NetIQ

Access Manager Appliance 4.5

Security Guide

April 2019

Legal Notice

Contents

About this Book and the Library

The *Security Guide* is intended to help 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.netiq.com/documentation/access-manager/index.html)
- Access Manager Developer Resources (https://www.netiq.com/documentation/access-manager-45-developer-documentation/)

NOTE: Contact namsdk@microfocus.com for any query related to Access Manager SDK.

1 Deployment Considerations

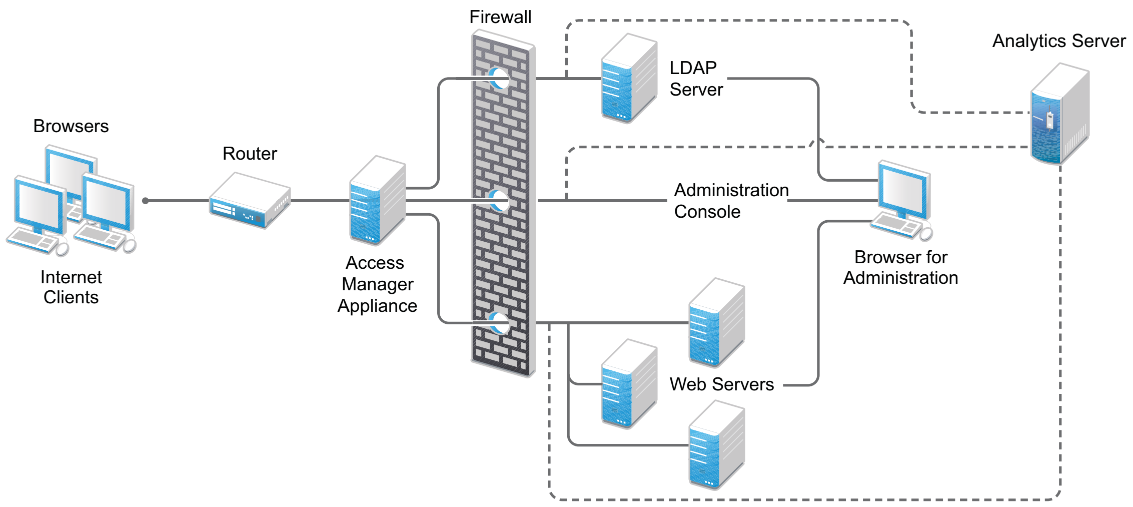
This section includes the following topic:

- Section 1.1, “Protecting Access Manager Appliance through Firewall,” on page 9

1.1 Protecting Access Manager Appliance through Firewall

Access Manager Appliance should be used with firewalls. Figure 1-1 illustrates a simple firewall setup for a basic Access Manager Appliance configuration.

Figure 1-1 *Access Manager Appliance and Firewall*



1.1.1 Access Manager Appliance in DMZ

First Firewall: If you place a firewall between browsers and Access Manager Appliance, 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 Appliance 4.5 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 Appliance 4.5 Installation and Upgrade Guide*.

You need to open ports on the second firewall according to the offered services.

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:

2.1 Managing Administration Console Session Timeout

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

1. Change to the Tomcat configuration directory:

   ```
   /opt/novell/nam/adminconsole/conf/web.xml
   ```

2. Open the `web.xml` file in a text editor and search for the `<session-timeout>` parameter.
3. Modify the value and save the file.
4. Restart Administration Console:

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

2.2 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**.

2.3 Securing Administrator Accounts

The admin user you create while installing Administration Console has all rights to Access Manager Appliance 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 Enforcing Case-Sensitive Universal Passwords (https://www.netiq.com/documentation/edirectory-91/edir_admin/data/b1j691df.html).
- **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 Appliance. 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 "Creating Multiple Admin Accounts" in the *NetIQ Access Manager Appliance 4.5 Administration Guide*.

2.4 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 Appliance 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 Appliance 4.5 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 Access Manager Appliance" in the *NetIQ Access Manager Appliance 4.5 Administration Guide*.

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

2.5 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 (ndsd) daemon.

