
Administration Guide

Advanced Authentication

Version 5.6

Legal Notice

For information about legal notices, trademarks, disclaimers, warranties, export and other use restrictions, U.S. Government rights, patent policy, and FIPS compliance, <https://www.netiq.com/company/legal/>.

Copyright © 2017 NetIQ Corporation, a Micro Focus company. All Rights Reserved.

Contents

About NetIQ Corporation	7
About this Book	9
1 Advanced Authentication Overview	11
How Is Advanced Authentication Better Than Other Solutions	11
Key Features	11
Advanced Authentication Server Components	12
Administration Portal	12
Self-Service Portal	13
Helpdesk Portal	13
Reporting Portal	13
Architecture	14
Basic Architecture	14
Enterprise Level Architecture	15
Enterprise Architecture With A Load Balancer	17
Terminologies	18
Authentication Method	18
Authentication Chain	18
Authentication Event	18
Endpoint	18
Part I Configuring Advanced Authentication	19
2 Logging In to the Advanced Authentication Administrative Portal	21
3 Configuring Advanced Authentication Server Appliance	23
Managing Dashboard	23
Adding Widgets	24
Customizing Dashboard	24
Updating Dashboard to View Real Time or Historical Data	25
Customizing the Default Widgets	25
Adding a Repository	29
Advanced Settings	31
Used Attributes	34
Local Repository	37
Configuring Methods	38
Bluetooth	39
Card	39
Email OTP	40
Emergency Password	40
Fingerprint	41
FIDO U2F	42
LDAP Password	45
OATH OTP	45
Password	47
PKI	48
RADIUS Client	50
SMS OTP	50

Security Questions	51
Smartphone	52
Swisscom Mobile ID	56
Voice	56
Voice OTP	57
Creating a Chain	58
Configuring Events	59
Configuring an Existing Event	60
Creating a Customized Event	64
Managing Endpoints	66
Configuring Policies	68
Admin UI Whitelist Policy	68
Authenticator Management Options Policy	69
Cache Options Policy	69
Delete Me Options	70
Endpoint Management Options	70
Event Categories	70
Geo Fencing Options	71
Helpdesk Options	71
Last Logon Tracking Options	71
Lockout Options	71
Login Options	72
Logon Filter for AD	72
Mail Sender	72
Password Filter for AD	74
SMS Sender	74
Services Director Options	77
SAML 2.0 Options	77
Voice Sender	77
Adding a License	79
Exporting the Database	79
4 Enrolling the Authentication Methods	81
Part II Configuring Integrations	83
5 OAuth 2.0	85
Building Blocks of OAuth 2.0	85
OAuth 2.0 Roles	85
OAuth 2.0 Grants	85
Sample OAuth 2.0 Application Integrated with Advanced Authentication	88
Running the Sample Web Application	93
OAuth 2.0 Attributes	93
Non Standard Endpoints	94
6 RADIUS Server	97
7 SAML 2.0	99
8 Examples of Integrations	101
Configuring Integration with Barracuda	101
Configure the Advanced Authentication RADIUS server:	102
Configure the Barracuda SSL VPN Appliance:	102

Authenticate in Barracuda SSL VPN Using Advanced Authentication	102
Configuring Integration with Citrix NetScaler	103
Configure the Advanced Authentication RADIUS Server	103
Configure the Citrix NetScaler Appliance	104
Authenticate in Citrix NetScaler Using Advanced Authentication	104
Configuring Integration with Dell SonicWall SRA EX-Virtual Appliance	105
Configure the Advanced Authentication RADIUS Server	105
Configure the Dell SonicWall SRA Appliance	106
Authenticate in Dell SonicWall Workspace Using Advanced Authentication	106
Configuring Integration with FortiGate	106
Configure the Advanced Authentication RADIUS Server	107
Configure the FortiGate Appliance	107
Authenticate in FortiGate Using Advanced Authentication	107
Configuring Integration with OpenVPN	108
Configure the Advanced Authentication RADIUS Server	108
Configure the OpenVPN Appliance	109
Configuring Integration with Palo Alto GlobalProtect Gateway	109
Adding the RADIUS Server	110
Adding an Authentication Profile	110
Configuring GlobalProtect Gateway	110
Configuring Integration with Salesforce	110
Configure Salesforce Domain Name	111
Configure the SAML Provider	111
Configure the Advanced Authentication SAML 2.0 Event	112
Configuring to Authenticate on Salesforce with SAML 2.0	113
Configuring Integration with ADFS	113
Configure the Advanced Authentication SAML 2.0 Event	114
Make the Corresponding Changes in ADFS	114

Part III Maintaining Advanced Authentication 117

9 Reporting 119

10 Logging 121

Syslog	121
RADIUS Logs	133
Async Logs	133

11 Searching a Card Holder's Information 135

About NetIQ Corporation

We are a global, enterprise software company, with a focus on the three persistent challenges in your environment: Change, complexity and risk—and how we can help you control them.

Our Viewpoint

Adapting to change and managing complexity and risk are nothing new

In fact, of all the challenges you face, these are perhaps the most prominent variables that deny you the control you need to securely measure, monitor, and manage your physical, virtual, and cloud computing environments.

Enabling critical business services, better and faster

We believe that providing as much control as possible to IT organizations is the only way to enable timelier and cost effective delivery of services. Persistent pressures like change and complexity will only continue to increase as organizations continue to change and the technologies needed to manage them become inherently more complex.

Our Philosophy

Selling intelligent solutions, not just software

In order to provide reliable control, we first make sure we understand the real-world scenarios in which IT organizations like yours operate—day in and day out. That's the only way we can develop practical, intelligent IT solutions that successfully yield proven, measurable results. And that's so much more rewarding than simply selling software.

Driving your success is our passion

We place your success at the heart of how we do business. From product inception to deployment, we understand that you need IT solutions that work well and integrate seamlessly with your existing investments; you need ongoing support and training post-deployment; and you need someone that is truly easy to work with—for a change. Ultimately, when you succeed, we all succeed.

Our Solutions

- ◆ Identity & Access Governance
- ◆ Access Management
- ◆ Security Management
- ◆ Systems & Application Management
- ◆ Workload Management
- ◆ Service Management

Contacting Sales Support

For questions about products, pricing, and capabilities, contact your local partner. If you cannot contact your partner, contact our Sales Support team.

Worldwide:	www.netiq.com/about_netiq/officelocations.asp
United States and Canada:	1-888-323-6768
Email:	info@netiq.com
Web Site:	www.netiq.com

Contacting Technical Support

For specific product issues, contact our Technical Support team.

Worldwide:	www.netiq.com/support/contactinfo.asp
North and South America:	1-713-418-5555
Europe, Middle East, and Africa:	+353 (0) 91-782 677
Email:	support@netiq.com
Web Site:	www.netiq.com/support

Contacting Documentation Support

Our goal is to provide documentation that meets your needs. The documentation for this product is available on the NetIQ Web site in HTML and PDF formats on a page that does not require you to log in. If you have suggestions for documentation improvements, click **Add Comment** at the bottom of any page in the HTML version of the documentation posted at www.netiq.com/documentation. You can also email Documentation-Feedback@netiq.com. We value your input and look forward to hearing from you.

Contacting the Online User Community

NetIQ Communities, the NetIQ online community, is a collaborative network connecting you to your peers and NetIQ experts. By providing more immediate information, useful links to helpful resources, and access to NetIQ experts, NetIQ Communities helps ensure you are mastering the knowledge you need to realize the full potential of IT investments upon which you rely. For more information, visit community.netiq.com.

About this Book

This Administration Guide is intended for system administrators and describes the procedure of Advanced Authentication Server appliance configuration.

Intended Audience

This book provides information for individuals responsible for understanding administration concepts and implementing a secure, distributed administration model.

1 Advanced Authentication Overview

Advanced Authentication™ is a multi-factor authentication solution that enables you to protect your sensitive data by using a more advanced way of authentication on top of the typical username and password authentication. With Advanced Authentication, you can authenticate on diverse platforms by using different types of authenticators such as Fingerprint, Card, and OTP. Advanced Authentication provides a single authentication framework that ensures secure access to all your devices with minimal administration.

Authentication comprises of the following three factors:

- ◆ Something that you know such as password, PIN, and security questions.
- ◆ Something that you have such as smartcard, token, and mobile phone.
- ◆ Something that you are such as biometrics (fingerprint or iris).

You can achieve multi-factor or strong authentication by using any two factors out of this list. For example, multi-factor authentication can include combination of a password and a token or a smartcard and a fingerprint.

This section contains the following topics:

- ◆ [“How Is Advanced Authentication Better Than Other Solutions” on page 11](#)
- ◆ [“Key Features” on page 11](#)
- ◆ [“Advanced Authentication Server Components” on page 12](#)
- ◆ [“Architecture” on page 14](#)
- ◆ [“Terminologies” on page 18](#)

How Is Advanced Authentication Better Than Other Solutions

Advanced Authentication leverages the needs of users to authenticate on different platforms with different needs. The following points explain how Advanced Authentication is different from other solutions:

- ◆ Works on multiple platforms such as Windows, Mac OS X, Linux and so on.
- ◆ Supports multi-site configurations that helps organizations to distribute the authentication globally.

Key Features

- ◆ **Multi-factor Authentication:** The solution provides a flexibility of combining more than twenty authentication methods to create authentication chains. You can assign these chains to different events to use the specific authentication chains for different kinds of endpoints.
- ◆ **Supports Multiple Repositories:** Advanced Authentication supports Active Directory, Active Directory Lightweight Domain Services, NetIQ eDirectory, and other RFC 2037 and RFC 2037 bis compliant LDAP repositories.

- ♦ **Supports Distributed Environments:** Advanced Authentication works on geographically distributed environments containing high loads.
- ♦ **Multitenancy:** A single Advanced Authentication solution can support multiple tenants to serve multiple customers with different environments.
- ♦ **Supports Multiple Platforms:** Advanced Authentication works on various platforms such as Windows, Linux, and Mac OS.
- ♦ **Helpdesk:** Advanced Authentication provides a separate role of Helpdesk or Security officer. A user with Helpdesk or Security Officer role can manage authenticators for the end users through the Helpdesk portal.
- ♦ **Supports the RADIUS Server:** Advanced Authentication Server contains a built-in RADIUS server to provide strong authentication for third-party RADIUS clients. Also, it can act as a RADIUS client for the third-party RADIUS servers.
- ♦ **Supports ADFS 3, OAuth 2.0, and SAML 2.0:** Advanced Authentication integrates with Active Directory Federation Services, OAuth 2.0, and SAML 2.0. This enables you to perform strong authentication for the users who need to access the third-party consumer applications.
- ♦ **Reporting:** Advanced Authentication provides the Reporting portal that enables you to access different security reports. You can also create customized reports based on your requirement.
- ♦ **Syslog support:** Advanced Authentication provides the central logging server that can be used for log forwarding. You can configure the solution to forward logs to an external Syslog server.
- ♦ **FIPS 140-2 Compliant Encryption:** Advanced Authentication adheres to Federal Information Processing Standard (FIPS) 140-2. You can enable FIPS 140-2 compliant encryption for new installations.
- ♦ **Supports Localization:** Advanced Authentication supports several languages such as Arabic, Chinese, Dutch, and Danish.

Advanced Authentication Server Components

Advanced Authentication server comprises of the following components:

- ♦ **Administration Portal**
For more information, see [“Administration Portal” on page 12](#)
- ♦ **Self-Service Portal**
For more information, see [“Self-Service Portal” on page 13](#)
- ♦ **Helpdesk Portal**
For more information, see [“Helpdesk Portal” on page 13](#)
- ♦ **Reporting Portal**
For more information, see [“Reporting Portal” on page 13](#)

Administration Portal

Administration Portal is a centralized portal that helps you to configure and manage various authentication settings such as methods, events, and so on. You can also configure various policies that are required for authentication. You can perform the following tasks:

- ♦ **Add repositories:** A repository is a database that stores users information. For example: An organization, Digital Airlines contains an Active Directory that stores all of the user’s information such as username, telephone, address, and so on. Administrator can add this Active Directory to

Advanced Authentication solution to help different departments in the organization such as the IT, finance, HR, and Engineering departments to authenticate based on their requirements. For more information about how to add repositories, see [“Adding a Repository”](#).

- ♦ **Configure methods:** A method or an authenticator helps to confirm the identification of a user (or in some cases, a machine) that is trying to log on or access resources. You can configure the required settings for the appropriate methods depending on the requirement by each department. For more information about how to configure methods, see [“Configuring Methods”](#).
- ♦ **Create chains:** A chain is a combination of methods. Users must authenticate with all the methods in a chain. For example, a chain with Fingerprint and Card method can be applicable for the IT department and a chain with Smartphone, LDAP Password, and HOTP is applicable for the Engineering department. For more information about how to create chains, see [“Creating a Chain”](#).
- ♦ **Configure events:** An event is triggered by an external device or application that needs to perform authentication such as a Windows machine, a RADIUS client, a third party client and so on. After creating the chain, Administrator maps the chain to an appropriate event. For more information about how to configure events, see [“Configuring Events”](#).
- ♦ **Map endpoints:** An endpoint is a device on which you can authenticate. Endpoints can be computers, Laptops, tablets, and so on. For more information about how to configure endpoints, see [“Managing Endpoints”](#).
- ♦ **Configure policies:** An administrator can manage policies that are specific to users, devices, or locations to control a user’s authentication. In Advanced Authentication, you can manage the policies in a centralized policy editor. For more information about how to configure policies, see [“Configuring Policies”](#).

Self-Service Portal

The Self-Service Portal allows users to manage the available authentication methods. This portal consists of **Enrolled authenticators** and **Add authenticator**. The **Enrolled authenticators** section displays all the methods that users have enrolled. The **Add authenticator** section displays additional methods available for enrollment. You must configure and enable the [Authenticators Management](#) event to enable users to access the Self-Service portal. For more information on Self-Service portal, see [Advanced Authentication- User](#) guide.

Helpdesk Portal

The Helpdesk Portal allows the helpdesk administrators to enroll and manage the authentication methods for users. Helpdesk administrators can also link authenticators of a user to help authenticate to another user’s account. For more information on Helpdesk portal, see the [Advanced Authentication- Helpdesk Administrator](#) guide.

Reporting Portal

The Reporting Portal allows you to create or customize security reports that provide information about user authentication. It also helps you understand the processor and memory loads. For more information on Reporting portal, see [“Reporting”](#).

Architecture

Advanced Authentication architecture is based on the following three levels of architecture:

- ◆ Basic Architecture
For more information, see [“Basic Architecture” on page 14](#)
- ◆ Enterprise Level Architecture
For more information, see [“Enterprise Level Architecture” on page 15](#)
- ◆ Enterprise Architecture With A Load Balancer
For more information, see [“Enterprise Architecture With A Load Balancer” on page 17](#)

Basic Architecture

The basic architecture of Advanced Authentication is a simple configuration that requires only one Advanced Authentication server.



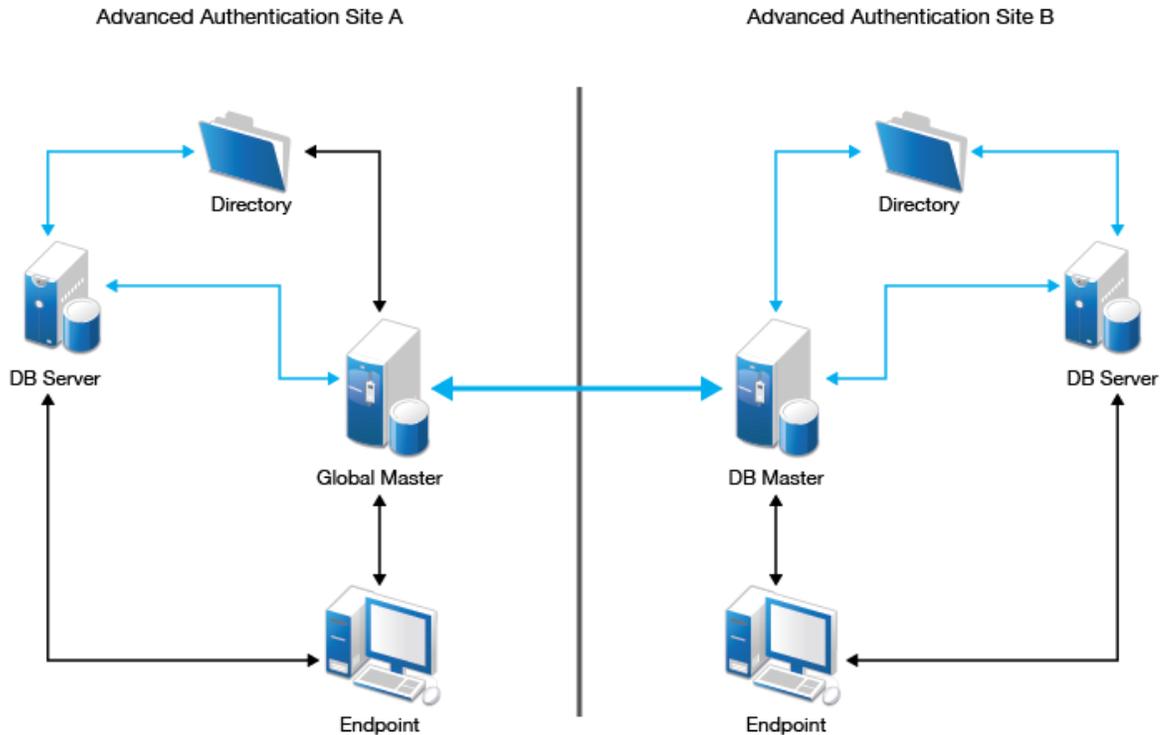
An Advanced Authentication server is connected to a directory such as Active Directory Domain Services, NetIQ eDirectory, Active Directory Lightweight Directory Service or other compliant LDAP directories. An Event Endpoint can be Windows, Linux or Mac OS X machine, NetIQ Access Manager, NetIQ CloudAccess, or RADIUS Client to authenticate through the RADIUS Server that is built-in the Advanced Authentication Server. For a complete list of supported events, see [Configuring Events](#).

Enterprise Level Architecture

In the enterprise level architecture of Advanced Authentication, you can create several sites for different geographical locations.

For example, the [Figure 1-1 on page 15](#) displays two Advanced Authentication sites, **Site A** and **Site B**.

Figure 1-1 Enterprise Level Architecture



- ♦ **Site A:** The first site that is created for headquarters in New York. The first Advanced Authentication server of site A contains the **Global Master** and **Registrar** roles. This server contains a master database and it can be used to register new sites and servers.
- ♦ **Site B:** Another site created for the office in London. The structure of site B is similar to site A. The Global Master in another site has the DB Master role. DB servers interact with the DB Master.

DB Server provides a database that is used for backup and fail-over. You can create a maximum of two DB servers per site. When the Global Master is unavailable, the DB server responds to the database requests. When the Global Master becomes available again, the DB server synchronizes with the Global Master and the Global Master becomes the primary point of contact for database requests again.

Endpoints interact with Global Master or DB Master servers. When these servers are not available, they interact with DB servers.

NOTE: DB servers connect to each other directly. If the Global Master is down, the DB servers will replicate.

A Global Master must have a connection to each of the LDAP servers. Hence in a data center with Global Master, you must have LDAP servers for all the used domains.

