supports cypher suites only with AES.
- ♦ **Message Authentication Code (MAC) Algorithm:** In these algorithms, MD5 and SHA1 are vulnerable. By default, Access Manager supports cypher suites only with SHA 256 or higher.

In Access Manager 4.3 and later, security is strengthened. These security measures can impact performance. For example, DHE and ECDHE ciphers are more secure, but they need more computation and therefore impacts performance. Between DHE and ECDHE, ECDHE reduces some computational cost comparatively and in turn it is better than DHE ciphers in terms of performance. You can configure the cipher optimally based on your security and performance requirements by referring to [The Sun JSSE Provider](#).

If you want to restore the previous security settings, see [Chapter 13, “Restoring Previous Security Level After Upgrading Access Manager,”](#) on page 75.

### In this Chapter

- ♦ [Disabling SSLv2 and SSLv3 Protocols](#)
- ♦ [Optimizing SSL Configuration with Ciphers](#)
- ♦ [Enabling Perfect Forward Secrecy](#)
- ♦ [Adding HTTP Strict Transport Security](#)
- ♦ [Disabling SSL Renegotiations](#)
- ♦ [Customizing the Size of Ephemeral Diffie-Hellman Keys](#)

## 8.1 Disabling SSLv2 and SSLv3 Protocols

In Access Manager 4.3 and later versions, SSL v2 and SSL v3 protocols are disabled by default for Administration Console, Identity Server, and between browsers and Access Gateway.

To disable SSL v2 and SSL v3 between Access Gateway and web servers, see [Section 4.4, “Disabling Weak Protocols between Access Gateway and Web Servers,”](#) on page 39.

## 8.2 Optimizing SSL Configuration with Ciphers

In addition to setting up Transport Level Security (TLS), using a cipher suite provides additional security to client-server communications from Identity Server, Access Gateway to the web browsers.

---

**IMPORTANT:** The settings specified in this section indicate an SSL configuration that provides an optimal level of security. If you plan to make any changes in the cipher information, ensure that you test the configuration before you deploy it in the production setup.

---

In Access Manager 4.3 and later, stronger ciphers for SSL communications have been configured for Administration Console, Identity Server, and communication between browsers and Access Gateway.

For information about how to specify SSL Configuration for communication between Access Gateway and web servers, see [Section 4.5, “Configuring Stronger Ciphers for SSL Communication between Access Gateway and Web Servers,”](#) on page 40

## 8.3 Enabling Perfect Forward Secrecy

When an SSL handshake is performed, SSL information regarding the capabilities of browser/client and server is exchanged and validated. An SSL session key that meets both the client’s and server’s criteria is established. After the session key is established, all subsequent communication between the client and the site is encrypted and thus secured.

The most common method for negotiating the session key is the RSA public-key cryptosystem. The RSA approach uses the server’s public key to protect the session key parameters created by the client after the key parameters are sent to the server. The server decrypts this handshake with its corresponding private key. If an attacker ever steals the server’s private key, they can decrypt your SSL session and any saved SSL sessions. This approach allows Wireshark or ssldump tools to decrypt the saved SSL communication by using an exported server certificate with private key.

Perfect Forward Secrecy (PFS) removes this shortcoming of the RSA approach. When PFS is enabled, no link between the server’s private key and each session key is established. If an attacker ever gets access to your server’s private key, the attacker cannot use the private key to decrypt any of your archived sessions.

In Access Manager 4.3 and later, PFS are enabled by default for Administration Console and Identity Server. For information about how to enable PFS in Access Gateway, see [Section 4.6, “Enabling Perfect Forward Secrecy,”](#) on page 40.

## 8.4 Adding HTTP Strict Transport Security

HTTP Strict Transport Security (HSTS) is a web security policy mechanism that protects HTTPS websites against downgrade attacks. Downgrade attacks are often implemented as part of a man-in-the-middle attack such as Poodle. HSTS also protects against cookie hijacking. It enables web servers to mandate that web browsers (or other complying user agents) should interact with it by using only secure HTTPS connections, and never through the insecure HTTP protocol.

HSTS are enabled by default for all Access Manager components.

## 8.5 Disabling SSL Renegotiations

SSL renegotiation is vulnerable to the man-in-the-middle attacks. It is disabled by default for all components except Analytics Server.