3 Click **Admin > Manage Roles and Tasks**.

4 Navigate to **LDAP > LDAP Options > View LDAP Servers**.

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.

Run the following commands:

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

8 Restart Tomcat.

`/etc/init.d/novell-ac restart` or `rcnovell-ac restart`

2.6 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
```

3. 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
```

4. Restart the eDirectory services:

```
/etc/init.d/ndsd restart
```

2.7 Preventing 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 file of Administration Console and Identity Server.

Perform the following steps:

Administration Console

1. Modify the cipher list in the `/opt/novell/nam/adminconsole/conf/server.xml` file 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
```

2. Restart Administration Console.

Identity Server

1. Modify the cipher list in the `/opt/novell/nam/idp/conf/server.xml` file 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
```

2. Restart Identity Server.

2.8 Default Security Settings in Configuration Files

- Section 2.8.1, “server.xml,” on page 15
- Section 2.8.2, “web.xml,” on page 15
- Section 2.8.3, “tomcat7.conf,” on page 16

2.8.1 server.xml

`/opt/novell/nam/adminconsole/conf`

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

```
<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="xxxxxxxxxxxxxxx"
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.

2.8.2 web.xml

`/opt/novell/nam/adminconsole/conf`

```
<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` as long as it is not within any existing tag.

2.8.3 tomcat7.conf

```
/opt/novell/nam/adminconsole/conf/tomcat7.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"
```

3 Securing Identity Server

This section includes the following topics:

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 **Devices** > **Identity Servers** > **Edit** and ensure that you deselect any unused protocol from the list under **Enabled Protocols**.

3.2 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 well 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 governed 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 Appliance 4.5 Administration Guide*.

The authentication contract, either 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 configuring the authentication methods, see "Configuring Authentication" in the *NetIQ Access Manager Appliance 4.5 Administration Guide*.

For more information about extending the authentication mechanisms, see "Identity Server Authentication API" in the *NetIQ Access Manager 4.5 SDK Guide*.

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

For securing the cookies to prevent session replay attacks, enable Advanced session Assurance. For more information, see "Setting Up Advanced Session Assurance" in the *NetIQ Access Manager Appliance 4.5 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 the 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. If your system needs this type of contract, you can re-create it from the method. To delete these contracts, go to Administration Console and click **Identity Servers > Servers > Edit > Local > Contracts**.

If you are using 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 the Access Manager Advanced Authentication plug-in. The following are supported authentication methods:

- Email Method
- Emergency Password Method
- FIDO U2F Method
- HOTP Method
- Password (PIN) Method
- RADIUS Method
- Security Questions Method
- Smartcard Method Support
- Smartphone Method

- SMS Method
- TOTP Method
- Voice Call Method

For more information about this authentication framework, see the product page and the Advanced Authentication Documentation.

3.3 Configuring SSL Communication between Browsers and Identity Server

See Section 7.1.2, "Enabling SSL between Browsers and Identity Server," on page 44.

3.4 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 45.

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 Section 7.1.3, "Enabling SSL between Identity Server and a Service Provider," on page 45.

Another way for SAML is the signing and/or encryption of assertions. For more information, see Section 3.5.2, "Configuring the Encryption Method for the SAML Assertion," on page 20.

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 Appliance 4.5 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 may 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 TDES (Triple Data Encryption Algorithm) or AES256 (Advanced Standard Encryption, 256-bit), you can modify the Tomcat `web.xml` file and specify your required method. To use PKCS 2.0 (RSA-OAEP) for encryption, see TID.

1. Open the `web.xml` file.

   Linux: `/opt/novell/nam/idp/webapps/nidp/WEB-INF/`

   Windows Server 2012: `\Program Files\Novell\Tomcat\webapps\nidp\WEB-INF/`

2. Add the following lines to the file:

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

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

3 Save the file and copy it to each Identity Server in the cluster.

4 Restart Tomcat on each Identity Server in the cluster.

Linux: Enter the following command:

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

```
rcnovell-idp restart
```

Windows: Enter the following commands:

```
net stop Tomcat8
```

```
net start Tomcat8
```

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#tripledes-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 21
- Section 3.6.2, “Configuring a Whitelist of Intersite Transfer Service Target URL,” on page 22
- Section 3.6.3, “Configuring a Whitelist of Assertion Consumer Service URL,” on page 22

3.6.1 Configuring a Global Whitelist of Target URL

1 Click **Devices** > **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.

Liberty: The target parameter is filtered. If the requested target is not the white list, the Identity Server does not login.