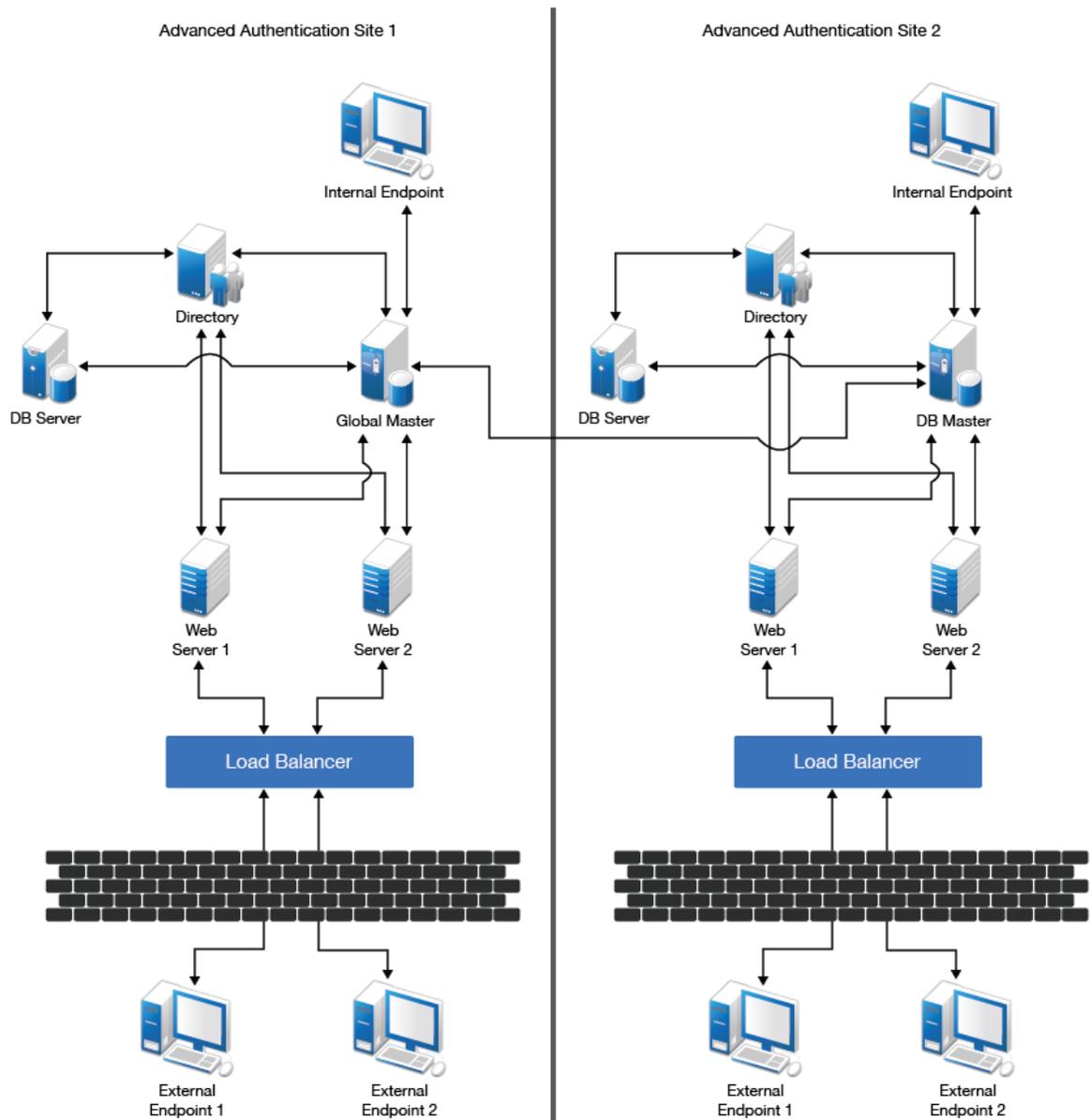
IMPORTANT: Ensure to take regular snapshots or to clone the primary site to protect from any hardware issues or any other accidental failures. It is recommended to do it each time after you change the configuration of repositories, methods, chains, events and policies, or add or remove servers in the cluster.

You can convert DB server of primary site to Global Master. This requires corresponding DNS changes. Nothing can be done if Global Master and all slaves are lost.

Enterprise Architecture With A Load Balancer

The enterprise architecture with a load balancer contains web servers and load balancers along with the components in [Enterprise Level Architecture](#). [Figure 1-2 on page 17](#) illustrates the Enterprise architecture with a load balancer.

Figure 1-2 Enterprise Architecture with Load Balancer



- ♦ **Web Servers:** Web server does not contain a database. It responds to the authentication requests and connects to Global Master. . It is not recommended to deploy more than 5-6 web servers per site.
- ♦ **Load Balancer:** A load balancer provides an ability to serve authentication requests from **External Endpoints**. A load balancer is a third-party component. It must be configured to interact with Web servers.

Terminologies

- ◆ [“Authentication Method” on page 18](#)
- ◆ [“Authentication Chain” on page 18](#)
- ◆ [“Authentication Event” on page 18](#)
- ◆ [“Endpoint” on page 18](#)

Authentication Method

An authentication method verifies the identity of an individual who wants to access data, resources, or applications. Validating that identity establishes a trust relationship for further interactions.

Authentication Chain

An authentication chain is a combination of authentication methods. A user must pass all methods in the chain to be successfully authenticated. For example, if you create a chain with LDAP Password and SMS, a user must first specify the LDAP Password. If the password is correct, the system sends an SMS with a One-Time-Password (OTP) to the user’s mobile. The user must specify the correct OTP to be authenticated.

You can create chains with multiple methods that are applicable for highly secure environments. You can create authentication chains for specific group of users in the repositories.

Authentication Event

An authentication event is triggered by an external device or application that needs to perform authentication. It can be triggered by a RADIUS Client (Citrix Netscaler, Cisco VPN, Juniper VPN and so on) or an API request. Each event can be configured with one or more authentication chains that enables a user to authenticate.

Endpoint

An endpoint is a device on which you can authenticate. Endpoints can be computers, Laptops, tablets, Smartphones, and so on.

Configuring Advanced Authentication

Advanced Authentication Server Appliance is intended for processing requests for authentication coming from the Advanced Authentication system users.

This chapter contains the following sections:

- ♦ [Chapter 2, “Logging In to the Advanced Authentication Administrative Portal,” on page 21](#)
- ♦ [Chapter 3, “Configuring Advanced Authentication Server Appliance,” on page 23](#)
- ♦ [Chapter 4, “Enrolling the Authentication Methods,” on page 81](#)

2 Logging In to the Advanced Authentication Administrative Portal

After you set up an applicable server mode, the Advanced Authentication Administrative portal is displayed.

To log in to the Advanced Authentication Administrative portal, perform the following steps:

- 1 Specify the administrator's credentials in the format: `repository\user` (**localadmin** by default).
- 2 Click **Next**.
- 3 The **Admin Password** chain is selected by default as the only available chain. Specify the password that you specified while setting up the DB Master server mode.
- 4 Click **Next**.

The Dashboard page is displayed.

- 5 You can change the language from the list on the upper-right corner of the Administration portal.
The languages supported are: Arabic, Chinese Simplified, Chinese Traditional, Danish, Dutch, English, French, German, Italian, Japanese, Polish, Portuguese (Brazilian), Russian, Spanish, Hebrew, and Swedish.

IMPORTANT: Password of **localadmin** account expires by default. For uninterrupted access to the Administration portal, it is strongly recommended to add authorized users or group of users from a configured repository to the **FULL ADMINS** role. Then you must assign chains, which contain methods that are enrolled for users, to the **AdminUI** event (at a minimum with an LDAP Password).

NOTE: It is not recommended to access the Advanced Authentication Administration portal through a load balancer, as the replicated data may not be displayed.

3 Configuring Advanced Authentication Server Appliance

In the Administration portal, you can configure and manage various authentication settings such as methods, events, and so on. You can also configure various policies that are required for authentication.

Advanced Authentication Administration portal contains the Help  option that guides you on how to configure all settings for your authentication framework. The Help section provides you with information on the specific section you are working on.

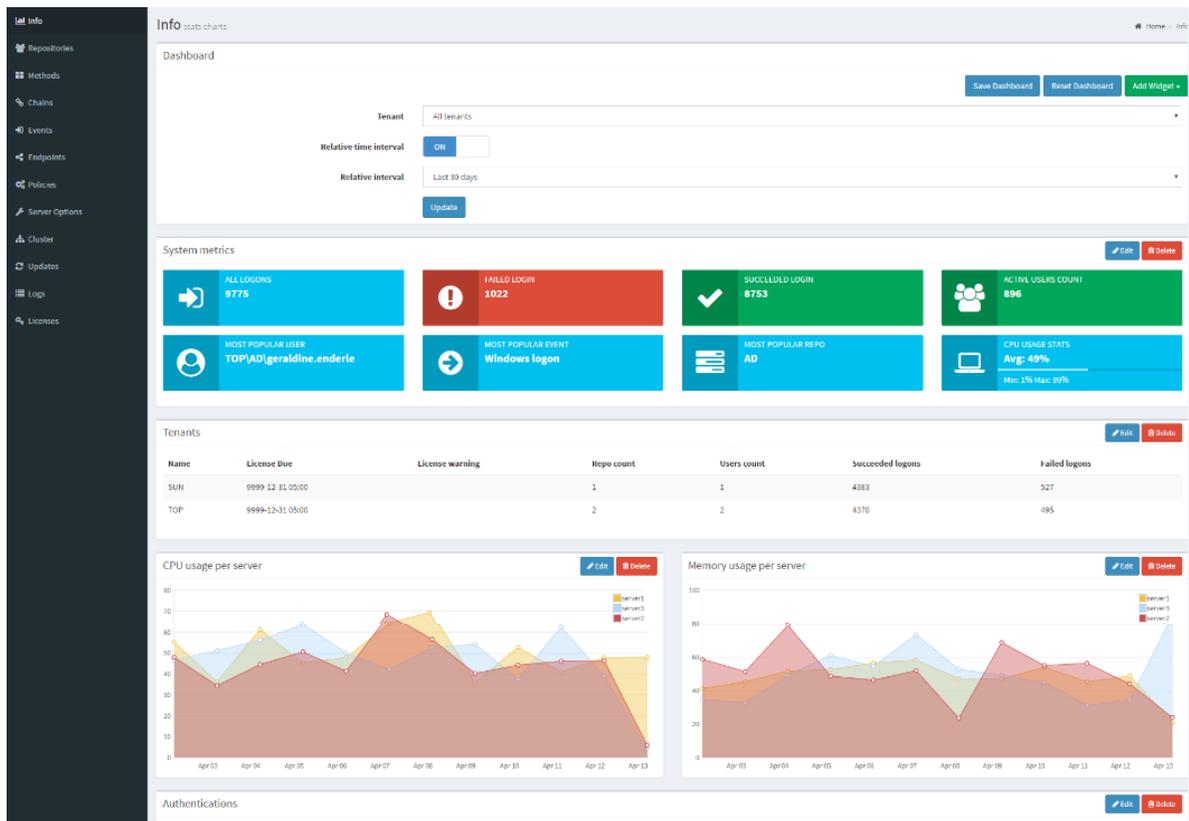
This chapter contains the following sections:

- ◆ [“Managing Dashboard” on page 23](#)
- ◆ [“Adding a Repository” on page 29](#)
- ◆ [“Configuring Methods” on page 38](#)
- ◆ [“Creating a Chain” on page 58](#)
- ◆ [“Configuring Events” on page 59](#)
- ◆ [“Managing Endpoints” on page 66](#)
- ◆ [“Configuring Policies” on page 68](#)
- ◆ [“Adding a License” on page 79](#)
- ◆ [“Exporting the Database” on page 79](#)

Managing Dashboard

After you login into the Advanced Authentication Administration console, the Dashboard is displayed. Dashboard contains widgets that you can add or customize to view a graphical representation of data. The information in the Dashboard helps administrators to track memory utilization, tenant information, successful or failed logins, and so forth.

You can view the Dashboard for all the tenants or specific tenants.



You can perform the following to manage the Dashboard:

- ◆ Add widgets
- ◆ Customize Dashboard
- ◆ Update Dashboard
- ◆ Customize the Default Widgets

Adding Widgets

To add widgets, perform the following steps:

- 1 Click **Add widget** in the top-right corner of the **Dashboard** screen.
- 2 Select the widget from the list that you want to add to the dashboard.
- 3 Specify the appropriate details for the widget in the **Add Widget** screen.

Customizing Dashboard

You can customize the Dashboard by moving the widgets or deleting the unused widgets.

To move the widgets, click on the widget and the drag icon  appears. You can then drag and drop the widget to the desired location of the Dashboard.

To delete unused widgets, click **Delete** on the top of each widget.

After customizing the dashboard, click **Save Dashboard** on the upper-right corner of the **Dashboard** screen.

Updating Dashboard to View Real Time or Historical Data

You can update Dashboard to view the data based on the time interval or historical data.

Viewing Dashboard based on Time Interval

To view records based on real time interval, enable **Relative time interval** to **ON** and select the **Relative interval**. The default time interval for Dashboard is 15 minutes.

Viewing Dashboard for Previous Records

To view previous records, enable **Relative time interval** to **OFF** and select the **Date range**. Click **Update**

Customizing the Default Widgets

Following are the default widgets when you login. You can edit these widgets according to your need:

- ◆ “System Metrics” on page 25
- ◆ “Tenants” on page 26
- ◆ “CPU Usage Per Server and Memory Usage Per Server” on page 26
- ◆ “Servers” on page 26
- ◆ “Authentications” on page 26
- ◆ “Logons Per Result” on page 26
- ◆ “Activity Stream” on page 26
- ◆ “Successful/Failed Logons” on page 27
- ◆ “Top Events With Successful Logon Per Chain” on page 27
- ◆ “Top Events With Failed Logon Per Method” on page 27
- ◆ “Top 10 Events” on page 27
- ◆ “Top 10 chains With Successful Result” on page 27
- ◆ “Top 10 Servers” on page 28
- ◆ “Top 10 Tenants” on page 28
- ◆ “Top 10 Repo” on page 28
- ◆ “Top 5 Events for Logons” on page 28
- ◆ “Top 5 Users for Logons” on page 28
- ◆ “Top 10 Users With Failed Logon” on page 28
- ◆ “Top 10 Users” on page 29
- ◆ “Top 10 Events” on page 29
- ◆ “Top 10 Methods With Failed Result” on page 29
- ◆ “Disk Usage Per Server” on page 29

System Metrics

This widget displays statistics about user’s login, popularity and so on. The following section defines each system metric:

- ◆ **All Logons**: Total number of logins.

- ♦ **Failed Login:** Total number of failed logins by the users.
- ♦ **Succeeded Login:** Total number of successful logins by the users.
- ♦ **Active Users Count:** The number of active users.
- ♦ **Most Popular User:** The user that has used the console most.
- ♦ **Most Popular Event:** The event that users have used the most.
- ♦ **Most Popular Repo:** The repository that users have used the most.
- ♦ **CPU Usage Stats:** The average percentage of CPU usage.

Tenants

This widget displays information about the tenants and their login.

CPU Usage Per Server and Memory Usage Per Server

These widgets display information about percentage of CPU and memory usage per time interval. They show average CPU and memory usage.

You can edit the title and change the time interval to view the records.

Servers

This widget displays the CPU, memory, and disk usage.

Authentications

This widget displays the total logon count for time interval. You can edit the display based on the events: **All logon events**, **Failed logon events**, or **successful logon events**.

Logons Per Result

This widget displays two lines: one for successful logons and one for failed logons.

To edit the widget, click **Edit** and select the appropriate fields. You can also select the number of records from **Size** and sort in the ascending (previous to latest) or descending (latest to previous) order.

Activity Stream

This widget displays information about user, tenant, chain, and method used for authentication, and the result.

You can edit the display based on the events: **All logon events**, **Failed logon events**, or **successful logon events**.

Successful/Failed Logons

This widget displays information about the successful or failed logins by users. You can edit the widget to customize the display based on the event type and parameters such as Tenant name, event name and so on.

To edit the widget, click **Edit** and select the appropriate fields. You can also select the number of records from **Size** and sort in the ascending (previous to latest) or descending (latest to previous) order.

Top Events With Successful Logon Per Chain

This widget displays the top events based on the successful logon for each chain. You can edit the widget to customize the display based on the event type and parameters such as Tenant name, event name and so on.

To edit the widget, click **Edit** and select the appropriate fields. You can also select the number of records from **Size** and sort in the ascending (previous to latest) or descending (latest to previous) order.

Top Events With Failed Logon Per Method

This widget displays the top events based on the failed logon for each chain. You can edit the widget to customize the display based on the event type and parameters such as Tenant name, event name and so on.

To edit the widget, click **Edit** and select the appropriate fields. You can also select the number of records from **Size** and sort in the ascending (previous to latest) or descending (latest to previous) order.

Top 10 Events

This widget displays the top ten events the user has performed. You can edit the widget to customize the display based on the event type.

To edit the widget, click **Edit** and select the appropriate fields. You can also select the number of records from **Size** and sort in the ascending (previous to latest) or descending (latest to previous) order.

Top 10 chains With Successful Result

This widget displays the top ten chains the user has successfully authenticated with. You can edit the widget to customize the display based on the event type.

To edit the widget, click **Edit** and select the appropriate fields. You can also select the number of records from **Size** and sort in the ascending (previous to latest) or descending (latest to previous) order.

Top 10 Servers

This widget displays the top ten servers the user has used to authenticate. You can edit the widget to customize the display based on the event type.

To edit the widget, click **Edit** and select the appropriate fields. You can also select the number of records from **Size** and sort in the ascending (previous to latest) or descending (latest to previous) order.

Top 10 Tenants

This widget displays the top ten tenants. You can edit the widget to customize the display based on the event type.

To edit the widget, click **Edit** and select the appropriate fields. You can also select the number of records from **Size** and sort in the ascending (previous to latest) or descending (latest to previous) order.

Top 10 Repo

This widget displays the top ten repositories. You can edit the widget to customize the display based on the event type.

To edit the widget, click **Edit** and select the appropriate fields. You can also select the number of records from **Size** and sort in the ascending (previous to latest) or descending (latest to previous) order.

Top 5 Events for Logons

This widget displays the top five events for login. You can edit the widget to customize the display based on the event type.

To edit the widget, click **Edit** and select the appropriate fields. You can also select the number of records from **Size** and sort in the ascending (previous to latest) or descending (latest to previous) order.

Top 5 Users for Logons

This widget displays the top five users for login. You can edit the widget to customize the display based on the event type.

To edit the widget, click **Edit** and select the appropriate fields. You can also select the number of records from **Size** and sort in the ascending (previous to latest) or descending (latest to previous) order.

Top 10 Users With Failed Logon

This widget displays the top ten users who have failed in the login attempt. You can edit the widget to customize the display based on the event type.

To edit the widget, click **Edit** and select the appropriate fields. You can also select the number of records from **Size** and sort in the ascending (previous to latest) or descending (latest to previous) order.

Top 10 Users

This widget displays the top ten users. You can edit the widget to customize the display based on the event type.

To edit the widget, click **Edit** and select the appropriate fields. You can also select the number of records from **Size** and sort in the ascending (previous to latest) or descending (latest to previous) order.

Top 10 Events

This widget displays the top ten events. You can edit the widget to customize the display based on the event type.

To edit the widget, click **Edit** and select the appropriate fields. You can also select the number of records from **Size** and sort in the ascending (previous to latest) or descending (latest to previous) order.

Top 10 Methods With Failed Result

This widget displays the top ten methods with failed authentication results. You can edit the widget to customize the display based on the event type.

To edit the widget, click **Edit** and select the appropriate fields. You can also select the number of records from **Size** and sort in the ascending (previous to latest) or descending (latest to previous) order.

Disk Usage Per Server

This widget displays information about the percentage of disk usage per time interval. It shows the average disk usage.

You can edit the title and change the time interval to view the records.

Adding a Repository

A repository is a central location where the user's data is stored. Advanced Authentication uses the repository only to retrieve the user information and configurations in Advanced Authentication do not affect the repository. The authentication templates are stored inside the appliance and are fully encrypted.

Advanced Authentication supports any LDAP compliant directory such as Active Directory Domain Services, NetIQ eDirectory, Active Directory Lightweight Directory Services, OpenLDAP, and OpenDJ.

When you add a new repository, you can match the users in the repository to the authentication chains. You require only the read permission to access a repository.

To add a repository, perform the following steps:

- 1 Click **Repositories > Add**.
- 2 Select an applicable repository type from the **LDAP type** list. The options are:
 - ♦ **AD** for Active Directory Domain Services
 - ♦ **AD LDS** for Active Directory Lightweight Domain Services

- ◆ **eDirectory** for NetIQ eDirectory
- ◆ **Other** for OpenLDAP, OpenDJ and other types

For **AD**, a repository name is automatically set to the NetBIOS name of the domain. For other LDAP repository types, you need to specify the name in **Name**.

- 3 Specify a container for the users in **Base DN**. When you select the **Subtree** option, Advanced Authentication performs a search for the users in all the child nodes. You can change the search scope by selecting the **Search one level only** option.
- 4 Specify a user account in **User** and specify the password of the user in **Password**. Ensure that the user's password has no expiry.
- 5 You can specify a container for the groups in **Group DN (optional)**. When you select the **Subtree** option, Advanced Authentication performs a search for the groups in all the child nodes. You can change the search scope by selecting the **Search one level only** option.
- 6 If you have selected **AD** as the **LDAP type**, select **DNS discovery** to find LDAP servers automatically. Specify the **DNS zone** and **Site name (optional)** and click **Perform DNS Discovery**. If you want to add LDAP servers manually, select **Manual setting**.

NOTE: If you specify an RODC (Read Only Domain Controller) in the LDAP server, the server uses this DC for read requests (get groups, get user info) and for logon requests (LDAP Password method and bind requests for Advanced Authentication LDAP user). These requests are redirected to a writable DC because RODC is installed in untrusted locations and does not have copies of the user's passwords. Therefore, if a writable DC is not available, Advanced Authentication will not be able to bind to the LDAP repository.

To solve this issue, you must enable the password replication of a user account specified in [Step 4](#). To do this, you must add the account to the **Allowed RODC Password Replication Group**.

However, even when you enable such replication, users cannot use the LDAP Password method because user's passwords are not replicated. It is recommended not to replicate passwords of all the users. For more information, see the article [Understanding "Read Only Domain Controller" authentication](#).

-
- 7 Click **Add server**. You can add the different servers in your network. The list is used as a pool of servers. Each time the connection is open, a random server is selected in the pool and unavailable servers are discarded.
 - 8 Specify an LDAP server's **Address** and **Port**.
 - 9 Turn **SSL** to **ON** to use the SSL technology (if applicable).
 - 10 Click **Save**, next to server's credentials.
 - 11 Add additional servers (if applicable).
 - 12 (Conditional) To configure custom attributes, expand **Advanced Settings**. The Advanced Settings are required for OpenDJ, OpenLDAP, and in some cases for NetIQ eDirectory.
 - 13 Click **Save**.