For information about how to disable it in Analytics Server, see [Disabling SSL Renegotiations](#).

## 8.6 Customizing the Size of Ephemeral Diffie-Hellman Keys

It is recommended not to use keys of sizes less than 1024 bits because of their insufficient strength. In Access Manager 4.3 and later, the default EDH key size in Administration Console and Identity Server is 2048.

For information about how to customize the size of EDH keys in Analytics Server, see [Section 5.1, “Customizing the Size of EDH Keys,”](#) on page 47.



# 9 Strengthening Certificates

This section discusses the following topics:

- ♦ [Section 9.1, “Key Size and Signature Algorithm Considerations,” on page 65](#)
- ♦ [Section 9.2, “Trusted Certificate Authorities,” on page 65](#)
- ♦ [Section 9.3, “Certificate Renewal,” on page 66](#)

## 9.1 Key Size and Signature Algorithm Considerations

Access Manager ships with a CA that can create certificates with a key size of 512, 1024, 2048, or 4096. Select the maximum size supported by the applications that you are protecting with Access Manager. Security increases with the increase in key size length. The default certificates created by Access Manager 4.2 and later are of 2048 key size. If you are upgrading Access Manager from a version older than 4.2, ensure that certificates with small key sizes are replaced with 2048 or above.

In signature algorithms, SHA1 is no longer considered secure. Access Manager supports creation of a certificate only with SHA-256 and SHA-512. When you are importing external certificates signed by a well-known third-party CA into Access Manager, ensure that they are of SHA-256 or above.

## 9.2 Trusted Certificate Authorities

Access Manager ships with a CA. During installation, Access Manager CA creates and distributes certificates. For added security, replace these certificates with certificates from a well-known CA.

To use certificates signed by an external CA, perform the following activities:

1. Obtain externally signed certificates.

For more information, see [“Obtaining Externally Signed Certificates”](#) in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

2. Configure Identity Server to use externally signed certificates.

For more information, see [“Configuring Identity Server to Use an Externally Signed Certificate”](#) in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

3. Configure Access Gateway to use externally signed certificates.

For more information, see [“Configuring Access Gateway to Use an Externally Signed Certificate”](#) in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

## 9.3 Certificate Renewal

Ensure that you renew certificates before it gets expired. Your security needs might allow for a longer or shorter period. You can configure to get certificate expiration notifications.

For more information, see “[Getting the Certificate Expiration Notification](#)” in the *NetIQ Access Manager CE 24.2 (v5.1) Best Practices Guide*.

When you install Administration Console, the following test certificates are automatically generated:

test-signing  
test-encryption  
test-connector  
test-provider  
test-consumer  
test-stunnel

For strong security, it is recommended that you replace these certificates, except the test-stunnel certificate, with certificates from a well-known certificate authority.

Ten years after you install Administration Console, new versions of these certificates are automatically generated as the old certificates expire. If you are using any of the test certificates in your configuration, Administration Console cannot use the new version until you reboot the machine.

Access Manager renews test-\* certificate for both primary and secondary Administration Console including the eDirectory certificate on secondary Administration Console automatically.

Certificates created manually by using Access Manager CA does not get renewed automatically.

Perform the following steps to renew manually created certificates. Lets assume that a certificate with the alias *signing* in the Identity Server signing keystore is about to expire.

- 1 Create a new certificate. (**Security > Certificates > New**)
- 2 Add the new certificate to the keystore with the alias of the certificate that will expire (*signing*). (**Actions > Add Certificate to Keystores**)
- 3 Select the option to overwrite.



# 10 XSS, XFS, and Clickjacking Attacks

This section included the following topics:

- ◆ [Section 10.1, “Cross-site Scripting Attacks,” on page 67](#)
- ◆ [Section 10.2, “Cross-Frame Scripting Attacks,” on page 70](#)
- ◆ [Section 10.3, “Clickjacking Attacks,” on page 70](#)

## 10.1 Cross-site Scripting Attacks

By default, Access Manager performs extensive checks to prevent Cross-site Scripting (XSS) attacks. However, Access Manager does not validate a JSP file if you have customized it. If you modify JSP files to customize the login, logout, error pages, and so forth, you must sanitize the respective JSP file to prevent XSS attacks.