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 Click **Devices** > **Identity Servers** > **Edit** > *[Liberty, 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 Click **Devices** > **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**
SAML2_ACS_URL_RESTRICT	True	If true, Identity Server allows authentication only to the trusted ACS URLs.
SAML2_ACS_DOMAIN_WHITELIST	Domain names separated with semi-colon (;)	Identity Server performs additional validation of the authentication request with the ACS domain whitelist.

3.7 Blocking Access to Identity Server Pages

Identity Server has a couple of pages that authenticated users can access and which 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 User Portal Page" and "Blocking Access to the WSDL Services Page" in the *NetIQ Access Manager Appliance 4.5 Administration Guide*.

3.8 Preventing the Error Page to Display the Tomcat Version

Accessing a non-existing page or providing wrong credentials on a protected page throws an HTTP 401 error with the Tomcat version. This issue happens on the Windows platform in the following scenarios:

- When Identity Server is the only component installed on the Windows server.
- Access Gateway Service is installed on Windows.

This issue does not happen on the Linux platform.

Perform the following steps to stop error pages to display the Tomcat version:

1 Go to `C:\Program Files\Novell\Tomcat\lib` and run `"C:\Program Files\Java\<jdk version>\bin\jar" -xf catalina.jar`

2 Move `catalina.jar` to another folder.

3 Go to `C:\Program Files\Novell\Tomcat\lib\org\apache\catalina\util` and edit the `serverInfo.properties` file:

  3a Remove `Apache Tomcat/7.0.23` from the line `server.info=`.

  3b Remove `7.0.23.0` from the line`server.number=`.

  3c Remove `Nov 20 2011 07:36:25` from the line `server.built=`.

4 Go to `C:\Program Files\Novell\Tomcat\lib` and run `jar -cf catalina.jar META-INF org`.

3.9 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 Appliancecreates a unique fingerprint of the device from which the session is established. During the session, at a configurable time interval, Access Manager Appliance 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.

For more information, see "Setting Up Advanced Session Assurance" in the *NetIQ Access Manager Appliance 4.5 Administration Guide*.

3.10 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 `/opt/novell/nids/lib/webapp/WEB-INF/web.xml`. You can modify filter's values depending on the requirement.

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

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

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

3.11 Enabling reCAPTCHA

Access Manager 4.4.1 and later versions support 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.

For more information, see Enabling reCAPTCHA (https://www.netiq.com/documentation/access-manager-45-appliance/admin/data/authclasseslist.html#recaptcha).

3.12 Preventing SWEET32 Attack

See Preventing SWEET32 Attack.

3.13 Restricting the Direct Access to Files in the nidp Folder

(Access Manager 4.5 Service Pack 4 and later)

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 `/opt/novell/nids/lib/webapp/WEB-INF/web.xml` file.
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.
4. Restart Identity Server by running the `rcnovell-idp restart` command.

3.14 Default Security Settings in Configuration Files

3.14.1 server.xml

`/opt/novell/nam/idp/conf`

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

```
<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_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_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="xxxxxxxxxxxxxxx"
maxThreads="600" minSpareThreads="5" port="8443" scheme="https"
secure="true"
sslImplementationName="com.example.nidp.common.util.net.server.NIDPSSLImpl
ementati
on" useServerCipherSuitesOrder="true" sslProtocol="TLSv1.2"
sslEnabledProtocols="SSLv2Hello,TLSv1.1,TLSv1.2" />
```

For information about connector attributes, see Apache Tomcat Configuration Reference.

3.14.2 web.xml

/opt/novell/nam/idp/webapps/nidp/WEB-INF/

```
<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` as long as it is not within any existing tag.

3.14.3 tomcat.conf

`/opt/novell/nam/idp/conf/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"
```

3.15 Configuring the Cookie Secure Flag

Identity Provider 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.