NOTE: If you use NetIQ eDirectory with the option **Require TLS for Simple Bind with Password** enabled, you may get the error: `Can't bind to LDAP: confidentialityRequired`. To fix the error, you must either disable the option or do the following:

1. Click **LDAP > LDAP Options > Connections** in the NetIQ eDirectory Administration portal.
2. Set **Client Certificate** to **Not Requested**.

3. Set a correct port number and select **SSL** in the Repository settings.
4. Click **Sync now** with the added repository.

-
- 14 You can change the search scope and the **Group DN (optional)** functionality. In Advanced Authentication 5.2, you had to specify a common **Base DN** for users and groups.
 - 15 To verify the synchronization of a repository, click **Edit** and you can view the information in **Last sync**.
 - 16 Click **Full sync** to perform a complete synchronization of the repository.
Advanced Authentication performs an automatic synchronization of modified objects (fastsync) on an hourly basis for AD. The complete synchronization (**Full sync**) is performed on a weekly basis.

NOTE: If an LDAP server is unavailable for 2.5 seconds, Advanced Authentication excludes it from the LDAP requests for a period of 3 minutes.

Advanced Settings

Advanced Settings allow you to customize attributes that Advanced Authentication reads from a repository. Click **+** to expand the **Advanced Settings**. The following list describes the different attributes in Advanced Settings:

- ◆ [“User Lookup Attributes” on page 31](#)
- ◆ [“User Name Attributes” on page 31](#)
- ◆ [“User Mail Attributes” on page 32](#)
- ◆ [“User Cell Phone Attributes” on page 32](#)
- ◆ [“Group Lookup Attributes” on page 32](#)
- ◆ [“Group Name Attributes” on page 32](#)
- ◆ [“Verify SSL Certificate” on page 33](#)
- ◆ [“Enable Paged Search” on page 33](#)
- ◆ [“Enable Nested Groups Support” on page 33](#)
- ◆ [“Framed IPv4 Address Attribute” on page 34](#)

User Lookup Attributes

Advanced Authentication validates the specified attributes for an entered user name.

For Active Directory (AD), the default attributes are `sAMAccountName` and `userPrincipalName`. For other repositories, `cn` is the default attribute.

User Name Attributes

Advanced Authentication shows a name from the first, non-empty specified field for an entered user name.

For AD, the default attributes are `sAMAccountName` and `userPrincipalName`. For other repositories, `cn` is the default attribute.

User Mail Attributes

Advanced Authentication validates the specified attributes to retrieve a user's email address.

Default attributes are `mail` and `otherMailbox`.

User Cell Phone Attributes

Advanced Authentication validates the specified attributes to retrieve a user's phone number. These attributes are used for methods such as SMS OTP, Voice, and Voice OTP. Previously, the first attribute of **User cell phone attributes** was used as a default attribute for authenticating with [SMS OTP](#), [Voice](#), and [Voice OTP](#) methods. Now, users can use different phone numbers for these methods. For example, Bob wants to authenticate with SMS OTP, Voice, and Voice OTP methods. He has a cell phone number, a home phone number, and an IP phone number and wants to use these numbers for each of these methods. He can define these phone numbers in the respective settings of these methods.

Default attributes: `mobile`, `otherMobile`.

NOTE: If you have multiple repositories, you must use the same configuration of **User cell phone attributes** for all the repositories.

Group Lookup Attributes

Advanced Authentication validates the specified attributes for an entered group name.

For Active Directory, the default attribute is `sAMAccountName`. For other repositories, `cn` is the default attribute.

Group Name Attributes

Advanced Authentication shows a name from the first, non-empty specified field for an entered group name.

For Active Directory, the default attribute is `sAMAccountName`. For other repositories, `cn` is the default attribute.

Advanced Authentication supports the RFC 2037 and RFC 2037 bis. RFC 2037 determines a standard LDAP schema and contains a `memberUid` attribute (POSIX style). RFC 2037 bis determines an updated LDAP schema and contains a `member` attribute. Active Directory, LDS, and eDir support RFC 2037 bis. OpenLDAP contains `posixAccount` and `posixGroup` that follows RFC 2037.

Advanced Authentication supports the following attributes for the Group Name attributes:

Attribute	Default Value	Value for the Repository
User Object Class	<code>user</code>	OpenDJ and OpenLDAP: <code>person</code>
Group Object Class	<code>group</code>	OpenDJ: <code>groupOfNames</code> OpenLDAP: <code>posixGroup</code>

Attribute	Default Value	Value for the Repository
Group Member Attribute	member	OpenDJ: member OpenLDAP: memberUId. If a required group contains <code>groupOfNames</code> class, disable POSIX style groups . If the group contains <code>posixGroup</code> , enable POSIX style groups . <ul style="list-style-type: none"> ◆ User UID attribute This attribute is available only when POSIX style groups is ON . Default value: <code>uid</code> .
Object ID Attribute	entryUUID	
This attribute is available only for other LDAP type only.		

NOTE: For information about the Logon filter settings (Legacy logon tag and MFA logon tag), see [Configuring Logon Filter](#).

Verify SSL Certificate

Enable **Verify SSL Certificate** to ensure that the LDAP connection to appliance is secured with a valid self-signed SSL certificate. This helps to prevent any attacks on the LDAP connection and ensures safe authentication. Click **Choose File** to browse the self-signed certificate.

Enable Paged Search

The **Enable paged search** option allows LDAP repositories to support paged search in which the repositories can retrieve a result of a query set in small portions. By default, this option is set to **ON**. For openLDAP (with file-based backend), the option must be set to **OFF**.

NOTE: You must not disable the option for Active Directory repositories. It can also affect the performance on other supported repositories such as NetIQ eDirectory.

Enable Nested Groups Support

This option allows you to enable or disable nested groups support. By default, the **Enable nested groups support** option is set to **ON**.

If **Enable nested groups support** option is set to **ON**, then Advanced Authentication will authenticate all the users of the group and its nested groups assigned to a chain. If **Enable nested groups support** option is set to **OFF**, then Advanced Authentication will authenticate only the members of the group assigned to the chain. The members of the nested groups cannot access the chain.

Consider there is a group by name **All Users** assigned to **SMS Authentication** chain and the **All Users** group has subgroups **Contractors** and **Suppliers**. When **Enable nested groups support** option is set to **ON**, then Advanced Authentication will authenticate **All Users** group and the nested groups **Contractors** and **Suppliers** for **SMS Authentication** chain. When the option is set to **OFF**,

then Advanced Authentication will authenticate only the members of **All Users** group and the nested group members will not have access to **SMS Authentication** chain. This improves the login performance of the appliance.

Framed IPv4 Address Attribute

This attribute is applicable for the Radius Server event.

For Active Directory, when the **Framed IPv4 Address** is blank, the Advanced Authentication RADIUS server returns value of the `msRADIUSFramedIPAddress` attribute as `Framed-IP-Address` after you log in with the RADIUS event. When you specify any other attribute in **Framed IPv4 Address attribute**, then the value of the specified attribute is returned as the `Framed-IP-Address` instead of the `msRADIUSFramedIPAddress` attribute value. You can configure the `Framed-IP-Address` in **Active Directory Users and Computers > Dial-in > Assign Static IP Addresses** and click **Static IP Addresses**. It supports only IPv4.

For the other repositories, when the **Framed IPv4 Address** is blank, the Advanced Authentication RADIUS server returns value of the `radiusFramedIPAddress` attribute as `Framed-IP-Address` after you log in with the RADIUS event. When you specify any other attribute in **Framed IPv4 Address attribute**, then the value of the specified attribute is returned as the `Framed-IP-Address` instead of the `radiusFramedIPAddress` attribute value.

Used Attributes

The following table describes the attributes that the appliance uses in the supported directories.

Attribute Name	LDAP Name	Description	Type	Supported in Active Directory	Supported in LDS	Supported in eDirectory
CN (Common Name)	CN	An identifier of an object	String	?	?	?
Mobile	Mobile	A phone number of an object's cellular or mobile phone	Phone number	?	?	?
Email Address	mail	An email address of a user	Email address	?	?	?
User-Principal-Name (UPN)	userPrincipalName	An Internet based format login name for a user	String	?	?	?
SAM-Account-Name	sAMAccountName	The login name used to support clients and servers running earlier versions of operating systems such as Windows NT 4.0	String	?	×	×
GUID	GUID	An assured unique value for any object	Octet String	×	×	?
Object Class	Object Class	An unordered list of object classes	String	?	?	?

Attribute Name	LDAP Name	Description	Type	Supported in Active Directory	Supported in LDS	Supported in eDirectory
Member	Member	A list that indicates the objects associated with a group or list	String	?	?	?
User-Account-Control	userAccountControl	Flags that control the behavior of a user account	Enumeration	?	×	×
ms-DS-User-Account-Control-Computed	msDS-User-Account-Control-Computed	Flags that are similar to userAccountControl, but the attribute's value can contain additional bits that are not persisted	Enumeration	?	?	×
Primary-Group-ID	primaryGroupID	A relative identifier (RID) for the primary group of a user	Enumeration	?	×	×
Object-Guid	objectGUID	A unique identifier for an object	Octet String	?	?	×
object-Sid	objectSid	A Binary value that specifies the security identifier (SID) of the user	Octet String	?	?	×
Logon-Hours	logonHours	Hours that the user is allowed to logon to the domain	Octet String	?	×	×
USN-Changed	uSNChanged	An update sequence number (USN) assigned by the local directory for the latest change including creation	Interval	?	?	×

NOTE: The `sAMAccountName` and `userPrincipalName` attributes are supported only for AD DS repository. The Active Directory LDS and eDirectory repositories do not support the attributes.

LDAP Queries for Repository Sync

Active Directory DS and AD LDS Queries

1. Search users

```
(&(usnChanged>=217368)(&(objectClass=user)(|(cn=*)(sAMAccountName=*)(userPrincipalName=*)))))
```

Requested attributes:

```
['objectSID', 'sAMAccountName', 'objectClass', 'logonHours', 'primaryGroupId', 'otherMobile', 'mobile', 'userAccountControl', 'cn', 'usnChanged', 'userPrincipalName', 'msDS-User-Account-Control-Computed', 'objectGUID', 'mail', 'otherMailbox', 'GUID']
```

2. Search groups

```
(&(usnChanged>=217368)(&(objectClass=group)(|(cn=*)(sAMAccountName=*)))))
```

Requested attributes:

```
['objectSID', 'sAMAccountName', 'objectClass', 'logonHours', 'primaryGroupId',  
'userAccountControl', 'cn', 'usnChanged', 'msDS-User-Account-Control-Computed',  
'objectGUID', 'GUID']
```

eDirectory Queries

The queries are the same as for Active Directory DS and Active Directory LDS, except for 'usnChanged' (this filter is not used).

1. Search users

```
(&(objectClass=user)(|(cn=*)(sAMAccountName=*)(userPrincipalName=*))
```

Requested attributes:

```
['objectSID', 'sAMAccountName', 'objectClass', 'logonHours', 'primaryGroupId',  
'otherMobile', 'mobile', 'userAccountControl', 'cn', 'userPrincipalName', 'msDS-  
User-Account-Control-Computed', 'objectGUID', 'mail', 'otherMailbox', 'GUID']
```

2. Search groups

```
(&(objectClass=group)(|(cn=*)(sAMAccountName=*))
```

Requested attributes:

```
['objectSID', 'sAMAccountName', 'objectClass', 'logonHours', 'primaryGroupId',  
'userAccountControl', 'cn', 'msDS-User-Account-Control-Computed', 'objectGUID',  
'GUID']
```

LDAP Queries During Logon

For Active Directory LDS queries, the attributes are same as Active Directory DS except for the objectSid (the filter is not used in queries on membership in groups).

In the examples below, the username is pjones, base_dn is DC=company,DC=com

Active Directory DS and Active Directory LDS queries

1. Basic user information

```
(&(objectClass=user)(|(cn=pjones)(sAMAccountName=pjones)(userPrincipalName=pjones)  
)
```

Requested attributes:

```
(&(objectClass=user)(objectGUID=\0f\d1\14\49\bc\cc\04\44\b7\b7\b7\19\06\15\c6\82\55))
```

Requested attributes:

```
['otherMobile', 'GUID', 'userAccountControl', 'msDS-User-Account-Control-  
Computed', 'mobile', 'primaryGroupId', 'cn', 'objectGUID', 'userPrincipalName',  
'objectSID', 'mail', 'sAMAccountName', 'objectClass', 'logonHours',  
'otherMailbox']
```

2. Group membership information for user

Active Directory specific query using objectSid filter:

```
(|(member=CN=pjones,CN=Users,DC=company,DC=com)(objectSid=S-1-5-21-3303523795-  
413055529-2892985274-513))
```

Requested attributes:

```
['GUID', 'userAccountControl', 'msDS-User-Account-Control-Computed',  
'primaryGroupId', 'objectGUID', 'cn', 'objectSID', 'objectClass',  
'sAMAccountName', 'logonHours']
```

3. Iteratively query about each group received from above query

```
(member=CN=Performance Monitor Users,CN=Builtin,DC=company,DC=com)
```

Requested attributes:

```
['GUID', 'userAccountControl', 'msDS-User-Account-Control-Computed',  
'primaryGroupId', 'objectGUID', 'cn', 'objectSID', 'objectClass',  
'sAMAccountName', 'logonHours']
```

eDirectory Queries

Basic user information

```
(&(objectClass=user)(|(cn=pjones)(sAMAccountName=pjones)(userPrincipalName=pjones)  
))
```

Requested attributes:

```
['otherMobile', 'GUID', 'userAccountControl', 'msDS-User-Account-Control-  
Computed', 'mobile', 'primaryGroupId', 'cn', 'objectGUID', 'userPrincipalName',  
'objectSID', 'mail', 'sAMAccountName', 'objectClass', 'logonHours',  
'otherMailbox']
```

```
(&(objectClass=user)(GUID=\57\b6\c2\c1\b9\7f\4b\40\b9\70\5f\9a\1d\76\6c\d2))
```

Requested attributes:

```
['otherMobile', 'GUID', 'userAccountControl', 'msDS-User-Account-Control-  
Computed', 'mobile', 'primaryGroupId', 'cn', 'objectGUID', 'userPrincipalName',  
'objectSID', 'mail', 'sAMAccountName', 'objectClass', 'logonHours',  
'otherMailbox']
```

Group membership information for user

```
(member=cn=pjones,o=AAF)
```

Requested attributes:

```
['GUID', 'userAccountControl', 'msDS-User-Account-Control-Computed',  
'primaryGroupId', 'objectGUID', 'cn', 'objectSID', 'objectClass',  
'sAMAccountName', 'logonHours']
```

Local Repository

The Local repository contains the Advanced Authentication server data. You can manage users and set roles for users in the local repository.

To edit a local repository, perform the following steps:

- 1 Click **Edit** in the **LOCAL** section of **Repositories**.
- 2 In the **Global Roles** tab, you can manage the Helpdesk administrators as **ENROLL ADMINS** and Advanced Authentication administrators as **FULL ADMINS**.

By default, there are no ENROLL ADMINS and the account LOCAL\ADMIN is specified as FULL ADMIN. You can change this by adding the user names from local or the repositories in [Members](#).

- 3 Click **Save**.
- 4 In the **Users** tab, you can manage the local users.
To add the new local account, click **Add** and specify the required information of the user.
- 5 In the **Settings** tab, you can edit the name of the Local repository.

Configuring Methods

A method is a way of authenticating the identity of an individual who attempts to access an endpoint. Advanced Authentication provides several such methods.

To configure an authentication method for Advanced Authentication, perform the following steps:

- 1 Click **Methods**.
- 2 Click the **Edit** icon  next to the authentication method.
- 3 Make the required changes.
- 4 Click **Save**.

After configuring the authentication methods, you must create an authentication chain and map the configured methods to the chain. You can also create a chain with a single method. For example, you can create different authentication chains for an organization that has two departments, IT and Finance. For the IT department, you can create a chain with **Password** and **Smartphone** methods. For the Finance department, a chain with only the **Fingerprint** method can be created. For more information about creating chains, see [“Creating a Chain”](#).

The methods do not appear in the Self-Service portal until you include them in a chain, and link that chain to an event.

You can configure the following methods in Advanced Authentication:

- ◆ [Bluetooth](#)
- ◆ [Card](#)
- ◆ [Email OTP](#)
- ◆ [Emergency Password](#)
- ◆ [Fingerprint](#)
- ◆ [LDAP Password](#)
- ◆ [OATH OTP](#)
- ◆ [Password](#)
- ◆ [PKI](#)
- ◆ [RADIUS Client](#)
- ◆ [Security Questions](#)
- ◆ [Smartphone](#)
- ◆ [SMS OTP](#)
- ◆ [Swisscom Mobile ID](#)
- ◆ [FIDO U2F](#)

- ◆ [Voice](#)
- ◆ [Voice OTP](#)

NOTE: Configurations that have been set by a top administrator for a particular method are grayed out. The configurations are not displayed, if the configurations are hidden by the top administrator.

Bluetooth

In the **Bluetooth** method, you can enroll your smartphone or a mobile device. For example, Bob wants to be authenticated through the Bluetooth method. He enrolls the Bluetooth method on the Advanced Authentication Self-Service portal. He can get authenticated with the Bluetooth method only when his smartphone is in the range.

By default, the **Enable reaction on device removal** option is enabled. When this option is enabled and a user tries to logs in to Windows using Bluetooth, Windows gets locked automatically in the following scenario:

- ◆ When the Bluetooth device is disabled
- ◆ When the Bluetooth device is out of range

NOTE: It is recommended to combine the Bluetooth method with another authentication method in a chain to increase the security.

Card

The **Card** authentication happens when a user places a contactless card on a card reader.

Advanced Authentication supports the Microsoft policy [Interactive logon: Smart card removal behavior](#) that allows you to specify an action on the card event. You can configure the policy to perform a force log off or lock a user session when a user places a card on the reader. Only Microsoft Windows supports this policy.

By default, the **Enable Tap&Go** option is disabled. When this option is disabled, a card must be placed on the reader when a user logs in. When the user removes the card from the reader, the Windows Client runs an action that is specified in the [Interactive logon: Smart card removal behavior policy](#). When you set this option to **ON**, users can tap a card to perform the following actions (depending on the [Interactive logon: Smart card removal behavior policy](#)) without keeping their cards on the reader:

- ◆ To log in
- ◆ To lock a session
- ◆ To log off

NOTE: The policy is supported for Microsoft Windows only and it is not supported for the PKI authenticators.

Email OTP

In the **Email OTP** authentication method, the server sends an email with a one-time password (OTP) to the user's e-mail address. The user must specify the OTP on the device where the user needs to get authenticated. It is a best practice to use the Email OTP authentication method with other methods such as **Password** or **LDAP Password** to achieve multi-factor authentication and to prohibit malicious users from sending SPAM mails to a user's email box with authentication requests.

To configure the Email OTP method, specify the following details:

Parameter	Description
OTP period	Lifetime of an OTP token in seconds. The default OTP period is 120 seconds. Maximum value for the OTP period is 360 seconds.
OTP Format	Length of an OTP token. The default value is 6 digits.
Sender email	Email address of the sender.
Subject	Subject of the mail.
Format	Format of an email message. The default format is Plain Text . The HTML format allows to use embedded images. You can specify an HTML format of the message in HTML .
Body	For the Plain Text format, you can specify the following variables: <ul style="list-style-type: none">◆ {user}: Username.◆ {endpoint}: Device that a user authenticates to.◆ {event}: Name of the event where the user is trying to authenticate to.◆ {otp}: One-Time-Password to be sent to the user.
Allow to override email address	Option that allows to prevent users from providing an email address that is not registered in the LDAP repository. The option is set to ON by default. Set to OFF to prevent users to specify a different email address during the enrollment.

Emergency Password

The **Emergency Password** method facilitates the use of a temporary password for users if they lose a smartcard or forget their smartphone. Only a helpdesk administrator can enroll the Emergency Password method for users.

WARNING: An administrator can misuse this method by trying to access other user's account. Full administrator must be vigilant to select the right helpdesk administrators.

To configure the Emergency Password method, specify the following details:

Parameter	Description
Minimum password length	The length of the password must be at least five characters long.
Password age (days)	The validity period of a password. The default value is 3 days.
Max logons	The maximum number of login attempts that a user can perform before the password gets expired. The default value is 10.
Complexity requirements	Set to ON to enforce users creating a complex password. Password must meet the following requirements: <ul style="list-style-type: none"> ◆ Contains at least one uppercase character ◆ Contains at least one lowercase character ◆ Contains at least one digit ◆ Contains at least one special character
Allow change options during enrollment	When set to ON , this option allows a helpdesk administrator to set Start date , End date , and Maximum logons manually in the Helpdesk portal. This manual configuration overrides the settings in the Emergency Password method.

Fingerprint

The **Fingerprint** method is one of the strong biometric authentication methods that Advanced Authentication provides. Users can authenticate with methods such as **Password** (something they know) and **Fingerprint** (something they are) for multi-factor authentication. Users need to place their finger on a fingerprint scanner to enroll and authenticate.

To configure the Fingerprint method, perform the following steps:

- 1 Set the **Similarity score threshold** by moving the slider to the desired score.

NOTE: Default and recommended value for **Similarity score threshold** is 25. Reducing the score may result in different fingerprints getting validated.