Perform either one of the following options to sanitize the customized JSP file:

- ◆ [Option 1: HTML Escaping](#)
- ◆ [Option 2: Filtering](#)
- ◆ [Option 3: Understanding Relaxed Query Parameters](#)

### 10.1.1 Option 1: HTML Escaping

Perform the following XSS checks for the customized JSP file to protect it from possible XSS attacks. For more information about XSS prevention techniques, see [XSS \(Cross Site Scripting\) Prevention Cheat Sheet](#).

Perform the following steps:

- 1 Verify if the `org.apache.commons.lang.StringEscapeUtils` class is available in the JSP file.

For information about how to edit a file, see “[Modifying Configurations](#)” in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

For example, the following import statement should be available in the import section of the JSP file:

```
<%@ page import="org.apache.commons.lang.StringEscapeUtils"%>
```

---

**NOTE:** The `escapeHtml` function supports all known HTML 4.0 entities but does not escape the character `'`, `"&apos;"` as it is not a legal entity.

Refer to the example below:

- ◆ Refer to the file: err\_latest.jsp

- ◆ Add the below import statement in the JSPs

```
<%@ page import="org.apache.commons.lang.StringEscapeUtils" %>
```

- ◆ Example of using the StringEscapeUtils

```
strUIComponentsToHide = (String)
```

```
StringEscapeUtils.escapeHtml(request.getParameter(NIDPConstants.HTTP_REQUEST_PARAM_NAME_UICOMPONENTS_TOHIDE))
```

When the `getAttribute` and `getParameter` methods are invoked on request, these strings should be passed to this `StringEscapeUtility` class by using `escapeHtml` method.

- ◆ Handling HTML 4.0 entities while using StringEscapeUtils:

For Example "&apos;" that the commonly used apostrophe escape character (&apos;) is not a legal entity and so is not supported.

```
strUIComponentsToHide = strUIComponentsToHide.replace("'", "&apos;");
```

See <https://commons.apache.org/proper/commons-lang/apidocs/org/apache/commons/lang3/StringEscapeUtils.html> and <https://commons.apache.org/proper/commons-text/javadocs/api-release/org/apache/commons/text/StringEscapeUtils.html>.

---

- 2 Verify if all URL query parameter values are sanitized.

The following code snippet sample shows how URL query parameter values (uname and target) can be sanitized:

```
<%//Fetch the values from URL query parametersString target = (String)
request.getAttribute("target");String uname = (String)
request.getAttribute("username"); String sanitizedUName = ""; if (uname
!= null){//Sanitize the value assigned to uname sanitizedUName =
StringEscapeUtils.escapeHtml(uname); } String sanitizedTarget = ""; if
(target != null){ //Sanitize the value assigned to target query param
sanitizedTarget = StringEscapeUtils.escapeHtml(target);} %>
```

- 3 Add double quotes ("" ) in value attribute (or any attribute that is dynamically assigned) for any HTML element that get assigned with above URL query param value.

```
<!-- The last 2 double quotes are mandatory to prevent XSS attacks --
><input type="text" class="smalltext" name="Ecom_User_ID" size="30"
value="<%=sanitizedUName%>">.....<!-- The last 2 double quotes are
mandatory to prevent XSS attacks --><input type="hidden" name="target"
value="<%=sanitizedTarget%>">
```

## 10.1.2 Option 2: Filtering

By default, the XSS detection filter is enabled in Identity Server's `web.xml` file.

For information about how to edit a file, see “[Modifying Configurations](#)” in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

The filter is as follows:

```

<filter>
    <filter-name>XSSDetectionFilter</filter-name>
    <filter-
class>com.novell.nidp.servlets.filters.xss.XSSDetectionFilter</filter-class>
description>
    <description>This filter is used to detect XSS attacks in NIDS</
description>
    <init-param>
        <param-name>active</param-name>
        <param-value>True</param-value>
    </init-param>
    <init-param>
        <param-name>level</param-name>
        <param-value>SCRIPT_TAGS</param-value>
    </init-param>
    <init-param>
        <param-name>exclude</param-name>
        <param-value>soap,wstrust,metadata,oauth</param-value>
    </init-param>
</filter>

```

---

**NOTE:** You can scan JavaScript directives along with the script directive by using the param-value SCRIPT\_TAGS\_AND\_JS\_DIRECTIVES instead of SCRIPT\_TAGS.