Navigate to `/opt/novell/nam/idp/webapps/nidp/WEB-INF/web.xml` and add the following entry:

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

4 Securing Access Gateway

In this Chapter

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.

- Section 4.2.1, "Securing the Embedded Service Provider Session Cookie," on page 29
- Section 4.2.2, "Securing the Proxy Session Cookie," on page 30

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 46.

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

1 On Access Gateway server, log in as an admin user.
2 Change to the Tomcat configuration directory.
   `/opt/novell/nam/mag/conf/`
3 In a text editor, open the `server.xml` file.
4 Search for the connector on port 9009.
5 Add the following parameter within the `Connector` element:
   ```
   secure="true"
   ```
6 Save the `server.xml` file.
7 Restart Tomcat.

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 30
- Section 4.2.2.2, "Preventing Cross-Site Scripting Vulnerabilities," on page 31

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 Click **Devices** > **Access Gateways** > **Edit** > **Reverse Proxy / Authentication**.
2 Select **Enable Secure Cookies.**

This option is used to secure the 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 Click **Devices** > **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 `/opt/novell/nam/mag/webapps/nesp/WEB-INF/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`.

3 Restart Tomcat.

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:

- Click **Devices** > **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.

- Click **Devices > 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 Click **Devices > 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 SSLLabs. 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
   ```

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, PFS is enabled.

Click **Devices > 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 Perfect Forward Secrecy (PFS) and prerequisites for enabling it, see Section 8.3, “Enabling Perfect Forward Secrecy,” on page 52.

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:

`/opt/novell/nam/mag/webapps/nesp/WEB-INF/web.xml`

```
<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. Same as above, you can configure `web.xml` for error-code 404 by adding one more `<error-page>` tag.

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 Appliancecreates a unique fingerprint of the device from which the session is established. During the session, at a configurable time interval, Access Manager Appliance 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. You should enable Advanced Session Assurance on the need basis. See "Best Practices for Enabling Advanced Session Assurance at the Proxy Service Resource Level" in the *NetIQ Access Manager Appliance 4.5 Administration Guide*.

For more information about Advanced Session Assurance and how to enable it, see "Setting Up Advanced Session Assurance" in the *NetIQ Access Manager Appliance 4.5 Administration Guide*.

4.9 AJP Communication Setting for Access Gateway

(Access Manager 4.5 Service Pack 4 and later)

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/serverl.xml` and `/opt/novell/nam/idp/conf/server.xml`.

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

1. Click **Devices** > **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/serverl.xml` and `/opt/novell/nam/idp/conf/server.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
&	&amp;
[	&#x5B;
]	&#x5D;
|	&#x7C;
{	&#x7B;
}	&#x7D;
^	&#x5E;
\	&#x5C;
`	&#x60;
"	&#x22;
<	&#x3C;
>	&#x3E;

2 Save the file.

3 Restart Access Gateway and Identity Server.

  Access Gateway: `/etc/init.d/novell-mag restart`

  Identity Server: `/etc/init.d/novell-idp restart`

4.10 Default Security Settings in Configuration Files

- Section 4.10.1, “ESP web.xml,” on page 35
- Section 4.10.2, “Access Gateway Advanced Options,” on page 35
- Section 4.10.3, “httpd.conf,” on page 35
- Section 4.10.4, “NovellAgSettings.conf,” on page 36

4.10.1 ESP web.xml

```
/opt/novell/nam/mag/webapps/nesp/WEB-INF/

<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>
```

4.10.2 Access Gateway Advanced Options

```
SSLProtocol TLSv1.1 +TLSv1.2

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

4.10.3 httpd.conf

```
/etc/opt/novell/apache2/conf
```

The mod_headers library is enabled.

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

4.10.4 NovellAgSettings.conf

`/etc/opt/novell/apache2/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"

5 Securing Analytics Server

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 47.

5.3 Disabling SSL Renegotiations

You should 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 Appliance 4.5 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 39
- Section 5.6.2, “web.xml,” on page 39

5.6.1 server.xml

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

```
<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="xxxxxxxxxxxxxxx" 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

6.1 Reconfiguring Secure Shell Ciphers

In a fresh install of Access Manager 4.3 and later, the SSH server is configured only with strong ciphers. However, in an upgraded setup, reconfigure SSH to remove the weak ciphers.