- 2 Select the number of fingers to be enrolled.

It is recommended to enroll more than one finger as any injuries to the enrolled finger may make it unable to use.

- 3 Select the number of scans required for enrollee's each finger.

NOTE: To improve the quality of the fingerprint enrollment, it is recommended to have multiple captures. The total number of captures including all the enrolled fingers must not exceed 25.

- 4 Click **Save**.

FIDO U2F

With the **FIDO U2F** authentication method, users can authenticate with the touch of a finger on the U2F device.

You can configure certificate settings for the FIDO U2F authentication method. By default, Advanced Authentication does not require the attestation certificate for authentication by the FIDO U2F compliant token. Ensure that you have a valid attestation certificate added for your FIDO U2F compliant token, when you configure this method. The Yubico and Feitian attestation certificates are pre-configured in the Advanced Authentication appliance.

Advanced Authentication supports the Microsoft policy [Interactive logon: Smart card removal behavior](#) that allows you to specify an action on the U2F. You can configure the policy to perform a force log off or lock a session when a user removes the U2F device from a computer. This policy is supported for Windows only. When the user removes the U2F device from the computer, the Windows Client runs an action that is specified in the [Interactive logon: Smart card removal behavior policy](#).

IMPORTANT: To use the FIDO U2F authentication for Access Manager in the **OAuth 2.0** event, you must configure an external web service to perform enrollment and authentication for one domain name. For more information, see [Configuring a Web Server to Use the FIDO U2F Authentication](#).

The YubiKey tokens may flash with a delay when the token is initialized in a combination mode. For example, when authentication uses OTP and U2F methods. This may cause the users to wait for the token to flash before enrollment or authentication. Therefore, it is recommended to flash the tokens only in the U2F mode if the other modes are not needed.

You can configure the following settings for this method:

- ◆ [Configuring Facets](#)
- ◆ [Configuring a Web Server to Use the FIDO U2F Authentication](#)

Configuring Facets

You can add a list of facets for the FIDO U2F tokens to work on multiple sub-domains of a single domain.

Previously, the U2F RFC standards allowed authentication only on the domain name on which the enrollment was done. But with the FIDO U2F standards update, the FIDO alliance introduces facets that allows users to authenticate even on domains on which the enrollment is not done.

For example, if a user enrolls a token on `https://some.domain` and wants to get authenticated on `https://app.some.domain`, you as an administrator can do this by adding `https://app.some.domain` as a facet of the primary domain `https://some.domain`.

WARNING: Even if you are not using the facets, ensure to configure the **Facets primary server URL suffix** to enable the users to authenticate with the FIDO U2F method. If the **Facets primary server URL suffix** is not configured then while authenticating with FIDO U2F, the user is prompted with a message `The visited URL doesn't match the application ID or it is not in use`.

To add facets, perform the following steps:

- 1 Expand **Facets settings**.
- 2 Specify the suffix of the primary facet in **Facets primary server URL suffix**. For example, you can specify `some.domain`.

NOTE: In **Facets primary server URL suffix**, if you specify any value with `https://` then user cannot enroll the U2F method.

3 Click **Add** to add prefixes for the facets.

4 Specify the prefix of the facet in **Facets prefixes**. For example, `app`.

From the above example, if a user logs in to `https://app.some.domain` with the U2F token enrolled on `https://some.domain`, the browser sends a plain GET request to the `https://URL/<tenant-ID/app-id.json` URL and waits for the list of allowed facets (sub-domains). If the list is returned, browser allows the user to use token on the URLs specified in the Facets prefixes list.

5 Click **Save**.

NOTE: The facets are supported only on the Google Chrome. The support for sub-domains is not stabilized in Chrome, so you may get an error message `The visited URL doesn't match the application ID` or it is not in use during enrollment and authentication.

Configuring a Web Server to Use the FIDO U2F Authentication

This section is applicable for Debian 8 Jessie. The procedure may differ for other distributives.

This sections explains how to configure web server to use the FIDO U2F authentication in NetIQ Access Manager for the **OAuth 2.0** event.

According to the FIDO U2F specification, both enrollment and authentication must be performed for one domain name. As NetIQ Access Manager and Advanced Authentication appliance are located on different servers, you must configure web server to enable performing the following actions:

- ◆ Port forwarding to Advanced Authentication appliance for the FIDO U2F method enrollment
- ◆ Port forwarding to NetIQ Access Manager for further authentication using FIDO U2F tokens

Perform the following actions to configure a web server to use the FIDO U2F authentication.

Installing Nginx Web Server

You must install the Nginx web server for URL forwarding.

To install Nginx, add the following two lines to the `/etc/apt/sources.list` file:

```
deb http://packages.dotdeb.org jessie all
deb-src http://packages.dotdeb.org jessie all
```

Preparing SSL Certificate

Run the following commands:

```
mkdir -p /etc/nginx/ssl
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/nginx/ssl/proxy.key -out /etc/nginx/ssl/proxy.crt
```

Preparing Nginx Proxy Configuration

Add the following to the `/etc/nginx/sites-available/proxy` file:

```

server {
listen 443 ssl;
error_log /var/log/nginx/proxy.error.log info;
server_name nam.company.local;
ssl_certificate /etc/nginx/ssl/proxy.crt;
ssl_certificate_key /etc/nginx/ssl/proxy.key;
location ~ ^/account {

proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-Server $host;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header Host $host;
proxy_pass https://<appliance_IP>$uri?$args;
}
location ~ ^/static {

proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-Server $host;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header Host $host;
proxy_pass https://<appliance_IP>$uri?$args;
}
location ~ ^/admin {

proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-Server $host;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header Host $host;
proxy_pass https://<appliance_IP>$uri?$args;
}
location / {

proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-Server $host;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header Host $host;
proxy_read_timeout 300;
proxy_pass https://<NAM_IP>;
}
}

```

Create a link and restart the nginx service running the following commands:

```

ln -s /etc/nginx/sites-available/proxy /etc/nginx/sites-enabled/proxy
service nginx reload

```

Adding DNS Entries

Ensure that the NetIQ Access Manager name server corresponds to the IP address of web server.

Enrolling U2F FIDO

To enroll U2F, open the link `https://<NAM_FQDN>/account`. The Self-Service portal of Advanced Authentication server appliance is displayed.

Enroll the U2F method in the Self-Service portal. For information about enrolling, see [“Enrolling the Authentication Methods”](#).

LDAP Password

In the **LDAP Password** method, the Advanced Authentication client retrieves password that is stored in the user repository from the Advanced Authentication server.

If you do not include the LDAP Password method in a chain, you will be prompted to perform a synchronization. When you set **Save LDAP password** to **ON**, the prompt is displayed only for the first time until the password is changed or reset. If you set this option to **OFF**, a prompt for synchronization is displayed each time.

To configure LDAP Password method, perform the following steps:

- ◆ Set **Enable SSPR integration** to **ON** if you want to enable the Self Service Password Reset integration for Advanced Authentication web portals.
- ◆ Specify the **SSPR link text**. This link is displayed on the login page where user enters the LDAP Password.
- ◆ Specify the **SSPR URL**. This URL points to the Self Service Password Reset portal.

LDAP password is stored on the Advanced Authentication server in two places:

1. User data: It is used for OS logon (Windows Client, Mac OS X Client, and Linux PAM Client) and is stored when **Save LDAP password** option in **LDAP Password** method is set to **ON**.
2. LDAP password authenticator: It is used while using cached logon. The password is stored when the **Enable local caching** option is set to **ON** in the **Cache Options Policy**.

OATH OTP

OATH (Initiative for Open Authentication) is an industry-wide collaboration to develop an open reference architecture using open standards to promote the adoption of strong authentication using OTP.

Advanced Authentication supports the following two different types of OATH OTP:

- ◆ **HOTP**: Counter based OTP
- ◆ **TOTP**: Time based OTP

For HOTP, you can specify the following parameters:

- ◆ **OTP format**: The number of digits in the OTP token. The default value is 6 digits. The value must be the same as of the tokens you are using.
- ◆ **OTP window**: The number of OTPs that the Advanced Authentication server will generate starting from the current HOTP counter value to match an HOTP entered by the user during authentication. The default value is 10. The maximum value for the OTP window is 100000 seconds.

This is required when users use tokens for accessing websites such as Google. After each use, the HOTP counter increases by 1. Therefore, the counter will be out of sync between the token and Advanced Authentication server. Also, users can press the token button accidentally.

WARNING: Do not increase the HOTP window value to more than 100 as it may decrease the security by causing false matches.

During enrollment or HOTP counter synchronization in the Self-Service portal, **Enrollment HOTP window** that has a value of 100,000 is used. This helps in the following:

- ♦ HOTP tokens may be used for a long period before the enrollment in Advanced Authentication and the value is unknown and can be equal to some thousands.
- ♦ Secure because users must provide 3 consequent HOTPs.

For TOTP, you can specify the following parameters:

- ♦ **OTP period (sec):** The value to specify how often a new OTP is generated. The default value is 30 seconds. The maximum value for the OTP period is 360 seconds.
- ♦ **OTP format:** The number of digits in the OTP token. The default value is 6 digits. The value must be the same as the tokens you are using.
- ♦ **OTP window:** The value to specify the periods used by Advanced Authentication server for TOTP generation. For example, if you have a period of 30 and a window of 4, then the token is valid for 4*30 seconds before current time and 4*30 seconds after current time, which is 4 minutes. These configurations are used because time can be out-of-sync between the token and the server and may impact the authentication. The maximum value for the OTP window is 64 periods.

IMPORTANT: It is not recommended to use an OTP window equal to 32 and higher for 4-digit OTP because it reduces security.

- ♦ **Google Authenticator format of QR code (Key Uri):** Option to scan the QR code for enrollment of the software token with the Google Authenticator. Enable this option to use the Google Authenticator or Microsoft Authenticator app in place of the Advanced Authentication smartphone app to scan a QR code.

IMPORTANT: OTP format must be set to 6 digits when you use the Google Authenticator or Microsoft Authenticator format of QR code.

Importing PSKC or CSV Files

You can import the **PSKC** or **CSV** files. These token files contain token information. To import these files, perform the following steps:

- 1 Click the **OATH Token** tab.
- 2 Click **Add**.
- 3 Click **Browse** and add a **PSKC** or **CSV** file.
- 4 Choose a **File type**. The options are:
 - ♦ **OATH compliant PSKC:** This file type must be compliant with OAuth. For example, HID OATH TOTP compliant tokens.
 - ♦ **OATH csv:** This file type must contain the format as described in [CSV File Format To Import OATH Compliant Tokens](#). You cannot use the YubiKey CSV files.
 - ♦ **Yubico csv:** In this file type, you must use one of the supported **Log configuration output** (see [YubiKey Personalization Tool > Settings tab > Logging Settings](#)) formats with comma as a delimiter.
 - ♦ Traditional format: In this file type, **OATH Token Identifier** must be enabled.
 - ♦ Yubico format: This file type is supported only for **HOTP Length** set to **6 Digits** and **OATH Token Identifier** set to **All numeric**.

IMPORTANT: **Moving Factor Seed** must not exceed 100000.

- 5 Add the encrypted **PSKC** files. For this, select **Password** or **Pre-shared key** in **PSKC file encryption type** and provide the information.
- 6 Click **Upload** to import tokens from the file.

NOTE: Advanced Authentication receives an **OTP format** from the imported tokens file and stores the information in the enrolled authenticator. Therefore, you need not change the default value of **OTP format** on the **Method Settings Edit** tab.

When the tokens are imported, you can see the list and you must assign the tokens to users. This can be done in the following two ways:

- ◆ Click **Edit** next to the token and select **Owner** and click **Save**.
- ◆ A user can self-enroll a token in the Self-Service portal. Administrator must let the user know an appropriate value from the **Serial** column for the self-enrollment.

NOTE: **Tenancy settings** are not supported for the OATH tokens. Therefore, the configurations in the **OATH Tokens** tab cannot be enforced on tenant administrators.

CSV File Format To Import OATH Compliant Tokens

A CSV file, which is imported as **OATH csv** file in the **Administration portal > Methods > OATH OTP > OATH Tokens** tab, must contain fields with the following parameters:

- ◆ Token's serial number
- ◆ Token's seed
- ◆ (Optional) Type of the token: TOTP or HOTP (by default HOTP)
- ◆ (Optional) OTP length (default value is 6 digits)
- ◆ (Optional) Time step (default value is 30 seconds)

Comma is a delimiter.

The following is an example of a CSV file:

```
Token001, 15d2fa517d3c6b791bd4cc2044c241429307001f
Token002, 8c557fc050721037fd31e1d3345b5d3263263e0f, totp, 8
Token003, 658208efea5ac49d5331ba781e66f2c808cccc8e, hotp, 6
Token004, 89f0dfe1c90379da6a11aaca2fc1070f606efe36, totp, 6, 60
```

IMPORTANT: For the YubiKey tokens, you must use the traditional format of the CSV (check **YubiKey Personalization Tool > Settings tab > Logging Settings**) with comma as a delimiter. Use Yubico csv file type (**Advanced Authentication Administration portal > Methods > OATH OTP > OATH Tokens**).

Password

In the **Password** authentication method, you can configure security options for passwords that are stored in the appliance. For example, the **local/admin** user who does not have an LDAP Password can use this option.

NOTE: Do not use the **Password** method in chains that contain only one factor. You must always combine the **Password** method with other factors.

You can configure the following options for the **Password** method:

- ◆ **Minimum password length:** The maximum length of the password.
- ◆ **Maximum password age:** The validity period of the password. The default value is 42 days. If you set the value to 0, the password never expires.
- ◆ **Complexity requirements:** Option to enable users to create a complex and not easily detectable password. Set to **ON** to enable this option. Password must meet the following requirements:
 - ◆ Contains at least one uppercase character
 - ◆ Contains at least one lowercase character
 - ◆ Contains at least one digit
 - ◆ Contains at least one special character
- ◆ **Rename to PIN:** Option to rename the Password to PIN. Set to **ON** to enable the option. The **Password** method is renamed to **PIN** in the Advanced Authentication Administration portal, Helpdesk portal, Self-Service portal, Windows Client, Mac OS X Client, and Linux PAM Client.

IMPORTANT: Advanced Authentication does not generate notifications about the password expiry. After the password expires, the local administrator cannot sign-in to the Administration portal and users using this method cannot get authenticated.

However, an administrator and a user can change their passwords in the Self-Service portal.

PKI

The Public Key Infrastructure (PKI) creates, stores, and distributes digital certificates. These certificates are used to verify whether a particular public key belongs to a specific entity.

In the PKI method, you must upload trusted root certificates. These certificates must meet the following requirements:

1. **Root CA** certificate is in the `.pem` format.
2. All certificates in the certification path (except Root CA) contain **AIA** and **CDP** http link to check revocation status.
3. The certificate for PKI device contains a key pair: public and private key in the x509 format. The certificates that do not comply with the requirements are ignored and hidden during enrollment.

NOTE: Advanced Authentication supports the `p7b` format of parent certificates. These `p7b` format files can contain certificates and chain certificates, but not the private key. They are Base64 encoded ASCII files with extensions `.p7b` or `.p7c`.

Configuring the Environment for a Standalone Root CA

- 1 Install **Web Server (IIS) Role**.
- 2 Create the `CertEnroll` Folder and grant **Share & NTFS** permissions to the **Cert Publishers** group.
- 3 Create **CertEnroll Virtual Directory** in IIS.
- 4 Enable **Double Escaping** on IIS Server.

- 5 Install **Enterprise Root CA** using Server Manager.
- 6 Enable **Object Access Auditing** on CA.
- 7 Configure the **AIA** and **CDP**.
- 8 Publish the Root CA Certificate to AIA.
- 9 Export **Root CA** in `.der` format and convert the format to `.pem`.
- 10 Export personal certificate (that was signed by Root CA) with private key and place it on a PKI device.

Configuring the Environment for a Subordinate CA

- 1 Install **Web Server (IIS) Role**.
- 2 Create the `CertEnroll` Folder and grant **Share & NTFS** permissions to **Cert Publishers** group.
- 3 Create **CertEnroll Virtual** Directory in IIS.
- 4 Enable **Double Escaping** on IIS Server.
- 5 Install the **Standalone Offline Root CA**.
- 6 Create a `CAPolicy.inf` for the standalone offline root CA.
- 7 Installing the **Standalone Offline Root CA**.
- 8 Enable **Auditing** on the Root CA.
- 9 Configure the **AIA** and **CDP**.
- 10 Install Enterprise Issuing CA.
- 11 Create `CAPolicy.inf` for Enterprise Root CA.
- 12 Publish the **Root CA Certificate** and **CRL**.
- 13 Install **Subordinate Issuing CA**.
- 14 Submit the Request and Issue subordinate **Issuing CA Certificate**.
- 15 Install the subordinate **Issuing CA Certificate**.
- 16 Configure **Certificate Revocation** and **CA Certificate Validity Periods**.
- 17 Enable **Auditing** on the Issuing CA.
- 18 Configure the **AIA** and **CDP**.
- 19 Install and configure the **Online Responder Role Service**.
- 20 Add the **OCSP URL** to the subordinate Issuing CA.
- 21 Configure and publish the **OCSP Response Signing Certificate** on the subordinate Issuing CA.
- 22 Configure **Revocation Configuration** on the **Online Responder**.
- 23 Configure **Group Policy** to provide the OCSP URL for the subordinate Issuing CA.
- 24 Export **Root CA** in `.der` format and convert the format to `.pem`.
- 25 Export personal certificate (that was signed by subordinate CA) with private key and place it on a PKI device.

For more information, see [Single Tier PKI Hierarchy Deployment](#) and [Two Tier PKI Hierarchy Deployment](#).

To upload a new trusted root certificate, perform the following steps.

- 1 Click **Add** in the **PKI Method Settings Edit** page.
- 2 Click **Browse**.

- 3 Choose a `.pem` certificate file and click **Upload**.
- 4 Click **Save**.

NOTE: You must upload only the **Root CA** on appliance.

RADIUS Client

In the **RADIUS Client** method, Advanced Authentication forwards the authentication request to a third-party RADIUS server. This can be any RADIUS server. For example, you can use RADIUS Client as an authentication method when you have a token solution such as RSA or Vasco. You want to migrate users to Advanced Authentication with the flexibility that users can use the old tokens while the new users can use any of the other supported authentication methods.

You can configure the following options for the **RADIUS Client** method:

- ◆ **Server:** The Hostname or IP address of the third-party RADIUS server.
- ◆ **Secret:** The shared secret between the RADIUS server and Advanced Authentication.
- ◆ **Port:** The port to where the RADIUS authentication request is sent. The default port is 1812.
- ◆ **Send repository name:** Option for a repository name to be used automatically with a username. For example, `company\pjones`. Set to **ON** to enable the option.
- ◆ **NAS Identifier:** An attribute that contains a string identifying the NAS originating the Access-Request. It is only used in Access-Request packets. Either `NAS-IP-Address` or `NAS-Identifier` must be present in an Access-Request packet.

SMS OTP

In the **SMS OTP** authentication method, a one time password (OTP) is sent with the SMS text to the user's phone. The user receives the OTP and enters it on the device where the authentication is happening. The OTP must be used within a specific time frame. The OTPs delivered through text messages prevent phishing and malicious attacks. SMS OTP is recommended to be used with other methods, such as Password or LDAP Password.

NOTE: In the User's settings of a repository, ensure that a phone number without extension is used. An SMS is not sent to the user's mobile where the phone number contains an extension.

To configure the SMS OTP method, specify the following details:

- ◆ **OTP Period:** The lifetime of an OTP in seconds. The default value is 120 seconds. The maximum value for the OTP period is 360 seconds.
- ◆ **OTP Format:** The number of digits in the OTP. The default value is 6.
- ◆ **Body:** The text in the SMS that is sent to the user. The following structure describes the text in the OTP:
 - ◆ `{user}`: Name of the user.
 - ◆ `{endpoint}`: Device the user is authenticating to.
 - ◆ `{event}`: Name of the event where the user is trying to authenticate to.
 - ◆ `{otp}`: One-Time Password.

- ♦ **User cell phone attribute:** The cell phone number of a user on which the OTP is sent through SMS. You can use custom attributes such as `mobile`, `homePhone`, `ipPhone`, and other attributes of a repository. You must define the attribute in “[User Cell Phone Attributes](#)” of the **Repositories** section.

NOTE: If you do not configure the attribute in the method settings, then the first attribute defined in the “[User Cell Phone Attributes](#)” section of Repository configuration is used when the user tries to authenticate. For example, if you define `mobile` as the first attribute in **User cell phone attribute** and do not configure the attribute in method settings of **SMS OTP**, then while authenticating, the first attribute, which is the `mobile` attribute, is used for the **SMS OTP** method authentication.

- ♦ **Allow to override phone number:** Option that allows to prevent users from providing a phone number that is not registered in the LDAP repository. The option is set to **ON** by default. Set to **OFF** to prevent users to specify a different phone number during the enrollment.

Security Questions

In **Security Questions** authentication method, an administrator can set up a series of predefined questions. A user must answer these questions to get authenticated. Security Questions are used when users forget their passwords.

Security questions are often easy to guess and can often bypass passwords. Therefore, Security Questions do not prove to be secure.