To improve the XSS Detection with Strict checking of the attacks, use the ALL\_PARAMETERS value for the XSSDetectionFilter, which includes SCRIPT\_TAGS\_AND\_JS\_DIRECTIVES and SCRIPT\_TAGS.

---

To disable it, set the <param-value> True to False as follows:

```

<init-param>
    <param-name>active</param-name>
    <param-value>False</param-value>
</init-param>

```

To exclude it from a specific request, add a URL string from that request in the <param-name>exclude</param-name> tag that contains the default excluded request path name.

For example: If wsfed request fails due to some reason, add wsfed in the exclude list. Now, Identity Provider will not filter wsfed specific requests.

The exclude init-param is as follows:

```

<init-param>
    <param-name>exclude</param-name>
    <param-value>soap,wstrust,metadata,oauth,wsfed</param-value>
</init-param>

```

---

**NOTE:** It is recommended to use the above option as it overrides the following approach:

---

This approach might have a minor performance impact due to the checks it performs. If you perform HTML escaping in customized JSP pages, you do not need to perform this additional filtering.

Perform the followings steps to sanitize Identity Server's customized JSP file:

- 1 The eMFrame\_xss.jar library prevents XSS based attacks.
- 2 Add a filter in Identity Server's [web.xml](#) file.

```
<filter><filter-name>XSS</filter-name><display-name>XSS</display-name><description>Filters XSS injections.</description> <filter-class>com.novell.emframe.fw.filter.CrossScriptingFilter</filter-class></filter> <filter-mapping><filter-name>XSS</filter-name><url-pattern>/*</url-pattern></filter-mapping>
```

For information about how to edit a file, see “[Modifying Configurations](#)” in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

### 10.1.3 Option 3: Understanding Relaxed Query Parameters

For more information to understand the Relaxed Query Characters, see [Apache Tomcat 9 Configuration Reference](#)(relaxedQueryChars).

## 10.2 Cross-Frame Scripting Attacks

Any intruder can call Identity Server portal login pages or the pages delivered by Access Gateway ESP with the default Identity Server configuration from an HTML iFrame. To prevent this vulnerability, Cross-Frame Scripting (XFS) has been disabled for both Identity Server and Access Gateway ESP in Access Manager 4.3 and later.

The configuration to prevent this attack is enabled by default in Access Manager 4.3 and later.

## 10.3 Clickjacking Attacks

Web applications allow external sites to include content by using iFrame. This enables an attacker to embed the malicious code beneath legitimate clickable content. An attacker can trick a web user into clicking the malicious content that the attacker can control.

The configuration to prevent this attack is enabled by default in Access Manager 4.3 and later.

# 11

## Getting the Latest Security Patches

The OpenSSL open source project team regularly releases updates to known OpenSSL vulnerabilities. Access Gateway uses the OpenSSL library for cryptographic functions. It is recommended to update Access Gateway with the latest OpenSSL patch.

### Getting the Latest Security Updates for Access Gateway Service

See “[Updating Security Patches for Access Gateway Service](#)” in the *NetIQ Access Manager CE 24.2 (v5.1) Installation and Upgrade Guide*.

### Getting the Latest Security Updates for Access Gateway Appliance

See “[Installing or Updating Security Patches for Access Gateway Appliance](#)” in the *NetIQ Access Manager CE 24.2 (v5.1) Installation and Upgrade Guide*.



# 12 Securing Access Manager Components on Cloud

When using Access Manager components on cloud, you must consider the following security recommendations in addition to the security recommendations mentioned in the previous chapters:

- ♦ [Section 12.1, “Prerequisite,” on page 73](#)
- ♦ [Section 12.2, “Protecting Administration Console on Cloud,” on page 73](#)

## 12.1 Prerequisite

- ♦ The operating system packages must be updated. If you are using SUSE or Red Hat, ensure that the operating system is updated with the latest patch update.
- ♦ It is recommended to not use keys of sizes less than 1024 bits on port 22.

## 12.2 Protecting Administration Console on Cloud

It is recommended to install Administration Console on a private subnet to restrict the access to Config data store. To protect Administration Console in a cloud environment, see [Deploying Access Manager on Public Cloud](#) in the [NetIQ Access Manager CE 24.2 \(v5.1\) Installation and Upgrade Guide](#).