Perform the following steps:

1. In `/etc/ssh/sshd_config` (server) and `/etc/ssh/ssh_config` (client), search for `Ciphers`. The following is the default configuration:

   ```
   # Ciphers aes128-ctr,aes192-ctr,aes256-
   ctr,arcfour256,arcfour128,aes128-cbc,3des-cbc
   ```

2. Uncomment this line and replace it with the following value:

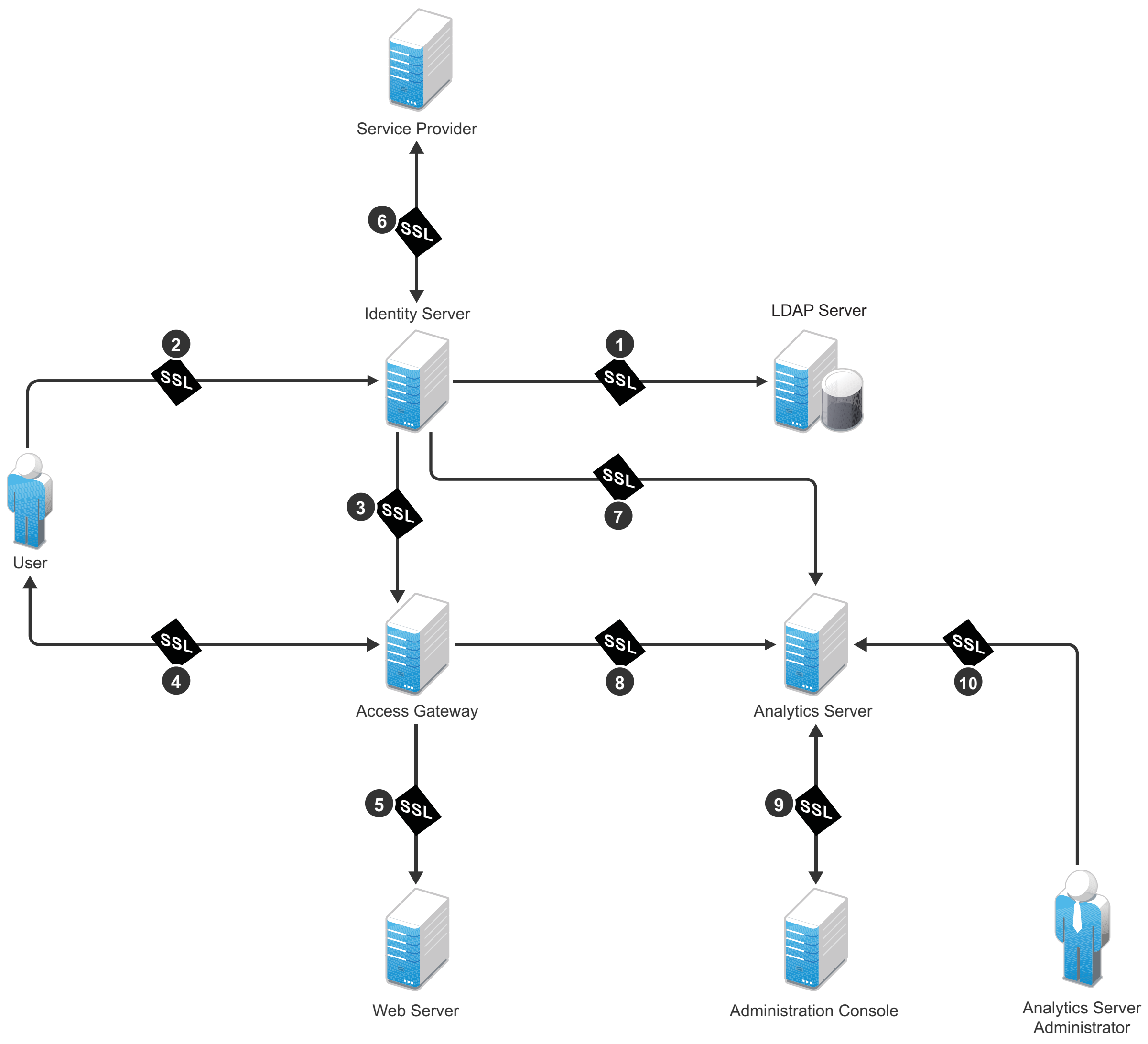
   ```
   Ciphers aes128-ctr,aes192-ctr,aes256-ctr,aes128-cbc,3des-cbc
   ```