You must follow few guidelines to use this method. You must use **Good** security questions that meet five criteria. Ensure that the answers to a good security question are:

1. **Safe:** Cannot be guessed or researched.
2. **Stable:** Does not change over time.
3. **Memorable:** Can be remembered.
4. **Simple:** Precise, easy, and consistent.
5. **Many:** Has many possible answers.

Some examples of good, fair, and poor security questions according to goodsecurityquestions.com are as follows. For a full list of examples, see the website ([http://goodsecurityquestions.com/.](http://goodsecurityquestions.com/))

GOOD

- ♦ What is the first name of the person you first kissed?
- ♦ What is the last name of the teacher who gave you your first failing grade?
- ♦ What is the name of the place your wedding reception was held?
- ♦ In what city or town did you meet your spouse/partner?
- ♦ What was the make and model of your first car?

FAIR

- ♦ What was the name of your elementary / primary school?
- ♦ In what city or town does your nearest sibling live?
- ♦ What was the name of your first stuffed animal, doll, or action figure?

- ♦ What time of the day were you born? (hh:mm)
- ♦ What was your favorite place to visit as a child?

POOR

- ♦ What is your pet's name?
- ♦ In what year was your father born?
- ♦ In what county where you born?
- ♦ What is the color of your eyes?
- ♦ What is your favorite _____?

Configure the following options for the **Security Questions** method:

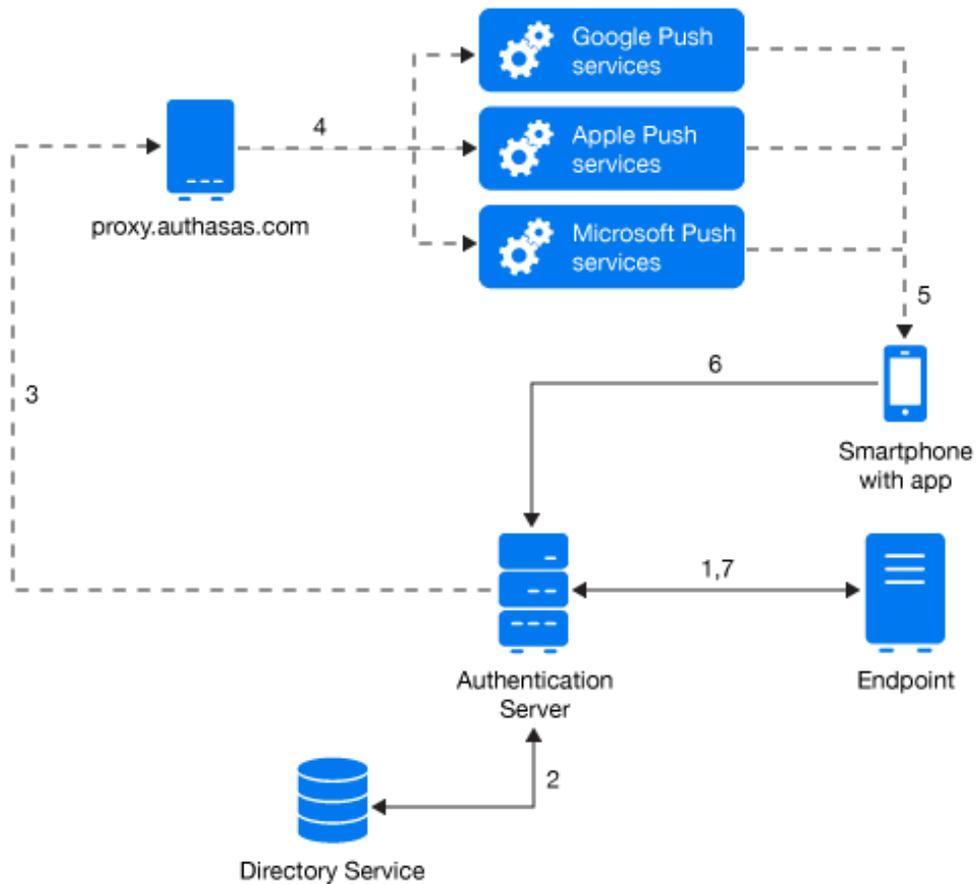
- ♦ **Min. answer length:** The minimum number of characters an answer must contain.
- ♦ **Correct answers for logon:** The number of answers a user must answer correctly to get access.
- ♦ **Total questions for logon:** The number of questions that are presented to the user while authenticating.

For example, if the **Correct answers for logon** is set to 3 and the **Total questions for logon** is set to 5, the user needs to specify only 3 correct answers out of a set of 5 questions.

Smartphone

Advanced Authentication provides the **Smartphone** method that facilitates users to authenticate through their Smartphone. The authentication happens through the NetIQ smartphone app to perform the out-of-band authentication. The out-of-band authentication is typically a two-factor authentication that requires a secondary verification through a separate communication channel along with the ID and password.

The authentication flow for the Smartphone method in Advanced Authentication is described in the following image.



A user wants to authenticate on an endpoint such as a laptop or a website with the Smartphone method. The following steps describe the authentication flow:

- 1 When the authentication request is initiated, the endpoint contacts the Advanced Authentication server.
- 2 The Advanced Authentication server validates the user's credentials.
- 3 After validating the credentials, the Advanced Authentication server sends a push message to proxy.authsas.com.
- 4 Depending on the platform of the Smartphone, the server selects an appropriate push service and then forwards the push message to the Smartphone.
- 5 The push message is then delivered to the user's Smartphone to inform that an authentication request has been initiated.
- 6 When the user opens the Smartphone app, the app reaches the Advanced Authentication server to validate if there is an authentication needed. The authentication is indicated by the **Accept** and **Reject** options. The user's selection is then sent to the server.
- 7 Finally, the server validates the authentication and the endpoint gets authenticated.

HTTPS protocol is used for the communication.

This authentication method is recommended to use in combination with another method such as Password or LDAP Password to achieve multi-factor authentication and protect a user from getting SPAM push messages.

Access Configurations

The following are the configurations required for the Smartphone method.

- ◆ Advanced Authentication server must be accessible by the specified **Server URL** address from smartphones (HTTPS, outbound).
- ◆ Advanced Authentication server must have a permitted outbound connection to proxy.authasas.com (HTTPS).

Scenario for Authenticating with the Smartphone Method

Bob wants to authenticate on the **myexample.com** website. When he logs in to the website, the Smartphone authentication method sends a push message to Bob's mobile phone. When he opens the Smartphone app installed on his phone, he sees **Accept** and **Reject** options. If he selects the **Accept** option, the authentication request is sent over the mobile network (secure) back to the Authentication framework. Without specifying an OTP code, Bob has been authenticated to **myexample.com**.

When your smartphone does not have a network connection, you can use a backup OTP as offline authentication.

To configure the Smartphone method, specify the following details:

Parameter	Description
Push salt TTL	The lifetime of an authentication request sent to the smartphone.
Learn timeout	The time that is valid for the user to scan the QR code for enrollment.
Auth salt TTL	The lifetime in which the out-of-band authentication needs to be accepted before authentication fails.
TOTP Length	The length of OTP token used for backup authentication.
TOTP step	The time a TOTP is displayed on a screen before the next OTP is generated. The default time is 30 seconds.
TOTP time window	The time in seconds in which the TOTP entered is accepted. The default time is 300 seconds.
Server URL	The URL of Advanced Authentication server to where the smartphone app connects for authentication.
Require PIN	Set to ON to enforce the Enable PIN setting for the Smartphone application. A user will not be able to edit the settings on the Smartphone NOTE: If the PIN is not set, then the user is prompted to set the PIN during authentication.
Minimum PIN length	The minimum PIN length. The available options are 4,5, and 6.
Require biometrics	Set to ON to enforce the fingerprint setting for the Smartphone application. A user will not be able to edit the settings on the Smartphone.

Parameter	Description
Use image on mobile devices	<p>Select the option to use a customized image on your Smartphone app.</p> <p>Browse the image. This image is displayed in the About screen of your Smartphone app. The resolution of the image must be 2732×637 pixels.</p> <p>NOTE: The Require PIN, Require biometrics, and Use image on mobile devices policies are automatically applied on the smartphone if a user has an enrolled authenticator in the smartphone app and the app is open on one of the screens: Authentication Requests, Enrolled Authenticators, or Requests History. It takes 2 to 30 seconds to display the authentication request.</p> <ul style="list-style-type: none"> ◆ If a user has configured a 4-digit PIN but a 6-digit PIN has been enforced by the administrator, then the user will be able to use the 4-digit PIN until the user decides to change the PIN. ◆ If Require biometrics is set in the policies, but a user's device does not support fingerprint, the policy will not be applied for the device. ◆ If a user has authenticators enrolled for two different Advanced Authentication servers with different policies, then the policies are combined for the device and the most secure policies are applied for the app.
Geo Zones	<p>You can configure Geo-fencing with the Smartphone method. Geo-fencing allows you to authenticate with the Smartphone method with one more factor, which is the geographical location. When you enable geo-fencing, users will be able to authenticate with Smartphone from only allowed geographical locations. You must enable the policy Geo Fencing Options to use geo-fencing.</p> <p>To configure geo-fencing, you need to draw a boundary of the location to be authenticated with a polygon. To configure geo-fencing, perform the following steps:</p> <ol style="list-style-type: none"> 1. Click Add. 2. Specify the name of the zone. 3. Click the Search icon and specify the address to locate the required geographical location. <p>You can click the full-screen  icon to view the map in the full screen.</p> <ol style="list-style-type: none"> 4. Click the polygon  icon in the menu bar of the map. 5. Click the starting point on the map and draw the boundary of the specific location to be authenticated. 6. Click to mark the end point of the boundary after you have finished drawing the geo zone. <p>You can also edit the marked polygon by clicking the edit  icon.</p> <ol style="list-style-type: none"> 7. Click Save.
<p>NOTE: To use geo-fencing, ensure that access to the location is enabled for the NetIQ Advanced Authentication app on the smartphone.</p>	

Swisscom Mobile ID

In the **Swisscom Mobile ID** authentication method, a PKI- based mobile signature secure encryption technology is stored on a user's SIM card. When the user tries to authenticate, the Swisscom Mobile ID is validated against the user's mobile phone attribute in the repository. If the number is validated, the user gets authenticated.

To configure the Swisscom Mobile ID method, specify the following details:

- ♦ **Application provider ID:** Identifier of the application provider.
- ♦ **Application provider password:** Password of the application provider.
- ♦ **Swisscom Mobile ID service URL:** Interface of the Swisscom Mobile ID.
- ♦ **Notification message prefix:** Message that is displayed on the user's mobile as a notification.

In addition, you can upload the Swisscom client certificates as follows:

1. Browse **Client SSL certificate**. The required certificate must be in a `.pem` format and self-signed with a private key.
2. Specify **Private key password** for the certificate.
3. Click **Save**.

NOTE: Users must activate the Mobile ID service for the [Swisscom SIM card](#).

For more information about the Swisscom Mobile ID method, see the [Mobile ID Reference guide](#).

Voice

In the **Voice** authentication method, a user receives a call with an OTP through voice.

The following workflow describes the Voice authentication method in Advanced Authentication:

- 1 A user tries to authenticate with the Voice method.
- 2 The user receives a call on the phone with a OTP.
- 3 User must specify the PIN that has also been enrolled in the Self-Service portal during the enrollment.
- 4 After the user specifies the PIN followed by a hash (#) symbol, user is authenticated with the Voice method.

IMPORTANT: Phone number with extensions are supported for this method.

Special characters “,” and “x” are used to indicate wait time and can be used as separators between phone number and extension.

For example, if +123456789 is the phone number and 123 is the extension, then it can be specified as +123456789,,,,123.

In the above example, “,” is specified 4 times and this multiplied by 0.5 (default value in Twilio) indicates the wait time, which is 2 (4*0.5) seconds. First, call is sent to the number 123456789 and after a wait period of 2 seconds, the extension 123 is dialed.

To configure the Voice method, specify the following details:

- ♦ **Minimum pin length:** The length of the PIN must be at least three characters long.

- ♦ **Maximum pin age:** The validity period of a PIN. The default value is 42 days. If you set the age to 0, the PIN will not expire.
- ♦ **User cell phone attribute:** The cell phone number of a user that is used to call the user for voice authentication. You can use custom attributes such as `mobile`, `homePhone`, `ipPhone`, and other attributes of a repository. You must define the attribute in “[User Cell Phone Attributes](#)” of the **Repositories** section.

NOTE: If you do not configure the attribute in the method settings, then the first attribute defined in the “[User Cell Phone Attributes](#)” section of Repository configuration is used when the user tries to authenticate. For example, if you define `mobile` as the first attribute in **User cell phone attribute** and do not configure the attribute in method settings of **Voice**, then while authenticating, the first attribute, which is the `mobile` attribute, is used for the **Voice** method authentication.

IMPORTANT: Advanced Authentication does not notify a user about the expiry of a PIN.

Voice OTP

In the **Voice OTP** authentication method, a user receives an OTP over a call. The user must specify this OTP on the device where the authentication is happening. The OTP must be used within a specific time frame. Voice OTP is recommended to use with other methods, such as Password or LDAP Password.

To configure the Voice OTP method, specify the following details:

- ♦ **OTP period:** The time period for which the Voice OTP is valid. Default time is 120 seconds. The maximum value for the Voice OTP period is 360 seconds.
- ♦ **OTP format:** The length of the Voice OTP token. Default length is 4.
- ♦ **Body:** The text or number in the Voice OTP that is sent to the user. Here, you can specify the `{otp}` variable, which is the actual one-time password. To repeat the one-time password during the call you can specify: Use the OTP for authentication: `{otp}`. OTP: `{otp}`.
- ♦ **User cell phone attribute:** Cell phone number of a user that is used to send the OTP through a call. You can use custom attributes such as `mobile`, `homePhone`, `ipPhone`, and other attributes of a repository. You must define the attribute in “[User Cell Phone Attributes](#)” of the **Repositories** section.

NOTE: If you do not configure the attribute in the method settings, then the first attribute defined in the “[User Cell Phone Attributes](#)” section of Repository configuration is used when the user tries to authenticate. For example, if you define `mobile` as the first attribute in **User cell phone attribute** and do not configure the attribute in method settings of **Voice OTP**, then while authenticating, the first attribute, which is the `mobile` attribute, is used for the **Voice OTP** method authentication.

- ♦ **Allow to override phone number:** Option that allows to prevent users from providing a phone number that is not registered in the LDAP repository. The option is set to **ON** by default. Set to **OFF** to prevent users to specify a different phone number during the enrollment.

Creating a Chain

A chain is a combination of authentication methods. A user must pass all methods in the chain to be successfully authenticated. For example, if you create a chain with LDAP Password and SMS OTP, a user must first specify the LDAP Password. If the LDAP password is correct, the system sends an SMS with a One-Time-Password (OTP) to the user's mobile. The user must specify the correct OTP to be authenticated.

Advanced Authentication contains the following chains that are created by default:

1. **LDAP Password Only:** Any user from a repository can use this chain to get authenticated with the LDAP Password (single-factor) method.
2. **Password Only:** Any user who has a Password method enrolled can use this chain to get authenticated with the Password (single-factor) method.

You can create any number of chains with multiple authentication methods. To achieve better security, you can include multiple methods in a chain.

Authentication comprises of the following three factors:

- ◆ **Something that you know** such as password, PIN, and security questions.
- ◆ **Something that you have** such as smartcard, token, and mobile phone.
- ◆ **Something that you are** such as biometrics (fingerprint or iris).

You can achieve multi-factor or strong authentication by using any two factors out of this list. For example, multi-factor authentication can include a combination of password and a token or a smartcard and a fingerprint.

After you create a chain, you can use the chain on specific user groups in your repository. The chain is then mapped to an event.

To create a new chain or edit an existing chain, perform the following steps:

- 1 Open the **Chains** section.
- 2 Click **Add** to create a chain. You can also click the edit icon  against the chain that you want to edit.
- 3 Specify a name of the chain in **Name**.
- 4 Specify a **Short name**. The short name is used by a user to move to a chain. For example, if you name a chain containing the LDAP Password and SMS methods as **SMS**, then a user can specify <username> sms and the user is forced to use **SMS** as the chain. This is helpful in scenarios when the primary chain is not available.

NOTE: This is applicable only for the RADIUS Server event.

- 5 Set **Is enabled** to **ON** to enable the chain.
- 6 Select the methods that you want to add to the chain from the **Methods** section. You can prioritize the methods in the list. For example, if you create a chain with LDAP Password and HOTP methods, then the user will be prompted for the LDAP Password method first and then the OTP.
- 7 Specify the groups that will use the authentication chain in **Roles & Groups**.

IMPORTANT: It is not recommended to use groups in Active Directory, from which you will not be able to exclude users. This is because you will not be able to free up a user's license.

- 8 Expand **Advanced Settings** by clicking +.
- 9 Set **Apply if used by endpoint owner** to **ON** if an **Endpoint owner** must use the chain.

NOTE: The Endpoint owner feature is supported for Windows Client, Mac OS Client, and Linux PAM Client only.

- 10 Specify the **MFA tags**. When a user logs in to Windows on a workstation with Advanced Authentication Windows Client installed, the user's account is moved to the group specified in **MFA tags**.

NOTE: This functionality is available when you set the **Enable filter** to **ON** in the **Logon Filter for AD** policy and have configured the **Logon Filter**.

For example if you specify a **Card users** group from Active Directory in **MFA tags**, then the user will be moved from the legacy group (specified in the **Advanced Settings** of Active Directory repository) to the **Card users** group.

- 11 Set **Required chain** to **Nothing**, if this is a normal (high-security) chain. If you want to configure a simple chain within a specific time period after successful authentication with a high-security chain, choose an appropriate high-security chain. In this case you also need to specify a **Grace period (mins)**. Within this time period the chain will be used instead of the appropriate high-security chain. The maximum value for grace period is 44640 minutes (31 days).

NOTE: You must assign both a high-security chain and a simple chain to an Event. The simple chain must be of higher order than the corresponding high-security chain. The options are available when the **Last logon tracking options** policy is set to **ON**.

For example, **LDAP Password+Card** is a high-security chain and **Card** is a simple chain. The users must use **LDAP Password+Card** chain once in every 8 hours and within this period, they must provide only the **Card** method to authenticate.

- 12 Click **Save**.

IMPORTANT: If you have configured more than one chain using one method (for example, **LDAP Password**, **LDAP Password+Smartphone**) and assigned it to the same group of users and the same event, then the top chain is always used if the user has enrolled all the methods in the chain. An exception is the use of a high-security chain and its appropriate simple chain, where the simple chain must be higher than its high-security chain.

Configuring Events

Advanced Authentication provides authentication events for the supported applications or devices. You can configure an event to leverage the Advanced Authentication functionalities for the respective application or device. The application or device triggers the respective authentication event when a user tries to access it.

You can create customized events for the following:

- ◆ Third-party integrations.
- ◆ To use Windows Client, Linux PAM Client or Mac OS X Client on both the domain joined and non-domain workstations and it requires to have a separate event to use the non-domain mode.
- ◆ Integrations using SAML 2.0 and OAuth 2.0.

This section contains the following:

- ◆ “Configuring an Existing Event” on page 60
- ◆ “Creating a Customized Event” on page 64

Configuring an Existing Event

1 Click **Events**.

2 Click the edit icon  against the event that you want to edit.

3 Ensure that **Is enabled** is set to **ON** if you want to use the event.

4 Select the event type.

For most of the predefined events, you cannot change the **Event type**. For events such as **Windows logon**, **Linux logon**, and **Mac OS logon**, you can change the **Event type** from **OS Logon (domain)** to **OS Logon (local)** if the workstations are not joined to the domain.

5 Select the **Authenticator category**. The **Authenticator category** option is displayed only if you have added categories in the “**Event Categories**” policy.

6 Select the chains that you want to assign to the current event.

In an event, you can configure a prioritized list of chains that can be used to get access to that specific event.

7 If you want to restrict access of some endpoints to the event, add all the endpoints that must have access to the **Endpoint whitelist**. The remaining endpoints are blacklisted automatically. If you leave the **Endpoints whitelist** blank, all the endpoints will be considered for authentication.

8 Set **Geo-fencing** to **ON** to enable geo-fencing. Move the permitted zones from **Available** to **Used**. For more information about configuring geo-fencing, see the **Smartphone** method.

IMPORTANT: You must enable the **Geo Fencing Options** policy to use the geo fencing functionality.

9 Select **Allow Kerberos SSO** if you want to enable single sign-on (SSO) to the Advanced Authentication portals. Kerberos SSO is supported for AdminUI, Authenticators Management, Helpdesk, and Report logon events.

10 Click **Save**.

11 If you want to revert the changes to the default configuration, click **Initialize default chains**.

NOTE: If you have configured more than one chain using one method (for example, **LDAP Password**, **LDAP Password+Smartphone**) and assigned it to the same group of users and to the same event, the top chain is always used if the user has enrolled all the methods in the chain. An exception is the use of a high-security chain and its appropriate simple chain, where the simple chain must be higher than its high-security chain.

TIP: It is recommended to have a single chain with the **Emergency Password** method at the top of the chains list in the **Authenticators Management** event and other events, which are used by users. The chain will be ignored if the user does not have the **Emergency Password** enrolled. The user can use the Emergency Password immediately after the helpdesk administrator enrolls the user with the Emergency Password authenticator.