# 13 Restoring Previous Security Level After Upgrading Access Manager

All protocols, ciphers, and filter configurations in all components are made highly secure by default in Access Manager 4.3 and later. If your Access Manager setup is configured with less secure settings, upgrading it to 4.3 or later may result in communications issues. The following are few example scenarios when you may need to restore your previous security settings:

- ◆ When browsers do not support TLS1.1 or TLS1.2 protocol or secure ciphers suites.
- ◆ When third-party service provider does not support TLS1.1 or TLS1.2 protocol or secure cipher suites along with the following configuration:
  - ◆ A SAML or Liberty federation with artifact binding between Access Manager and third-party service provider.
  - ◆ WS-Trust federation between Access Manager and third-party service provider.
- ◆ When OAuth clients or OAuth resource servers do not support TLS1.1 or TLS1.2 protocol or secure cipher suites.

## Backed Up Configuration Files

When you upgrade Access Manager, the upgrade script backs up the following files to enable you restoring the previous configuration:

- ◆ **Administration Console:** tomcat8.conf, server.xml, web.xml
- ◆ **Identity Server:** tomcat.conf, server.xml, web.xml
- ◆ **Access Gateway:** web.xml, httpd.conf, NovellAgSettings.conf, tomcat.conf, sever.xml

The backup files are located in `/root/nambackup` (separate folders for Administration Console, Identity Server, and Access Gateway).

---

**NOTE:** Compare each upgraded configuration file with the corresponding backup file. If the backup file includes the similar configuration as it is in the upgraded file, you do not need to make any changes.

---

## In this Chapter

- ◆ [Restoring Previous Security Settings for Administration Console](#)
- ◆ [Restoring Previous Security Settings for Identity Server](#)
- ◆ [Restoring Previous Security Settings for Access Gateway](#)

# 13.1 Restoring Previous Security Settings for Administration Console

## In this Section

- ♦ [Restoring the Previous Protocols Settings](#)
- ♦ [Restoring the Previous Settings of Ciphers for SSL Communication](#)
- ♦ [Disabling Perfect Forward Secrecy](#)
- ♦ [Restoring the Previous Size of EDH Keys](#)
- ♦ [Removing HTTP Strict Transport Security](#)

## 13.1.1 Restoring the Previous Protocols Settings

- 1 Download backup files from the `/root/nambkup/ac <time stamp of upgrade>` folder.  
For information about how to download backup files, see [“Downloading Files from a Server”](#) in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.
- 2 Open the backup `server.xml` file from the backup folder and search for the `sslProtocol` attribute in `NIDP_Name="devman"` and `NIDP_Name="connector"` inside the `Connector` element and copy the attribute values.
- 3 Open Administration Console’s new `server.xml`.  
For information about how to open and modify a file, see [“Modifying Configurations”](#) in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.
- 4 Search for the `sslProtocol` attribute in the `NIDP_Name="devman"` and `NIDP_Name="connector"` inside the `Connector` element. You will see the following value:  

```
sslProtocol="TLSv1.2" sslEnabledProtocols="SSLv2Hello,TLSv1.1,TLSv1.2"
```
- 5 Replace this attribute value with the previous value that you copied in step 4.

## 13.1.2 Restoring the Previous Settings of Ciphers for SSL Communication

- 1 Download backup files from the `/root/nambkup/ac <time stamp of upgrade>` folder.  
For information about how to download backup files, see [“Downloading Files from a Server”](#) in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.
- 2 Open the backup `server.xml` file from the backup folder and search for the `cipher` attribute in `NIDP_Name="devman"` and `NIDP_Name="connector"` inside the `Connector` element and copy the list of ciphers.
- 3 Open the new `server.xml` of Administration Console.  
For information about how to open and modify a file, see [“Modifying Configurations”](#) in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.
- 4 Search for the `cipher` attribute in `NIDP_Name="devman"` and `NIDP_Name="connector"` inside the `Connector` element.
- 5 Replace this list of ciphers with the list copied in step 2.

### 13.1.3 Disabling Perfect Forward Secrecy

- 1 Download backup files from the `/root/nambkup/ac <time stamp of upgrade>` folder. For information about how to download backup files, see “[Downloading Files from a Server](#)” in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.
- 2 Open the backup `server.xml` from the backup folder and search for the cipher attribute in `NIDP_Name="devman"` and `NIDP_Name="connector"` inside the `<Connectors>` element and copy the list of ciphers.
- 3 Open the new `server.xml` file of Administration Console. For information about how to open and modify a file, see “[Modifying Configurations](#)” in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.
- 4 Search for the cipher attribute in `NIDP_Name="devman"` and `NIDP_Name="connector"` inside the `<Connectors>` element.
- 5 Replace the list of ciphers with the value you copied in step 2.
- 6 Remove the `useServerCipherSuitesOrder` attribute.