3. Restart SSH by running the `service sshd restart` command.

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.

Topics include:

- Section 7.1.1, "Configuring a SSL Channel between Identity Server and LDAP Servers," on page 44
- Section 7.1.2, "Enabling SSL between Browsers and Identity Server," on page 44
- Section 7.1.3, "Enabling SSL between Identity Server and a Service Provider," on page 45

7.1.1 Configuring a SSL Channel between Identity Server and LDAP Servers

Channel 1 in Figure 7-1, "SSL Communication Channels," on page 43.

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, usernames 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** > **Servers** > **Edit** > **Local** > **User Stores**.
2. Click [name of the user store] > [name of the replica].
3. Select **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 43.

1. Click **Devices** > **Identity Servers** > **Edit**.
2. Change **Protocol** to HTTPS (the system changes the port to 8443).
3. In the **SSL Certificate** line, click the **Browse** icon > **Replace** 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, go to **Access Gateway** > **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 43.

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 on 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 Appliance 4.5 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 Section 7.2.1, “Enabling SSL between Browsers and Access Gateway,” on page 46.

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 43).

1 In Administration Console, click **Devices** > **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 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.

5 Click **OK** > **Reverse Proxy / Authentication**.

7.2.2 Enabling SSL between Access Gateway and Web Servers

Channel 5 in Figure 7-1, "SSL Communication Channels," on page 43.

SSL must be enabled between Access Gateway and browsers before you can enable it between Access Gateway and its web servers. See Section 7.2.1, "Enabling SSL between Browsers and Access Gateway," on page 46.

1 Click **Devices** > **Access Gateways** > **Edit** > **[Name of Reverse Proxy]** > **[Name of Proxy Service]** > **Web Servers**.

2 Select **Connect Using SSL**.

3 (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.

   You need to 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 your Web server documentation.

4 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

1 Click **Devices** > **Access Gateways** > **Edit** > **[Name of Reverse Proxy]**.

2 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.

3 Specify an **Alias** for the certificate.

4 On the Server Configuration page, click **Reverse Proxy / Authentication**.

5 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 43.

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 “Using Externally Signed Certificates” in the *NetIQ Access Manager Appliance 4.5 Administration Guide*.
2. 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 Appliance 4.5 Administration Guide*.

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 4.3 and later are 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 4.3 and later support cypher suites only with AES.
- **Message Authentication Code (MAC) Algorithm:** In these algorithms, MD5 and SHA1 are vulnerable. By default, Access Manager 4.3 and later support 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 12, “Restoring Previous Security Level After Upgrading Access Manager Appliance,” on page 61.The

This section discusses the following topics:

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 31.

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 32

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 32.

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.

In Access Manager 4.3 and later, HSTS are enabled by default for all components.

8.5 Disabling SSL Renegotiations

SSL renegotiation is vulnerable to the man-in-the-middle attacks. In Access Manager 4.3 and later, it is disabled by default for all components except Analytics Server.

For information about how to disable it in Analytics Server, see Section 5.3, “Disabling SSL Renegotiations,” on page 37.

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 37.

9 Strengthening Certificates

This section discusses the following topics:

9.1 Key Size and Signature Algorithm Considerations

Access Manager Appliance 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 Appliance. 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 Appliance 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 Appliance 4.5 Administration Guide*.
2. 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 Appliance 4.5 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 (https://www.netiq.com/documentation/access-manager-45/bestpractices/data/bzabzrh.html#bzabzrh).