NOTE: Configurations that have been set by a top administrator for a particular event are grayed out. The configurations are not displayed, if the configurations are hidden by the top administrator.

By default, Advanced Authentication contains the following events.

- ◆ [ADFS Event](#)
- ◆ [AdminUI Event](#)
- ◆ [Authenticators Management Event](#)
- ◆ [Helpdesk Event](#)
- ◆ [Helpdesk User Event](#)
- ◆ [Linux Logon Event](#)
- ◆ [Mac OS Logon Event](#)
- ◆ [NAM Event](#)
- ◆ [NCA Event](#)
- ◆ [RADIUS Server Event](#)
- ◆ [Report Logon Event](#)
- ◆ [Windows Logon Event](#)

ADFS Event

Use this event to integrate Advanced Authentication with ADFS using the ADFS plug-in. For more information about ADFS, see “[Configuring Advanced Authentication Server](#)” in the *Advanced Authentication - ADFS Plug-in guide*.

NOTE: The ADFS plug-in is discontinued and it is recommended to use integration with ADFS using SAML. For more information about integration with ADFS using SAML, see “[Configuring Integration with ADFS](#)”.

AdminUI Event

Use this event to access the Administration portal. You can configure the chains that can be used to get access to the `/admin` URL.

NOTE: You can promote users or group of users from a repository to the **FULL ADMINS** role in [Repositories > Local](#). After this, you must assign chains in which the methods are enrolled for users with the **AdminUI** event (at a minimum with an LDAP Password).

WARNING: You must be careful when changing the default chains that are assigned to this event. You may block the access to the Administration portal.

Authenticators Management Event

Use this event to access the Self-Service portal. In the Self-Service portal, users can enroll to any of the methods that are configured for any chain and they are a member of the group assigned to the chain.

Add an **LDAP Password** chain as the last chain in the list of chains to ensure secure access to the portal for users who have methods enrolled.

IMPORTANT: If the Administration portal uses a repository that does not have any user, you must enable a chain with **Password** only (Authenticators Management - Password) for this event. This action enables you accessing the Self-Service portal or changing the password in the Self-Service portal.

You can also perform basic authentication with Advanced Authentication. To achieve basic authentication, set the **Allow basic authentication** option to **ON** in the **Event Edit** screen for Authenticators Management.

NOTE: The basic authentication is supported only for the **Authentication Management** event and for the Password (PIN), LDAP Password, and HOTP methods.

You must specify `/basic` with the URL to login to the enrollment page. The Login page appears and the format of the Username you must provide is: `username:PASSWORD|LDAP_PASSWORD|HOTP:1`. For example: `admin:PASSWORD:1`.

When you log in to the Self Service portal, by default the chain with the highest priority is displayed. To display the other chains with the enrolled methods, set **Show chain selection** to **ON**.

NOTE: If you enable to show the chain selection, but a chain is not displayed in the list of available chains in the Self-Service portal, ensure that all the methods of the chain are enrolled by the user.

For more information, see “[Authenticators Management](#)” in the *Advanced Authentication- User* guide.

Helpdesk Event

Configure the settings of this event to enable the Helpdesk administrator to access the Helpdesk portal. One of the roles of a Helpdesk administrator is to set an emergency password for users. An emergency password is a temporary password for users when they lose their smart card or smart phone. Some companies restrict self-enrollment and have the Helpdesk administrator who does the enrollment after hiring. You can promote the repository administrators or users as Helpdesk administrators in the **Repositories > LOCAL > Edit > Global Roles > ENROLL ADMINS** section.

You can manage the enrollment and re-enrollment of the authenticators in one of the following ways:

- ◆ Restrict the self-enrollment and force users to enroll through the Helpdesk.
- Or
- ◆ Restrict only the re-enrollment or deletion of authenticator from the Self-Service portal using the **Disable re-enrollment** option.

For more information, see “[Authenticators Management](#)” in the *Advanced Authentication- Helpdesk Administrator* guide.

Helpdesk User Event

Configure the settings of this event to enable the Helpdesk administrator to authenticate users in the Helpdesk portal. This event is applicable for the **User to manage** screen that appears on the Helpdesk portal.

You must enable the **Ask credentials of management user** option in the [Helpdesk Options](#) policy before using this event.

Linux Logon Event

Configure the settings of this event to enable login to the Linux Client. If you want to use Linux Client on non-domain joined workstations, change the **Event type** from **OS Logon (domain)** to **OS Logon (local)**.

Mac OS Logon Event

Configure the settings of this event to enable login to the Mac OS Client. If you want to use Mac OS Client on non-domain joined workstations, change the **Event type** from **OS Logon (domain)** to **OS Logon (local)**.

NAM Event

Configure the settings of this event to facilitate the integration of Advanced Authentication with [NetIQ Access Manager](#).

NCA Event

Configure the settings of this event to facilitate the integration of Advanced Authentication with [NetIQ CloudAccess](#). CloudAccess must be configured to use Advanced Authentication as an authentication card and user stores must be added for the repositories for the integration to work. For more information, see the Advanced Authentication CloudAccess documentation.

RADIUS Server Event

The Advanced Authentication server contains a built-in RADIUS server to authenticate any RADIUS client using one of the chains configured for the event.

Report Logon Event

Configure the settings of this event to log in to the Advanced Authentication Reporting portal. For more information about the Reporting portal, see [Chapter 9, "Reporting," on page 119](#).

Windows Logon Event

Configure the settings of this event to log in to the Windows Client.

Creating a Customized Event

You can create customized events for the following.

- ♦ Third-party integrations.
- ♦ When you must use Windows Client or Linux PAM Client, or Mac OS X Client on both the domain joined and non-domain workstations and you must have a separate event to use the non-domain mode.
- ♦ For integrations using SAML 2.0 and OAuth 2.0.

You can create the following types of customized events:

- ♦ [Generic](#)
- ♦ [OS Logon \(domain\)](#)
- ♦ [OAuth2](#)
- ♦ [SAML2](#)

Creating a Generic Event

You can create a generic event for Windows Client, Mac OS X Client, and Linux PAM Client workstation when these clients are not joined or bound to a domain.

Perform the following steps to create a generic event:

- 1 Click **Events > Add**.
- 2 Specify a name for the event.
- 3 Set **Is enabled** to **ON**.
- 4 Select **Generic** in the **Event type**.
- 5 Select the **Authenticator category**. The **Authenticator category** option is displayed only if you have added categories in the “**Event Categories**” policy.
- 6 Select the chains that you want to assign to the current event.
- 7 If you want to restrict access of some endpoints to the event, add all the endpoints that must have access to the **Endpoint whitelist**. The remaining endpoints are blacklisted automatically. If you leave the **Endpoints whitelist** blank, all the endpoints will be considered for authentication.
- 8 Set **Geo-fencing** to **ON** to enable geo-fencing. Move the permitted zones from **Available** to **Used**. For more information about configuring geo-fencing, see the [Smartphone](#) method.

IMPORTANT: You must enable the [Geo Fencing Options](#) policy to use the geo fencing functionality.

- 9 Click **Save**.

NOTE: When you create a custom event, you must specify the custom event in the configuration file of the related endpoints. For more information, see the [Advanced Authentication - Linux PAM Client](#), [Advanced Authentication - Mac OS X Client](#), or [Advanced Authentication - Windows Client](#) guides related to the specific endpoint.

Creating an OS Logon (Domain) Event

You can create this event when the third-party application needs to read password of a user after authentication. For example, when Windows Client, Mac OS X Client, or Linux PAM Client workstation is joined or bound to a domain, the third-party application must read the password of the user.

The steps to create an **OS Logon (domain)** event are similar to the [Generic](#) event.

Creating an OAuth 2.0 Event

You can create this event for third-party integrations with OAuth 2.0.

To create an **OAuth 2** event, perform the following steps:

- 1 Click **Events > Add**.
- 2 Specify a name for the event.
- 3 Set **Is enabled** to **ON**.
- 4 Select **OAuth2** in the **Event type**.
- 5 Select the **Authenticator category**. The **Authenticator category** option is displayed only if you have added categories in the “[Event Categories](#)” policy.
- 6 Select the chains that you want to assign to the current event.
- 7 Specify the **Redirect URIs**. The **Client ID** and **Client secret** are generated automatically. The **Client ID**, **Client secret**, and **Redirect URI** are consumed by the consumer web application. After successful authentication, the redirect URI web page specified in the event is displayed.
- 8 In **Advanced Settings**, perform the following actions:
 - ◆ Set the **Use for Owner Password Credentials** option to **ON**, if the consumer web application provides authorization in the form of Resource Owner Password Credentials Grant.
 - ◆ Set the option to **OFF**, if the consumer web application provides authorization in the form of Authorization Code Grant or Implicit Grant.

NOTE: If option is set to **ON**, you can use only the **LDAP Password only** chain for this event. It is recommended to use separate events for Resource Owner Password Credentials Grant (**Use for Owner Password Credentials > ON**) and Authorization Code Grant / Implicit Grant (**Use for Owner Password Credentials > OFF**).

- 9 Click **Save**.

After you have created an **OAuth 2** event, perform the following steps to access the consumer web application:

- 1 Specify the **Client ID**, **Client secret**, and **redirect URIs** in the consumer web application.
- 2 Specify the appliance end point (authorization end point) in the web application. For example, `https://<Appliance IP>/osp/a/TOP/auth/oauth2/grant`.
- 3 Authenticate with the required authentication method(s) to access the consumer web application.

NOTE: Authorization is provided in the form of Authorization Code Grant or Implicit Grant or Resource Owner Password Credentials Grant.

Creating a SAML 2.0 Event

You can create this event for third-party integrations with SAML 2.0.

- 1 Click **Events > Add**.
- 2 Specify a name for the event.
- 3 Set **Is enabled** to **ON**.
- 4 Select **SAML 2** in the **Event type**.
- 5 Select the **Authenticator category**. The **Authenticator category** option is displayed only if you have added categories in the “**Event Categories**” policy.
- 6 Select the chains that you want to assign to the current event.
- 7 If you want to restrict access of some endpoints to the event, add all the endpoints that must have access to the **Endpoint whitelist**. The remaining endpoints are blacklisted automatically. If you leave the **Endpoints whitelist** blank, all the endpoints will be considered for authentication.
- 8 Set **Geo-fencing** to **ON** to enable geo-fencing. Move the permitted zones from **Available** to **Used**. For more information about configuring geo-fencing, see the **Smartphone** method.

IMPORTANT: You must enable the **Geo Fencing Options** policy to use the geo fencing functionality.

- 9 You can either insert your Service Provider's SAML 2.0 metadata in **SP SAML 2.0 metadata** or click **Browse** and select a Service Provider's SAML 2.0 metadata XML file to upload it.

NOTE: You must enable the **SAML 2.0 options** policy for the SAML 2.0 event to work appropriately.

- 10 Click **Save**.

Managing Endpoints

Endpoints are devices where the Advanced Authentication server authenticates. An endpoint can be a Windows workstation for Windows Client endpoint, or Advanced Authentication Access Manager appliance for the NAM endpoint and so on.

The endpoints are automatically added when you install a plug-in such as NAM or install Windows Client. The RADIUS endpoint, an OSP endpoint that is used for WebAuth authentication, and Endpoint41 and Endpoint42 are the predefined endpoints.

NOTE: Endpoint41 and Endpoint42 are created for the integration with legacy NAM and NCA plug-ins, which are used in NAM 4.2 and earlier versions with Advanced Authentication 5.1.

The NAM and NCA plug-ins work with the hard coded endpoint ID and secret. In Advanced Authentication 5.2 and later, you must register the endpoints. This breaks the backward compatibility with old plug-ins. These two legacy endpoints allow to keep the old plug-ins working.

To configure an endpoint for Advanced Authentication, perform the following steps:

- 1 In the **Endpoints** section, click **Edit** against the endpoint you want to edit.
- 2 You can rename the endpoint, change its description or endpoint type.
- 3 Set **Is enabled** to **ON** to enable the endpoint.

- 4 Set **Is trusted** to **ON** if the endpoint is trusted. In some integrations such as Migration Tool, Password Filter, NAM, and NCA you must enable the **Is trusted** option for their endpoints.
- 5 Specify an **Endpoint Owner** if you have configured a specific chain to be used by the Endpoint owner only. This is a user account that must be able to use a different **chain** than the other users for authentication.

The Endpoint Owner feature is supported for Windows Client, Mac OS Client, and Linux PAM Client only.

NOTE: Additional information such as **Operating System**, **Software** version, **Last session** time and **Device** information are displayed. Also in **Advanced properties**, RAM information is displayed.

The **Last session** time is updated only when the **Delete old endpoint device and update endpoint last session option** is set to **ON** in the **Replica options** policy.

Advanced Authentication Windows Client 5.6 or newer must be installed on the endpoint.

- 6 Click **Save**.

You can create an endpoint manually. This endpoint can be used for the third-party applications that do not create endpoints.

To create an endpoint manually, perform the following steps:

- 1 In the **Endpoints** section, click **Add**.
- 2 On the **Add endpoint** page, specify a **Name** of the endpoint and its **Description**.
- 3 Set the **Type** to **Other**.
- 4 Set **Is enabled** to **ON**.
- 5 Leave **Endpoint Owner** blank.
- 6 Click **Save**. The **New Endpoint secret** window is displayed.
- 7 Take down the values specified in **Endpoint ID** and **Endpoint Secret** and place them in a secure place in your application.

NOTE: You will not be able to get the **Endpoint ID** and **Endpoint Secret** later on the appliance.

- 8 Click **OK**.

NOTE: **Tenancy settings** are not supported for Endpoints.

IMPORTANT: You must ensure not to remove an endpoint that has at least one component running on it such as Windows Client, Logon Filter, RD Gateway plug-in, or ADFS plug-in. Endpoint is removed automatically when you uninstall Windows Client. However you must remove the endpoint manually when you uninstall Logon Filter, RD Gateway plug-in or ADFS plug-in.

If you remove an endpoint accidentally, ensure to remove the records with prefix **endpoint*** from the `%ProgramData%\NetIQ\Windows Client\config.properties` file and re-start the machine. This recreates the endpoint.

Configuring Policies

Policies contain configuration settings for the Advanced Authentication methods, events, and so on. For example, to use the **Email OTP** method, you must configure the server and port settings in the **Mail sender** policy and to use the Multitenancy mode, you must enable the **Multitenancy options** policy.

Advanced Authentication provides the following policies:

- ◆ [Admin UI Whitelist Policy](#)
- ◆ [Authenticator management options](#)
- ◆ [Cache options](#)
- ◆ [Delete me options](#)
- ◆ [Endpoint management options](#)
- ◆ [Event categories](#)
- ◆ [Geo fencing options](#)
- ◆ [Helpdesk Options](#)
- ◆ [Last Logon Tracking Options](#)
- ◆ [Lockout Options](#)
- ◆ [Login Options](#)
- ◆ [Logon Filter for AD](#)
- ◆ [Mail sender](#)
- ◆ [Password Filter for AD](#)
- ◆ [SAML 2.0 options](#)
- ◆ [SMS sender](#)
- ◆ [Services Director Options](#)
- ◆ [Voice sender](#)

To configure a policy, perform the following steps:

- 1 Click **Policies** in the Administration portal.
- 2 Click the **Edit** icon  against the policy you want to configure.
- 3 Click **Save**.

IMPORTANT: The configured policies are applied for all the Advanced Authentication servers.

Admin UI Whitelist Policy

In this policy, you can configure the security settings to allow only permitted IP addresses to use the Advanced Authentication Administration portal.

By default, all the IP addresses are considered as whitelist. To configure a restriction so that only a particular IP address can access the Administration portal, perform the following steps:

- 1 Click **Add** in the **Admin UI whitelist** policy.
- 2 Specify the address in the format `10.20.30.0/255.255.255.0` or `10.20.30.0/24`.

Advanced Authentication has an automatic validation check to prevent administrators from losing access to the Administration portal. If your IP address is out of the range, a message: `Your IP address is not whitelisted. You will lose access! Please add your IP is displayed.`

- 3 Click **Save**.

Authenticator Management Options Policy

This policy allows you to configure the following two settings:

- ◆ **Enable sharing**: This setting allows a user to authenticate with his or her authenticator to another user's account. The helpdesk administrator can link an authenticator of one user to another user.

To enable sharing authenticators, set **Enable sharing** to **ON**.

NOTE: Linked authenticators work only in the online mode. Cached login does not work for the linked authenticators.

The supported methods for sharing authenticators are TOTP, HOTP, Password, Fingerprint, Card, and FIDO U2F.

- ◆ **Disable re-enrollment**: This setting allows you to restrict users from re-enrolling, editing, and deleting the enrolled authenticators in the Self-Service portal.

To disable re-enrollment or removal of authenticators, set **Disable re-enrollment** to **ON**.

WARNING: If you access the Administration portal with local user credentials such as `local\admin`, you might get into a lockout situation. This can happen when the administrator's password expires and it is not possible to change the password through the Self-Service portal. Therefore, to use the **Disable re-enrollment** option, you must configure the access of a repository account to the Administration portal. To do this:

- ◆ Add authorized users or a group of users from a repository to the **FULL ADMINS** role.
- ◆ Assign chains, which contain methods that are enrolled for users, to the **AdminUI** event (at a minimum with an LDAP Password method).

NOTE: This setting disables re-enrollment and removal of the authenticators only in the Self-Service portal. The setting has no effect on the Helpdesk portal.

Cache Options Policy

In this policy, you can disable the local caching of authenticators. The policy is supported for Windows Client, Mac OS X Client, and Linux PAM Client for chains that use the methods: LDAP Password, Password, HOTP, TOTP, Smartphone (offline mode), Card, FIDO U2F, Fingerprint, and PKI.

The caching functionality enables the storing of credentials on the Client for offline authentication when the Advanced Authentication server is not available. Therefore a user who has successfully logged in once to the server with the authentication, can now login with the offline authentication.

By default, the **Enable local caching** option is enabled. To disable the caching, set the option to **OFF** and click **Save**.

NOTE: To cache Fingerprint data, you need to install Microsoft.NET Framework 4 or higher on your workstation. The caching period cannot be configured. The cache will be cleared only if the **Enable local caching** option is disabled.

Delete Me Options

In this policy, you can configure settings that enable deleting all the user data from the server, including the enrolled methods.

When you set **Enable delete me feature** to **ON**, the **Delete me** option is displayed in a drop-down on the user name on the top-right corner of the Self-Service portal.

Endpoint Management Options

In this policy, you can configure settings for endpoint management.

Set **Require admin password to register endpoint/workstation** to **ON** for endpoints to provide the local administrator's credentials during the registration of endpoint.

You must disable the option when installing any components from the Advanced Authentication distributives package that uses endpoints (Advanced Authentication Windows Client, Mac OS X Client, Linux PAM Client, Logon Filter, and RDG plug-in). Otherwise, the endpoints are not created. You must use the option for third-party integrations only.

Event Categories

In this policy you can add categories, which can be used in an event to support multiple enrollments for a method. For each event, you can specify one category.

To add a category, perform the following steps:

- 1 Click **Event categories**.
- 2 Click **Add**.
- 3 Specify a name and description for the category.
- 4 Click **Save**.
- 5 Click **Events** and edit the required event to specify the category.

Ensure that users or helpdesk administrators enroll authenticators for the new category.

NOTE:

- ♦ You can enroll only one authenticator of one type for each category.
 - ♦ The **Authenticator category** option in **Events** is not displayed when no category is created.
 - ♦ The LDAP Password method is an exception. There is one LDAP password authenticator always, it can be used with any category.
-

Geo Fencing Options

In this policy, you can create authentication zones by drawing boundaries for a geographical location. When you enable the geo-fencing policy, users can authenticate with their Smartphones only from the allowed geographical locations.

To enable geo-fencing, set **Enable geo fencing** to **ON**. For more information about how to configure the geo-zones, see the [“Smartphone”](#) method.

NOTE: When you enable the **Geo-fencing options** policy, the functioning of the TOTP mode of the Smartphone method, which is used in the offline mode, is affected. An error message `TOTP login is disabled` is displayed to the users when they try to authenticate with this method.

Helpdesk Options

In this policy, you can configure settings to prompt the helpdesk administrators to provide the credentials of the users in the Helpdesk portal. This enhances security. This policy is applicable to the [Helpdesk User](#) event.

Set **Ask credentials of management user** to **ON** to prompt the helpdesk administrator to provide the credentials of the user in the Helpdesk portal. Ensure that you have specified a chain (with all the methods of the chain enrolled for the users) for the [Helpdesk User](#) event.

When you set the option to **OFF**, it may not be secure, but the user management is done faster.

Last Logon Tracking Options

The **Last Logon Tracking options** policy allows Advanced Authentication to enable tracking for the last login. This policy helps you to automatically move to a simple chain that contains less factors within a few hours of authentication done with a high-security chain.

For example, if a user authenticates with the `LDAP Password+Card` chain once in a day, the user can further use only the `Card` method without the `LDAP Password` method, or if a user authenticates with the `Fingerprint` method once in every four hours, the user can authenticate once with this chain and next authentication he can use only the `Smartphone` chain.

To enable tracking, set **Enable tracking option** to **ON**. You must enable this policy for the [Require chain](#) option while creating a chain.