### 13.1.4 Restoring the Previous Size of EDH Keys

- 1 Open the `tomcat8.conf` ([tomcat.conf](#)) file.
- 2 Remove the following line:

```
JAVA_OPTS="{JAVA_OPTS} -Djdk.tls.ephemeralDHKeySize=2048"
```

For information about how to modify a file, see “[Modifying Configurations](#)” in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

### 13.1.5 Removing HTTP Strict Transport Security

- 1 Open the Administration Console `web.xml` file and comment out the `httpHeaderSecurity` filter definition.

```
<filter>
<filter-name>httpHeaderSecurity</filter-name>
<filter-class>org.apache.catalina.filters.HttpHeaderSecurityFilter</
filter-class>
<async-supported>true</async-supported>
</filter>
```

- 2 Comment out the following parameter that sets up an appropriate maximum age value:

```
<init-param>
<param-name>hstsMaxAgeSeconds</param-name>
<param-value>31536000</param-value>
</init-param>
```

- 3 Comment out the filter mapping.

```
<filter-mapping>  
<filter-name>httpHeaderSecurity</filter-name>  
<url-pattern>/*</url-pattern>  
<dispatcher>REQUEST</dispatcher>  
</filter-mapping>
```

For information about how to modify a file, see “[Modifying Configurations](#)” in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

## 13.2 Restoring Previous Security Settings for Identity Server

### In this Section

- ◆ [Restoring the Previous Protocols Settings](#)
- ◆ [Restoring the Previous Settings of Ciphers for SSL Communication](#)
- ◆ [Disabling Perfect Forward Secrecy](#)
- ◆ [Restoring the Previous Settings of the Size of EDH Keys](#)
- ◆ [Removing HTTP Strict Transport Security](#)
- ◆ [Removing the Clickjacking Filter](#)

### 13.2.1 Restoring the Previous Protocols Settings

- 1 Download backup files from the `/root/nambkup/idp <time stamp of upgrade>` folder.  
For information about how to download backup files, see “[Downloading Files from a Server](#)” in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.
- 2 Open the backup `server.xml` file from the backup folder, search for the `sslProtocol` attribute, and copy the value.
- 3 Open Identity Server’s new `server.xml` file and search for the `sslProtocol` attribute.  
For information about how to open and modify a file, see “[Modifying Configurations](#)” in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

You will see the following value:

```
sslProtocol="TLSv1.2" sslEnabledProtocols="SSLv2Hello,TLSv1.1,TLSv1.2"
```

- 4 Replace this attribute value with the value that you copied in step 2.

### 13.2.2 Restoring the Previous Settings of Ciphers for SSL Communication

- 1 Download backup files from the `/root/nambkup/idp <time stamp of upgrade>` folder.  
For information about how to download backup files, see “[Downloading Files from a Server](#)” in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.
- 2 Open the backup `server.xml` from the backup folder, search for the cipher attribute in `NIDP_Name="connector"` inside the `<Connectors>` element, and copy the list of ciphers.

- 3 Open Identity Server's new [server.xml](#) and search for the `cipher` attribute in `NIDP_Name="connector"` in the `<Connector>` element.  
For information about how to open and modify a file, see ["Modifying Configurations"](#) in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.
- 4 Replace this list of ciphers with the list copied in step 2.

### 13.2.3 Disabling Perfect Forward Secrecy

- 1 Download backup files from the `/root/nambkup/idp <time stamp of upgrade>` folder.  
For information about how to download backup files, see ["Downloading Files from a Server"](#) in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.
- 2 Open the backed up `server.xml` from the backup folder, search for the `cipher` attribute in `NIDP_Name="connector"` inside the `<Connectors>` element, and copy the list of ciphers.
- 3 Open Identity Server's new [server.xml](#) file. Search for the `cipher` attribute in `NIDP_Name="connector"` in the `<Connectors>` element.  
For information about how to open and modify a file, see ["Modifying Configurations"](#) in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.
- 4 Replace the list of ciphers with the value you copied in step 2.