When you install Administration Console, the NAM-RP certificate is automatically generated and associated with NAM-RP Reverse Proxy (**Devices** > **Access Gateways** > **[AG-Cluster]** > **[NAM-RP]**).

Access Manager renews test-* certificate for both primary and secondary Administration Console including the edir 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:

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:

- HTML Escaping. See Option 1: HTML Escaping in the NetIQ Access Manager Appliance 4.5 Administration Guide.
- Filtering. See Option 2: Filtering in the NetIQ Access Manager Appliance 4.5 Administration Guide

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 IFrames. 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 Manager Appliance uses the OpenSSL library for cryptographic functions. It is recommended to update Access Manager Appliance with the latest OpenSSL patch.

See "Installing or Updating Security Patches for Access Manager Appliance and Analytics Server" in the *NetIQ Access Manager Appliance 4.5 Installation and Upgrade Guide*.

12 Restoring Previous Security Level After Upgrading Access Manager Appliance

All protocols, ciphers, and filter configurations in all components are made highly secure by default in Access Manager Appliance 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 to Access Manager 4.3 or later, the upgrade script backs up the following files to enable you restoring the previous configuration:

- **Administration Console:** tomcat7.conf (tomcat7w.exe on Windows), server.xml, web.xml
- **Identity Server:** tomcat.conf (tomcat7w.exe on windows), server.xml, web.xml
- **Access Gateway:** web.xml, httpd.conf, NovellAgSettings.conf, tomcat.conf (tomcat8w.exe on Windows), sever.xml

The backup files are located at the following location:

`/root/nambkup` (separate folders for Administration Console, Identity Server, and Access Gateway)

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

This section includes the following topics:

12.1 Restoring Previous Security Settings for Administration Console

12.1.1 Restoring the Previous Protocols Settings

1. Open the backup `server.xml.` For location of the backup file, see "Backed Up Configuration Files" on page 61.
2. Search for the sslProtocol attribute in `NIDP_Name="devman"` and `NIDP_Name="connector"` inside the `Connector` element and copy the attribute values.
3. Change to the Tomcat configuration directory and open the 4.3 or later `server.xml` file:

   `/opt/novell/nam/adminconsole/conf`
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 2.

12.1.2 Restoring the Previous Settings of Ciphers for SSL Communication

1. Open the backup `server.xml.` For location of the backup file, see "Backed Up Configuration Files" on page 61.
2. Search for the cipher attribute in `NIDP_Name="devman"` and `NIDP_Name="connector"` inside the `Connector` element and copy the list of ciphers.
3. Change to the Tomcat configuration directory:

   `/opt/novell/nam/adminconsole/conf`
4. Open the `server.xml` file. 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.

12.1.3 Disabling Perfect Forward Secrecy

1. Open the backup `server.xml.` For location of the backup file, see "Backed Up Configuration Files" on page 61.
2. Search for the cipher attribute in `NIDP_Name="devman"` and `NIDP_Name="connector"` inside the <`Connectors`> element and copy the list of ciphers
3. Change to the Tomcat configuration directory:

```
/opt/novell/nam/adminconsole/conf
```

4. Open the `server.xml` file. 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.

12.1.4 Restoring the Previous Size of EDH Keys

1. Open the `/opt/novell/nam/adminconsole/conf/tomcat7.conf` file.
2. Remove the following line:

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

12.1.5 Removing HTTP Strict Transport Security

1. Change to the Tomcat configuration directory:

```
/opt/novell/nam/adminconsole/conf/web.xml
```

2. Open the `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>
```

3. 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>
```

4. Comment out the filter mapping.

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

12.2 Restoring Previous Security Settings for Identity Server

- Section 12.2.1, “Restoring the Previous Protocols Settings,” on page 64
- Section 12.2.2, “Restoring the Previous Settings of Ciphers for SSL Communication,” on page 64
- Section 12.2.3, “Disabling Perfect Forward Secrecy,” on page 64
- Section 12.2.4, “Restoring the Previous Settings of the Size of EDH Keys,” on page 65

12.2.1 Restoring the Previous Protocols Settings

1. Open the backup `server.xml.` For location of the backup file, see “Backed Up Configuration Files” on page 61.
2. Search for the sslProtocol attribute and copy the attribute value.
3. Change to the Tomcat configuration directory:

   Linux: `/opt/novell/nam/idp/conf`
4. Open the 4.3 or later `server.xml` file.