To configure a high-security chain and the corresponding simple chain, see [Creating Chains](#).

Lockout Options

In this policy, you can configure settings to lock a user’s account when the user reaches the maximum failure attempts of login. This enhances security by preventing the guessing of passwords and one-time passwords (OTPs).

You can configure the following options in this policy:

- ♦ **Enable:** An option to enable the lockout settings.
- ♦ **Failed attempts:** The limit of failure attempts of authentication, after which the user’s account is locked. The default value is 3.
- ♦ **Lockout period:** The period within which the user’s account is locked and the user cannot authenticate. The default value is 300 seconds.

- ♦ **Lock in repository:** The option to lock the user account in repository. You cannot use **Lockout period** if you enable this option. Only the system administrator must unlock the user in the repository.

IMPORTANT: You must configure the appropriate settings in your repository for the options to function appropriately. For Active Directory Domain Services, you must enable the [Account lockout threshold policy](#) on Domain Controllers.

For NetIQ eDirectory, you must configure the [Intruder Detection](#) properly.

After a user's account is locked (not in the repository), you can unlock the user account. To do this, click **Repositories > Edit > Locked Users** and click **Remove** against the user's account name.

Login Options

In this policy, you can configure the settings to add repositories that are used as default repositories. Therefore while logging in, you need not prefix the repository name before the username for authentication.

For example, if `pjones` is a member of the company repository, then while logging in, instead of specifying `company\pjones`, you can specify only `pjones`.

To add a repository as default, move the repository from **Available** to **Default** and click **Save**.

Logon Filter for AD

In this policy you can configure settings to enable the use of Logon Filter that you must install on all the Domain Controllers in the domain and configure it. Logon Filter allows you to automatically update group membership if you login with the Advanced Authentication Windows Client.

To enable the policy, set **Enable filter** to **ON** and click **Save**.

NOTE: Before enabling the policy, you must ensure the Advanced Authentication Logon Filter is installed on all the Domain Controllers in the domain. Else, you might face problems with password validation during password synchronization on workstations that have the Windows Client installed.

For information about how to configure Logon Filter, see [Configuring Logon Filter](#).

Mail Sender

In the **Mail sender** policy, you can configure settings for the following:

- ♦ **Email OTP** method to send email messages with one-time passwords to users.
- ♦ To send the replication conflict notification email to configured users.

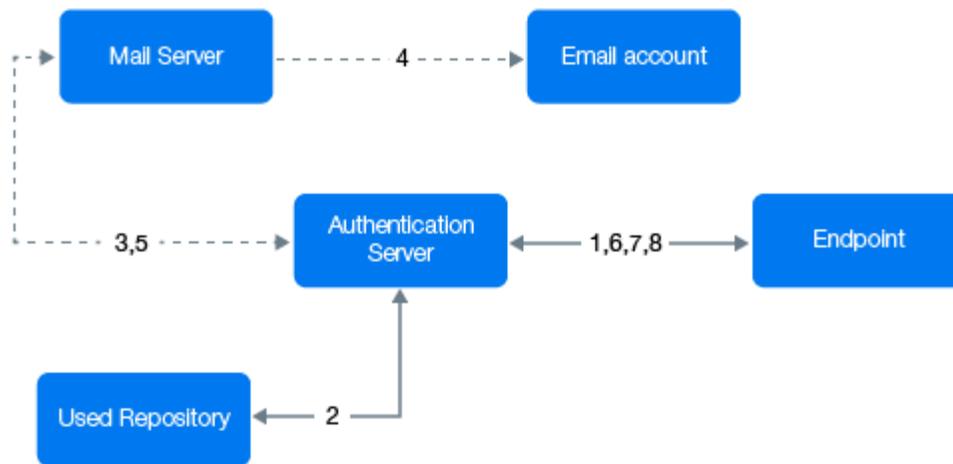
To configure the **Mail sender** settings, perform the following steps:

- 1 Specify the following details:
 1. **Host:** The outgoing mail server name. For example, `smtp.company.com`.
 2. **Port:** The port number. For example, 465.
 3. **Username:** The username of an account that is used to send the authentication or notification mail. For example, `noreply` or `noreply@company.com`.
 4. **Password:** The password for the specified account.

5. **Sender email:** An email address of an account that is used to send the authentication or notification mail.
 6. **TLS and SSL:** The cryptographic protocol used by the mail server.
 - 2 You can test the configurations for the Mail sender policy in the **Test** section.
 - 2a Specify the email address in **E-mail** to which you want to send the Email OTP.
 - 2b Specify a message to be sent to the phone in **Message**.
 - 2c Click **Send test message!**.
 - 3 Click **Save**.
- Real messaging uses async sender. Ensure that you have configured a chain with the **Email OTP** method and assigned it to an event. Login to the Self-Service portal and test the Email authenticator. If it does not work, click **async log**.

Authentication Flow

The authentication flow for the Mail sender is described in the following image.



A user wants to authenticate on an endpoint such as a laptop or a website with the **Email OTP** method. The following steps describe the authentication flow:

- 1 When the authentication request is initiated, the endpoint contacts the Advanced Authentication server.
- 2 The Advanced Authentication server validates the user's credentials and gets an email address of the user from a repository.
- 3 Advanced Authentication server sends the request to a configured mail server to send an email message with the content that includes a one-time password (OTP) for authentication.
- 4 Mail server sends the message to the user's email address.
- 5 Mail server sends the sent signal to the Advanced Authentication server.
- 6 Advanced Authentication server sends a request to the user to specify an OTP on the endpoint.
- 7 The user specifies the OTP from the email message. The Advanced Authentication server gets the OTP.
- 8 Advanced Authentication server validates the authentication. The authentication is done or denied.

HTTPS protocol is used for the internal communication.

Access configuration

Advanced Authentication server - Mail Server (SMTP, outbound).

Password Filter for AD

In this policy, you can configure settings to synchronize the password update between the appliance and Active Directory through the Password Filter. The Password Filter automatically updates the LDAP Password stored in Advanced Authentication, whenever the password is changed or reset in the Active Directory. This helps you to authenticate without getting any prompt to synchronize the password after it is changed or reset.

You can perform the following settings in this policy:

- ◆ Set **Update password on change** to **ON** to update the LDAP password automatically in Advanced Authentication when it is changed in the Active Directory. This helps you to authenticate without getting a prompt to synchronize the password after it is changed.

Set **Update password on change** to **OFF** to prompt the user to synchronize the LDAP password while logging in to Windows when the password is changed in the Active Directory.

- ◆ Set **Update password on reset** to **ON** to update the LDAP password automatically in Advanced Authentication when it is reset in the Active Directory. This helps users to authenticate without getting a prompt to synchronize the password if it is reset.

Set **Update password on reset** to **OFF** to prompt the user to synchronize the LDAP password while logging in to Windows when the user's password has been reset in the Active Directory.

NOTE: Endpoint for the Password Filter must be trusted. To do this, perform the following steps:

- 1 Click **Endpoints** in the Advanced Authentication Administration portal.
 - 2 Edit an endpoint of the Password Filter.
 - 3 Set **Is trusted** to **ON** and add a description.
 - 4 Save the changes.
-

SMS Sender

In this policy, you can configure the settings for the **SMS OTP** method. The **SMS OTP** method sends SMS messages with one-time passwords to the users. Advanced Authentication contains predefined settings for Twilio and MessageBird services.

The **Sender Service** consists of the following three options:

- ◆ **Generic**
- ◆ **Twilio**
- ◆ **MessageBird**

To configure SMS sender manually perform the following steps:

- 1 Select **Generic** in **Sender service**.
- 2 Specify a **Service URL** value. For example, Clickatell `http://api.clickatell.com/http/sendmsg?`.
- 3 Leave **HTTP Basic Auth Username** and **HTTP Basic Auth Password** blank.
- 4 Select **POST** from **HTTP request method**.

5 Click **Add** and create the following parameters in **HTTP request body**.

- ◆ name: **user**
value: name of your account
- ◆ name: **to**
value: {phone}
- ◆ name: **text**
value: {message}
- ◆ name: **api_id**, this is a parameter that is issued after addition of an HTTP sub-product to your Clickatell account. A single account may have multiple API IDs associated with it.
- ◆ name: **from**
value: sender's phone number

6 Click **Add secure** and create the following parameter in **HTTP request body**.

- ◆ name: **password**
value: current password that is set on the account

For more information about the additional parameters for Clickatell, see the [Clickatell documentation](#).

NOTE: The parameters may differ for different SMS service providers. But the {phone} and {message} variables are mandatory.

To configure SMS sender settings for **Twilio** service, perform the following steps:

1 Select **Twilio** in **Sender service**.

2 Specify the following details:

- ◆ **Account sid** and **Auth token**: In Twilio, the Account SID acts as a username and the Auth Token acts as a password.
- ◆ **Use Copilot**: The copilot option is used to send SMS from a Twilio's phone number of your location. This is helpful when SMS messages have to be sent across the geographical locations. For example, with copilot, SMS will be sent from Indian phone number to the Indian users. Without copilot, SMS will be sent from US phone number to the Indian users.

For more information on Copilot option and its features, see <https://www.twilio.com/copilot#phone-number-intelligence> and <https://www.twilio.com/docs/api/rest/sending-messages-copilot#features>.

- ◆ **Messaging Service SID**: Service SID.
- ◆ **Sender phone**: Sender's phone number.

For more information, see the [Twilio website](#).

To configure SMS sender settings for **MessageBird** service, perform the following steps:

1 Select **MessageBird** in **Sender service**.

2 Specify the **Username**, **Password**, and **Sender name**.

For more information, see the [MessageBird website](#).

IMPORTANT: MessageBird API v2 is not supported. To activate MessageBird API v1, perform the following steps:

- 1 Go to the MessageBird account.
 - 2 Click **Developers** in the left navigation bar and open the [API access](#) tab.
 - 3 Click **Do you want to use one of our old API's (MessageBird V1, Mollie or Lumata)? Click here.**
-

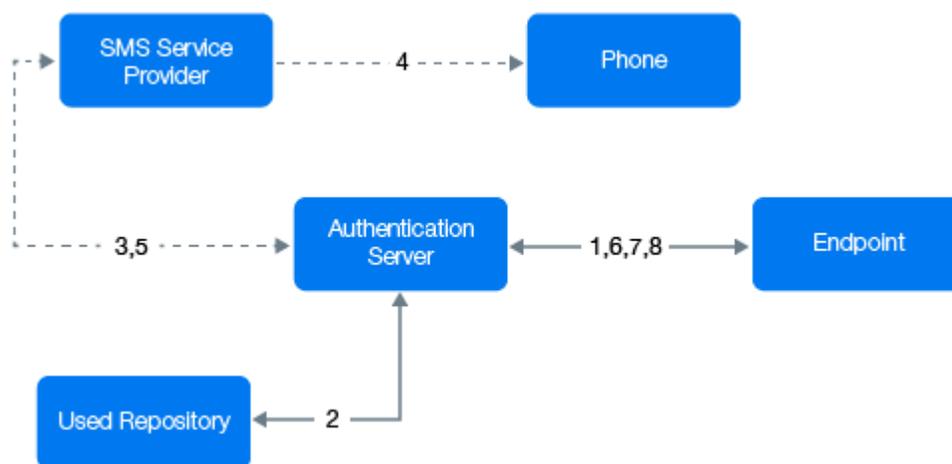
You can test the configurations for the SMS sender policy in the **Test** section.

- 1 Specify the phone number in **Phone** to which you want to send the SMS OTP.
- 2 Specify a message to be sent to the phone in **Message**.
- 3 Click **Send test message!**.
- 4 Click **Save**.

Real messaging uses async sender. Ensure that you have configured a chain with the **SMS** method and assigned it to an event. Then sign-in to the Self-Service portal and test the SMS authenticator. If it does not work, see the **async** logs.

Authentication Flow

The authentication flow for the SMS sender in Advanced Authentication is described in the following image.



A user wants to authenticate on an endpoint such as a laptop or a website with the SMS method. The following steps describe the authentication flow:

- 1 When the authentication request is initiated, the endpoint contacts the Advanced Authentication server.
- 2 The Advanced Authentication server validates the user's credentials and gets a phone number of the user from a Repository.
- 3 Advanced Authentication server sends the request to a configured SMS Service Provider to send an SMS message with the content that includes a one-time password (OTP) for authentication.
- 4 SMS Service Provider sends the SMS message to the user's phone.
- 5 SMS Service Provider sends the 'sent' signal to the Advanced Authentication server.
- 6 Advanced Authentication server sends a request to the user to specify an OTP on the endpoint.

- 7 The user specifies the OTP from the SMS message. The Advanced Authentication server gets the OTP.
- 8 Advanced Authentication server then validates the authentication. The authentication is done or denied.

HTTP/HTTPS protocol is used for the communication.

Access configuration

Advanced Authentication server - SMS Service Provider (HTTP/HTTPS, outbound).

Services Director Options

In this policy, you can configure settings required to integrate with the Services Director.

Perform the following steps to configure this policy:

- 1 Set **Enable integration** to **ON** to enable the integration of Advanced Authentication with Services Director.
- 2 Specify the **Public DNS name** of Advanced Authentication, **Services Director DNS Name**, **Tenant Admin Name**, and **Tenant Admin Password** of Services Director to integrate it with Advanced Authentication.

SAML 2.0 Options

In this policy, you can configure settings to specify the Identity Provider's URL and to download the SAML 2.0 metadata file. The downloaded SAML 2.0 metadata file is used to configure the service provider.

For more information about configuring this policy, see ["SAML 2.0"](#).

For information about how to configure Advanced Authentication integration with Salesforce using SAML 2.0, see ["Configuring Integration with Salesforce"](#).

Voice Sender

In this policy, you can configure the settings for the [Voice](#) and [Voice OTP](#) methods. Advanced Authentication supports the Twilio service for the Voice methods.

To configure Voice Sender settings for [Twilio](#) service, perform the following steps.

- 1 Specify the following details in the **Voice sender** policy:
 - ♦ **Account sid** and **Auth token**: In Twilio, the Account SID acts as a username, and the Auth Token acts as a password.
 - ♦ **Sender phone**: The phone number of the sender.
 - ♦ **Public server url**: The public URL to which the Twilio service connects for authentication. You can use http protocol for testing purpose, but for production environment you must use https protocol. You must have a valid certificate when you use https.

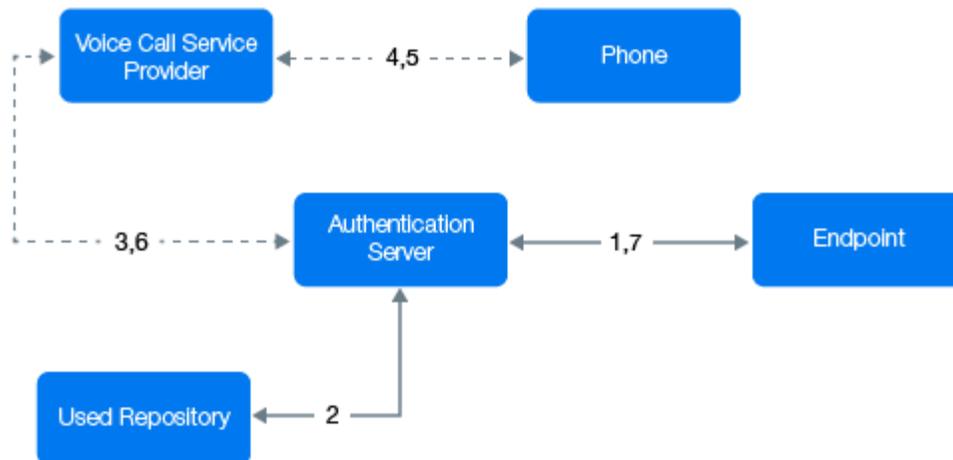
- 2 In the **Enroll without a phone** section, you can configure settings for the user to enroll the Voice authenticator without a phone number in the repository.
 - ♦ Set **Allow user enrollment without a phone** to **OFF** to ensure that a user does not enroll the Voice authenticator without a phone. The user gets an error message that you can specify in **Error message**.
 - ♦ Set **Allow user enrollment without a phone** to **ON** for the user to enroll the Voice authenticator without a phone.
- 3 You can test the configurations for the Voice sender policy in the **Test** section.
 - 3a Specify the phone number in **Phone** to which you want to send the Voice OTP.
 - 3b Specify a message to be sent to the phone in **Message**.
 - 3c Click **Send test message!**.
- 4 Click **Save**.

Real messaging uses async sender. Ensure that you have configured a chain with the **Voice OTP** method and assigned it to an event. Then sign-in to the Self-Service portal and test the Voice authenticator. If it does not work, see the **async** logs.

IMPORTANT: The users may receive calls with the voice `Application error`. This happens because of incorrect settings or invalid certificates. Ensure that the certificate is valid and is not expired. Invalid certificates cannot be applied by Twilio.

Authentication Flow

The authentication flow for the Voice sender in Advanced Authentication is described in the following image.



A user wants to authenticate on an endpoint such as a laptop or a website with the **Voice Call** method. The following steps describe the authentication flow:

- 1 When the authentication request is initiated, the endpoint contacts the Advanced Authentication server.
- 2 The Advanced Authentication server validates the user's credentials and gets a phone number of the user from a repository.
- 3 Advanced Authentication server sends the request to a configured voice call service provider (Twilio) to call the user.

- 4 The voice call service provider calls the user.
- 5 The user picks up the phone, listens to the call, and specifies the PIN followed by the hash (#) sign.
- 6 Voice call provider sends the specified PIN to the Advanced Authentication server.
- 7 Advanced Authentication server then validates the authentication. The authentication is done or denied.

HTTP/HTTPS protocol is used for the communication.

Access configuration

Advanced Authentication server - Voice Call Service Provider (HTTP/HTTPS, inbound/ outbound).

Adding a License

To add a license for Advanced Authentication, perform the following steps:

1. Click **Licenses**.
2. Click **Add**.
3. Click **Browse** and select the valid license.
4. Click **Upload** to upload the license.

A user license is consumed when a user enrolls at least one authenticator through an automatic enrollment, enrollment by a Helpdesk administrator, or self-enrollment. This is an exception for the LDAP password, as a license is not consumed for it. An automatic enrollment is done only when a user performs a first authentication.

TIP: To free up a user's license, perform the following steps:

- 1 Exclude the user from a group that is assigned to chains.
 - 2 Click **Repositories** and edit a repository.
 - 3 Click **Full sync** to perform a full synchronization of the repository.
The existing user's authenticators are removed.
-

Exporting the Database

IMPORTANT: The Advanced Authentication upgrade is not supported from Advanced Authentication 5.6 to upcoming 6.0 version. Therefore, it is required to export the database, install, and configure 6.0 version, and import the database in the appliance.

Advanced Authentication facilitates you to export the entire database to .cpt format. In this way, you can create backup of the database or migrate the database to Advanced Authentication 6.0 version. The exported database includes configuration of the following sections:

- ◆ Dashboard
- ◆ Repositories
- ◆ Methods
- ◆ Chains

- ◆ Events
- ◆ Endpoints
- ◆ Policies
- ◆ Logs
- ◆ Licenses
- ◆ Tenant database
- ◆ Server Options
 - ◆ Web server SSL certificate for HTTPS
 - ◆ Login page background
- ◆ Enrollment
 - ◆ Enrolled Authenticators
 - ◆ Shared Authenticators
 - ◆ Emergency Passwords

NOTE: The exported database does not include configuration of the following sections:

- ◆ Web Authentication
- ◆ Debug logs
- ◆ Cluster configuration in Global Master server
- ◆ Updates.

To export the database, perform the following steps:

- 1 Click **Export** in the Administration console.
- 2 Click **Export Database**.

The exported database file is saved in the `.cpt` format on your local drive.

NOTE: You must host the exported `.cpt` file in the FTP or HTTP server to import the database on to Advanced Authentication 6.0 appliance. Ensure to provide the local administrator password to decrypt the imported file.

NOTE: The Tenant administrators cannot export the database.

4 Enrolling the Authentication Methods

Advanced Authentication server supports the following ways to enroll the authentication methods:

- ♦ **Automatic enrollment:** This type of enrollment is used for the **SMS, Email, RADIUS, LDAP Password, and Swisscom Mobile ID** methods.

The methods are enrolled automatically if the chains containing them are assigned to any event.

- ♦ **Enrollment by Administrator:** This type of enrollment is used for the **OATH Tokens**.

An administrator can import tokens from the PSKC or CSV files in the **Administration portal > Methods > OATH OTP > OATH Tokens** tab. You can assign tokens to the specific users.

- ♦ **Enrollment by Helpdesk administrator:** This type of enrollment is used by the Helpdesk administrator.

A Helpdesk administrator can access the Helpdesk portal with the address: `https://<NetIQ Server>/helpdesk`. In the Helpdesk portal, the Helpdesk administrator can enroll the authentication methods for users. A Helpdesk administrator must be a member of the **Enroll Admins** group (**Repositories > Local > Edit > Global Roles**) to manage users' authenticators.

- ♦ **Enrollment by User:** This method is applicable for the users. A user can access the Self-Service portal with the address: `https://<NetIQ Server>/account`, where the users can enroll any of the authentication methods.



Configuring Integrations

Advanced Authentication facilitates clients to integrate with the third-party solutions using the following interface.