### 13.2.4 Restoring the Previous Settings of the Size of EDH Keys

- 1 Open Identity Server's [tomcat.conf](#).
- 2 Remove the following line:

```
JAVA_OPTS="$ {JAVA_OPTS} -Djdk.tls.ephemeralDHKeySize=2048 "
```

For information about how to open and modify a file, see ["Modifying Configurations"](#) in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

### 13.2.5 Removing HTTP Strict Transport Security

- 1 Open Identity Server's [web.xml](#) file and comment out the `httpHeaderSecurity` filter definition.

```
<filter>
  <filter-name>httpHeaderSecurity</filter-name>
  <filter-
class>org.apache.catalina.filters.HttpHeaderSecurityFilter</filter-
class>
  <async-supported>true</async-supported>
</filter>
```

- 2 Comment out the `hstsMaxAgeSeconds` parameter:

```
<init-param>
  <param-name>hstsMaxAgeSeconds</param-name>
  <param-value>31536000</param-value>
</init-param>
```

- 3 Comment out the filter mapping.

```
<filter-mapping>
  <filter-name>httpHeaderSecurity</filter-name>
  <url-pattern>/*</url-pattern>
  <dispatcher>REQUEST</dispatcher>
</filter-mapping>
```

For information about how to open and modify a file, see [“Modifying Configurations”](#) in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

## 13.2.6 Removing the Clickjacking Filter

- 1 Open Identity Server’s [web.xml](#) file.
- 2 Comment out the following Tomcat filter configuration:

```
<filter>
  <filter-name>TomcatSameOriginFilter</filter-name>
  <filter-class>org.apache.catalina.filters.HttpHeaderSecurityFilter</
filter-class>
  <init-param>
    <param-name>antiClickJackingOption</param-name>
    <param-value>SAMEORIGIN</param-value>
  </init-param>
</filter>
<filter-mapping>
  <filter-name>TomcatSameOriginFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

For information about how to open and modify a file, see [“Modifying Configurations”](#) in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

## 13.3 Restoring Previous Security Settings for Access Gateway

### In this Section

- ♦ [Restoring the Previous Protocol Settings between Browsers and Access Gateway](#)
- ♦ [Restoring the Previous Ciphers Settings between Browsers and Access Gateway](#)
- ♦ [Removing the Clickjacking Filter](#)
- ♦ [Removing HTTP Strict Transport Security](#)

### 13.3.1 Restoring the Previous Protocol Settings between Browsers and Access Gateway

- 1 Download backup files from the `/root/nambkup/mag <time stamp of upgrade>/conf` folder.

For information about how to download backup files, see [“Downloading Files from a Server”](#) in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

- 2 Open the `NovellAgSettings.conf` file, search for `SSL Protocol`, and copy the value associated with it.

For information about how to open and modify a file, see “[Modifying Configurations](#)” in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

- 3 On the Home page, click **Access Gateways > Edit > Advanced Options** and replace the following configuration with the value copied in `NovellAgSettings.conf` in step 2:

```
SSLProtocol TLSv1.1 +TLSv1.2
```

## 13.3.2 Restoring the Previous Ciphers Settings between Browsers and Access Gateway

- 1 Download the backup files from the `/root/nambkup/mag <time stamp of upgrade>/conf` folder.

For information about how to download backup files, see “[Downloading Files from a Server](#)” in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

- 2 Open the `NovellAgSettings.conf` file, search for `SSL`, and copy the value.

For information about how to open and modify a file, see “[Modifying Configurations](#)” in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

- 3 On the Home page, click **Access Gateways > Edit > Advanced Options** and replace the following configuration with the value copied from `NovellAgSettings.conf` in step 2:

```
SSLCipherSuite !aNULL:!eNULL:!EXPORT:!DSS:!DES:!RC4:ALL:!EDH
```

If `NovellAgSettings.conf` does not contain this line, delete this line in Access Gateway Advanced Options.

## 13.3.3 Removing the Clickjacking Filter

In Access Gateway’s `web.xml`, comment out the following Tomcat filter configuration:

```
<filter>
  <filter-name>TomcatSameOriginFilter</filter-name>
  <filter-class>org.apache.catalina.filters.HttpHeaderSecurityFilter</
filter-class>
  <init-param>
    <param-name>antiClickJackingOption</param-name>
    <param-value>SAMEORIGIN</param-value>
  </init-param>
</filter>
<filter-mapping>
  <filter-name>TomcatSameOriginFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

For information about how to open and modify a file, see “[Modifying Configurations](#)” in the *NetIQ Access Manager CE 24.2 (v5.1) Administration Guide*.