   Search for the `sslProtocol` attribute. 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.

12.2.2 Restoring the Previous Settings of Ciphers for SSL Communication

1. Open the backup `server.xml`. For location of the backup file, see “Backed Up Configuration Files” on page 61.
2. Search for the cipher attribute in NIDP_Name=`"connector"` inside the <`Connectors`> element and copy the list of ciphers.
3. Using command prompt, change to the Tomcat configuration directory:

   Linux: `/opt/novell/nam/idp/conf`
4. Open the 4.3 or later `server.xml` file.

   Search for the `cipher` attribute in NIDP_Name=`"connector"` inside the <`Connector`> element.
5. Replace this list of ciphers with the list copied in step 2.
6. (Conditional) If you have multiple Identity Servers in your cluster configuration, repeat these steps on each Identity Server.

12.2.3 Disabling Perfect Forward Secrecy

1. Open the backed up `server.xml`. For location of the backup file, see “Backed Up Configuration Files” on page 61.
2. Search for the cipher attribute in NIDP_Name=`"connector"` inside the <`Connectors`> element and copy the list of ciphers
3. Using command prompt, change to the Tomcat configuration directory `/opt/novell/nam/idp/conf`:
4. Open the `server.xml` file. Search for the cipher attribute in NIDP_Name=`"connector"` inside the <`Connectors`> element.
5. Replace the list of ciphers with the value you copied in step 2.

12.2.4 Restoring the Previous Settings of the Size of EDH Keys

1. Open the `/opt/novell/nam/idp/conf/tomcat.conf` file.
2. Remove the following line:

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

12.2.5 Removing HTTP Strict Transport Security

1. Change to the Tomcat configuration directory:

   `/opt/novell/nam/idp/conf/web.xml`
2. Open the `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>
   ```
3. Comment out the hstsMaxAgeSeconds parameter:

   ```
   <init-param>
       <param-name>hstsMaxAgeSeconds</param-name>
       <param-value>31536000</param-value>
   </init-param>
   ```
4. Comment out the filter mapping.

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

12.2.6 Removing the Clickjacking Filter

1. In the `/opt/novell/nids/lib/webapp/WEB-INF/web.xml` file, 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>
```

2 Restart Identity Server by running /etc/init.d/novell-idp restart.

12.3 Restoring Previous Security Settings for Access Gateway

- Section 12.3.1, "Restoring the Previous Protocol Settings between Browsers and Access Gateway," on page 66
- Section 12.3.2, "Restoring the Previous Ciphers Settings between Browsers and Access Gateway," on page 66
- Section 12.3.3, "Removing the Clickjacking Filter," on page 67
- Section 12.3.4, "Removing HTTP Strict Transport Security," on page 67

12.3.1 Restoring the Previous Protocol Settings between Browsers and Access Gateway

1 In the nambkup folder, open the `NovellAgSettings.conf` file from the `mag <time stamp of upgrade>/conf` folder.

2 Search for `SSL Protocol` and copy the value associated with it.

3 Click **Devices** > **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
```

12.3.2 Restoring the Previous Ciphers Settings between Browsers and Access Gateway

1 In the nambkup folder, open the `NovellAgSettings.conf` file from the `mag <time stamp of upgrade>/conf` folder.

2 Search for SSL and copy the value

3 Click **Devices** > **Access Gateways** > **Edit** > **Advanced Options** and replace the following configuration with the value copied in `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.

12.3.3 Removing the Clickjacking Filter

1. In the `/opt/novell/nesp/lib/webapp/WEB-INF/web.xml` file, 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>
```

2. Restart ESP by running the following command:

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

NOTE: You can also restart ESP through Administration Console. Select the cluster node > **Action** > **Service Provider** > **Restart Service Provider**.

12.3.4 Removing HTTP Strict Transport Security

1. Click **Devices** > **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
```