## 13.3.4 Removing HTTP Strict Transport Security

1 On the Home page, click **Access Gateways > Edit > Advanced Options**.

2 Set the following option:

```
SetStrictTransportSecurity off
```

3 Restart Apache.

```
/etc/init.d/novell-apache2 restart OR rcnovell-apache2 restart
```



# A Appendix

The following lists the default ciphers for the Identity Server, Administration Console, Analytics Server, and Access Gateway.

- ◆ [Section A.1, “Default Ciphers for Identity Server,” on page 83](#)
- ◆ [Section A.2, “Default Ciphers for Administration Console,” on page 83](#)
- ◆ [Section A.3, “Default Ciphers for Analytics Server,” on page 84](#)
- ◆ [Section A.4, “Default Ciphers for Access Gateway,” on page 84](#)

## A.1 Default Ciphers for Identity Server

The following is the list of default ciphers supported by Access Manager Identity Server:

- ◆ TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256
- ◆ TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA256
- ◆ TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256
- ◆ TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256
- ◆ TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_GCM\_SHA384
- ◆ TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384
- ◆ TLS\_RSA\_WITH\_AES\_256\_CBC\_SHA256
- ◆ TLS\_DHE\_RSA\_WITH\_AES\_256\_CBC\_SHA256
- ◆ TLS\_DHE\_DSS\_WITH\_AES\_256\_CBC\_SHA256
- ◆ TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA256
- ◆ TLS\_DHE\_RSA\_WITH\_AES\_128\_CBC\_SHA256
- ◆ TLS\_DHE\_DSS\_WITH\_AES\_128\_CBC\_SHA256

## A.2 Default Ciphers for Administration Console

The following is the list of default ciphers supported by Access Manager Administration Console:

- ◆ TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256
- ◆ TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA256
- ◆ TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256
- ◆ TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256
- ◆ TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_GCM\_SHA384
- ◆ TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384
- ◆ TLS\_RSA\_WITH\_AES\_256\_CBC\_SHA256

- ◆ TLS\_DHE\_RSA\_WITH\_AES\_256\_CBC\_SHA256
- ◆ TLS\_DHE\_DSS\_WITH\_AES\_256\_CBC\_SHA256
- ◆ TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA256
- ◆ TLS\_DHE\_RSA\_WITH\_AES\_128\_CBC\_SHA256
- ◆ TLS\_DHE\_DSS\_WITH\_AES\_128\_CBC\_SHA256

## A.3 Default Ciphers for Analytics Server

The following is the list of default ciphers supported by Access Manager Analytics Server:

- ◆ TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256
- ◆ TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA256
- ◆ TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256
- ◆ TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256
- ◆ TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_GCM\_SHA384
- ◆ TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384
- ◆ TLS\_RSA\_WITH\_AES\_256\_CBC\_SHA256
- ◆ TLS\_DHE\_RSA\_WITH\_AES\_256\_CBC\_SHA256
- ◆ TLS\_DHE\_DSS\_WITH\_AES\_256\_CBC\_SHA256
- ◆ TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA256
- ◆ TLS\_DHE\_RSA\_WITH\_AES\_128\_CBC\_SHA256
- ◆ TLS\_DHE\_DSS\_WITH\_AES\_128\_CBC\_SHA256

## A.4 Default Ciphers for Access Gateway

The following is the list of default ciphers supported by Access Manager Access Gateway:

- ◆ ECDHE-ECDSA-AES128-GCM-SHA256
- ◆ ECDHE-RSA-AES128-GCM-SHA256
- ◆ ECDHE-ECDSA-AES256-GCM-SHA384
- ◆ ECDHE-RSA-AES256-GCM-SHA384
- ◆ ECDHE-ECDSA-CHACHA20-POLY1305
- ◆ ECDHE-RSA-CHACHA20-POLY1305
- ◆ DHE-RSA-AES128-GCM-SHA256
- ◆ DHE-RSA-AES256-GCM-SHA384