- ◆ [OAuth 2.0](#)
- ◆ [RADIUS Server](#)
- ◆ [SAML 2.0](#)
- ◆ [REST API](#)

The information about configuring Advanced Authentication with some of the third-party solutions is as follows:

- ◆ [Configuring Integration with Barracuda](#)
- ◆ [Configuring Integration with Citrix NetScaler](#)
- ◆ [Configuring Integration with Dell SonicWall SRA EX-Virtual Appliance](#)
- ◆ [Configuring Integration with FortiGate](#)
- ◆ [Configuring Integration with OpenVPN](#)
- ◆ [Configuring Integration with Salesforce](#)
- ◆ [Configuring Integration with ADFS](#)

5 OAuth 2.0

In OAuth 2.0 authorization, the third-party client requests access to the resources that are controlled by the resource owner. Instead of using the resource owner's credentials to access the protected resources, the third-party client obtains an access token. The third-party clients can be web applications, mobile phones, handheld devices, and desktop applications.

This section contains the following topics:

- ♦ “Building Blocks of OAuth 2.0” on page 85
- ♦ “Sample OAuth 2.0 Application Integrated with Advanced Authentication” on page 88
- ♦ “OAuth 2.0 Attributes” on page 93
- ♦ “Non Standard Endpoints” on page 94

Building Blocks of OAuth 2.0

The following are the building blocks of OAuth 2.0.

- ♦ [OAuth 2.0 Roles](#)
- ♦ [OAuth 2.0 Grants](#)

OAuth 2.0 Roles

OAuth 2.0 consists of the following four roles:

- ♦ **Resource Owner:** Entity that grants access to a protected resource. It can be a system or a person (end-user) owning the resources.
- ♦ **Resource Server:** Server that hosts the protected resources. It accepts and responds to the protected resource requests using the access tokens.
- ♦ **Client:** Application that requests and get authorization on behalf of the resource owner to access a protected resource.
- ♦ **Authorization Server:** Server that issues access tokens to the client after the successful authentication of the resource owner and obtaining authorization.

OAuth 2.0 Grants

By default, Advanced Authentication supports the following OAuth 2.0 grant types. However, if you require to use the **Resource owner password credential** grant, you have to enable it using Advanced Authentication settings. For more information on OAuth 2.0 grant types, see the [link \(https://tools.ietf.org/html/rfc6749\)](https://tools.ietf.org/html/rfc6749).

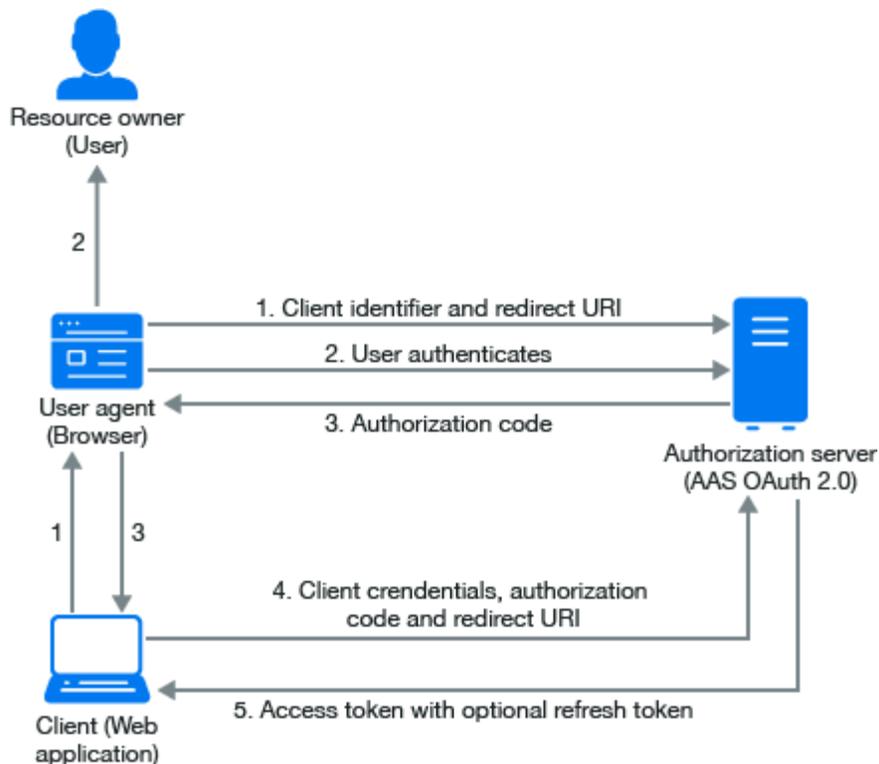
- ♦ “Authorization Code” on page 86
- ♦ “Implicit Grant” on page 87

Authorization Code

In authorization code, an authorization server acts as an intermediary between the client and the resource owner. Instead of requesting authorization directly from the resource owner, the client directs the resource owner to an authorization server, which in turn directs the resource owner back to the client with the authorization code.

The authorization grant type depends on the method used by the application to request authorization, and the grant types supported by the API.

The following diagram describes the workflow of authorization code grant.



The workflow for authorization code includes the following steps:

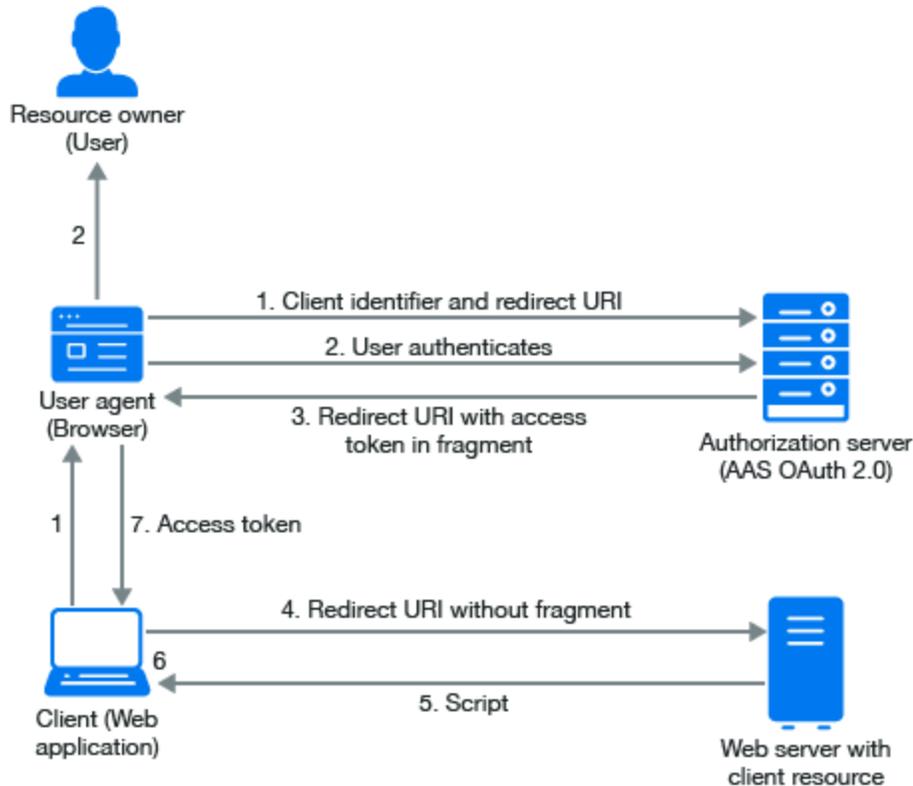
1. The OAuth client initiates the flow when it directs the user agent of the resource owner to the authorization endpoint. The OAuth client includes its client identifier, requested scope, local state, and a redirection URI.
2. The authorization server authenticates the resource owner through the user agent and recognizes whether the resource owner grants or denies the access request.
3. If the resource owner grants access, the OAuth client uses the redirection URI provided earlier to redirect the user agent back to the OAuth client. The redirection URI includes an authorization code and any local state previously provided by the OAuth client.
4. The OAuth client requests an access token from the authorization server through the token endpoint. The OAuth client authenticates with its client credentials and includes the authorization code received in the previous step. The OAuth client also includes the redirection URI used to obtain the authorization code for verification.
5. The authorization server validates the client credentials and the authorization code. The server also ensures that the redirection URI received matches the URI used to redirect the client in Step 3. If valid, the authorization server responds back with an access token.

Implicit Grant

The implicit grant is similar to the authorization code grant with two distinct differences.

- ♦ It is used for user-agent-based clients. For example, single page web apps that cannot keep a client secret because all the application code and storage is easily accessible.
- ♦ Secondly, instead of the authorization server returning an authorization code which is exchanged for an access token, the authorization server returns an access token.

The following diagram describes the workflow of Implicit grant.



The workflow for implicit grant includes the following steps:

1. The OAuth client initiates the flow by directing the user agent of the resource owner to the authorization endpoint. The OAuth client includes its client identifier, requested scope, local state, and a redirection URI. The authorization server sends the user agent back to the redirection URI after access is granted or denied.
2. The authorization server authenticates the resource owner through the user agent and verifies whether the resource owner grants or denies the access request.
3. If the resource owner grants access, the authorization server redirects the user agent back to the client using the redirection URI provided earlier. The redirection URI includes the access token in the URI fragment.
4. The user agent follows the redirection instructions by making a request to the web server without the fragment. The user agent retains the fragment information locally.

5. The web server returns a web page, which is typically an HTML document with an embedded script. The web page accesses the full redirection URI including the fragment retained by the user agent. It can also extract the access token and other parameters contained in the fragment.
6. The user agent runs the script provided by the web server locally, which extracts the access token and passes it to the client.

Sample OAuth 2.0 Application Integrated with Advanced Authentication

To create a sample web application, you need Python v3 (the sample script prepared on v3.4.3).

The following web application describes the functionalities supported when Advanced Authentication is integrated with OAuth 2.0. OAuth 2.0 server is an authorization and resource server. As an Authorization Server, the OAuth server can prompt the users to go through authentication chains and as a resource server, the OAuth server can prompt the users to provide user details.

You must create the following five files:

1. Sample script (oauth2_test.py)

```

from bottle import Bottle, request, run, redirect, SimpleTemplate, template
from urllib.parse import urlparse, urlunparse, urlencode, quote
import urllib.request
import base64
import ssl
import json

app = Bottle()

client_id = 'id-rSCzuBLQgXCATfkXZ4fsedAo8sPsWxSs'
client_secret = 'secret-91DpzWFD26RriURR7KJ1pryFx7V9QeDm'
redirect_uri = 'http://localhost:8088/' # this app callback URI
authorization_endpoint = 'https://192.168.0.151/osp/a/TOP/auth/oauth2/grant'
attributes_endpoint = 'https://192.168.0.151/osp/a/TOP/auth/oauth2/getattributes'
state = {}

@app.get('/getattr')
def get_attributes():
    params = urlencode({
        'attributes': 'client username userRepository user_dn user_cn mail sid
upn netbiosName',
        'access_token': state['access_token']
    })
    url = attributes_endpoint + '?' + params
    print('getattr url: {}'.format(url))
    req = urllib.request.Request(url)
    gcontext = ssl.SSLContext(ssl.PROTOCOL_TLSv1_2) # avoid cert checking
    with urllib.request.urlopen(req, context=gcontext) as response: # perform
GET request and read response
        rsp = response.read()
        attributes = json.loads(rsp.decode('utf-8'))
        return template('attributes.html', items=attributes.items(),
refresh_token=urllib.parse.quote(state['refresh_token']))

@app.get('/')

```

```

def do_get():
    code = request.query.get('code')
    if code:
        # got code from OAuth 2 authentication server
        token = get_token_code(code)
        state.update(token)
        return template('token.html', items=token.items(),
refresh_token=urllib.parse.quote(token['refresh_token']))
    else:
        return template('main.html')

@app.get('/logon')
def do_logon():
    pr=list(urlparse(authorization_endpoint))
    # set query
    pr[4]=urlencode({
        'response_type': 'code',
        'client_id': client_id,
        'redirect_uri': redirect_uri
    })
    # perform redirection to OAuth 2 authentication server
    redirect(urlunparse(pr))

@app.get('/logon-implicit')
def do_logon_implicit():
    # parse authorization_endpoint URL
    pr = list(urlparse(authorization_endpoint))
    # set query
    pr[4] = urlencode({
        'response_type': 'token',
        'client_id': client_id,
    })
    # perform redirection to OAuth 2 authentication server
    redirect(urlunparse(pr))

@app.get('/logon-creds')
def do_logon_creds():
    return template('logonform.html')

@app.post('/logon-creds')
def do_logon_creds_post():
    username = request.forms.get('username')
    password = request.forms.get('password')
    token = get_token_password(username, password)
    state.update(token)
    return template('token.html', items=token.items(),
refresh_token=urllib.parse.quote(token['refresh_token']))

def get_token_password(username, password):
    # prepare POST parameters - encode them to urlencoded
    data = urlencode({
        'grant_type': 'password',
        'username': username,
        'password': password
    })
    data = data.encode('ascii') # data should be bytes
    resp_text = post_data(data, prepare_headers())
    print(resp_text)
    return json.loads(resp_text)

```

```

@app.get('/refresh')
def do_refresh():
    token = refresh_access_token(request.query.get('refresh_token'))
    state.update(token)
    return template('token.html', items=token.items(),
refresh_token=state.get('refresh_token', ''))

def get_token_code(code):
    # prepare POST parameters - encode them to urlencoded
    data = urlencode({
        'grant_type': 'authorization_code',
        'code': code,
        'redirect_uri': redirect_uri
    })
    data = data.encode('ascii') # data should be bytes
    resp_text = post_data(data, prepare_headers())
    print(resp_text)
    return json.loads(resp_text)

def refresh_access_token(refresh_token):
    print('refresh_token: {}'.format(refresh_token))
    # prepare POST parameters - encode them to urlencoded
    data = urlencode({
        'grant_type': 'refresh_token',
        'refresh_token': refresh_token,
    })
    data = data.encode('ascii') # data should be bytes
    resp_text = post_data(data, prepare_headers())
    print(resp_text)
    return json.loads(resp_text)

def prepare_headers(use_content_type_hdr = True):
    hdrs = {
        'Authorization': 'Basic {}'.format(base64.b64encode(
            '{}:{}'.format(quote(client_id, safe=''), quote(client_secret,
safe='')).encode('ascii')).decode(
            'ascii')),
    }
    if use_content_type_hdr:
        hdrs.update({'Content-type': 'application/x-www-form-urlencoded'})
    return hdrs

def post_data(data, headers):
    print('post_data\nheaders:\n{}\nndata:\n{}'.format(headers, data))
    req = urllib.request.Request(authorization_endpoint, data, headers)
    gcontext = ssl.SSLContext(ssl.PROTOCOL_TLSv1_2) # avoid cert checking
    with urllib.request.urlopen(req, context=gcontext) as response: # perform
POST request and read response
        rsp = response.read()
    return rsp.decode('utf-8')

run(app, host='0.0.0.0', port=8088)

```

NOTE: In the script, you must change the values for `client_id`, `client_secret`, and Advanced Authentication server address in `authorization_endpoint` and `attributes_endpoint` (lines 10-14).

2. Main menu (main.html)

```
<!DOCTYPE html>
<html>
<head lang="en">
  <meta charset="UTF-8">
  <title></title>
  <script type="text/javascript">
    //
      function getHashParam(name) {
        var hash = window.location.hash;
        if (hash) {
          if (name = (new RegExp('[#&amp;]' + encodeURIComponent(name) +
            '=[^&amp;]*'))).exec(hash))
            return decodeURIComponent(name[1]);
        }
      }
      function showResult() {
        if (window.location.hash) {
          document.getElementById('result').innerHTML = '&lt;table
border="1"&gt;'+
            '&lt;tr&gt;&lt;td&gt;access_token&lt;/
td&gt;&lt;td&gt;'+getHashParam('access_token')+'&lt;/td&gt;&lt;/tr&gt;'+
            '&lt;tr&gt;&lt;td&gt;token_type&lt;/
td&gt;&lt;td&gt;'+getHashParam('token_type')+'&lt;/td&gt;&lt;/tr&gt;'+
            '&lt;tr&gt;&lt;td&gt;expires_in&lt;/
td&gt;&lt;td&gt;'+getHashParam('expires_in')+'&lt;/td&gt;&lt;/tr&gt;'+
            '&lt;/table&gt;';
        } else {
          document.getElementById('result').innerHTML = 'Implicit
granted token is not found';
        }
      }
    // ]]&gt;
  &lt;/script&gt;
&lt;/head&gt;
&lt;body onload="showResult();"&gt;
&lt;div id="result"&gt;result&lt;/div&gt;&lt;br/&gt;
&lt;br/&gt;
Click &lt;a href="/logon"&gt;here&lt;/a&gt; to obtain an authentication token through
Authorization Code Grant&lt;br/&gt;
Click &lt;a href="/logon-implicit"&gt;here&lt;/a&gt; to obtain an authentication token
through Implicit Grant (the token will be received in hash part of THIS
page)&lt;br/&gt;
Click &lt;a href="/logon-creds"&gt;here&lt;/a&gt; to obtain an authentication token through
Resource Owner Password Credentials Grant&lt;br/&gt;
&lt;/body&gt;
&lt;/html&gt;</pre></div><div data-bbox="179 739 447 755" data-label="Section-Header"><h2>3. Token information (token.html)</h2></div><div data-bbox="770 936 916 953" data-label="Page-Footer"><p>OAuth 2.0 91</p></div>
```

```

<!DOCTYPE html>
<html>
<head lang="en">
  <meta charset="UTF-8">
  <title></title>
</head>
<body>
Token<br/>
<table border="1">
  % for k, v in items:
  <tr>
    <td>{{k}}</td>
    <td>{{v}}</td>
  </tr>
  % end
</table>
<br/>
<a href="/getattr">Get attributes</a><br/>
<a href="/refresh?refresh_token={{refresh_token}}">Refresh token</a>
</body>
</html>

```

4. Attributes information (attributes.html)

```

<!DOCTYPE html>
<html>
<head lang="en">
  <meta charset="UTF-8">
  <title></title>
</head>
<body>
Attributes<br/>
<table border="1">
  % for k, v in items:
  <tr>
    <td>{{k}}</td>
    <td>{{v}}</td>
  </tr>
  % end
</table>
<br/>
<a href="/refresh?refresh_token={{refresh_token}}">Refresh token</a>
</body>
</html>

```

5. Logon form for Resource Owner Password Credentials Grant mode (logonform.html)

```

<!DOCTYPE html>
<html>
<head lang="en">
  <meta charset="UTF-8">
  <title></title>
</head>
<body>
<form method="post" action="/logon-creds">
  User name: <input type="text" name="username"><br/>
  Password: <input type="password" name="password"><br/>
  <input type="submit">
</form>
</body>
</html>

```

Running the Sample Web Application

Perform the following steps to run the sample web application.

- 1 Run the script `python oauth2_test.py`.
- 2 Open the URL `http://localhost:8088`.

A message is displayed with the following modes:

```
Authorization Code Grant
Implicit Grant (the token will be received in hash part of THIS page)
Resource Owner Password Credentials Grant (is not supported by default but it
can be activated in AAF)
```

- 3 Select the grant based on your requirement.

- ◆ **Authorization Code Grant**

1. Ensure that **Use for Owner Password Credentials** is set to **OFF** in the **Advanced settings** section for the OAuth 2.0 event.

2. Click the first link.

The NetIQ Access page is displayed with the user name request.

3. Specify the **Username**.

4. Click **Next**.

5. Authenticate using all required methods of the chain.

The result page shows the `access_token`, `token_type` and `expires_in`.

- ◆ Click **Get attributes** to look at the attributes.
- ◆ Click **Refresh token** to refresh token. The `access_token` value is updated.

- ◆ **Implicit Grant**

1. Ensure that **Use for Owner Password Credentials** is set to **OFF** in the **Advanced settings** section for the OAuth 2.0 event.

2. Click the first link.

The NetIQ Access page is displayed with the user name request.

3. Specify the **Username**.

4. Click **Next**.

5. Authenticate using all the required methods of the chain.

The result page shows the `access_token`, `token_type` and `expires_in`.

- ◆ **Resource Owner Password Credentials Grant**

1. Open **Advanced settings** for the OAuth 2.0 event.

2. Set **Use for Owner Password Credentials** to **ON**.

3. Click the third link.

A request for Username and Password is displayed.

4. Specify the username and password, then click **Submit**.

The result page displays the `access_token`, `token_type`, and `expires_in`.

OAuth 2.0 Attributes

The following table displays the OAuth 2.0 attributes for a test user from the Active Directory.

Attribute	Value
user_name	pjones
repository_name	TESTCOMPANY
naafUserSID	S-1-5-21-3320677580-2179873152-1514081409-1103
naafUserDN	CN=Paul Jones,CN=Users,DC=testcompany,DC=local
naafUserCN	Paul Jones
naafUserUPN	pjones@testcompany.local
naafUsernameNetBIOS	TESTCOMPANY\pjones
client	id-0TRljvJEe3qKwJiXvy3IbjvcixfiiY1Q
naafUserEmail	pjones@testcompany.com

The following table displays the OAuth 2.0 attributes for a local user.

Attribute	Value
user_name	ADMIN
repository_name	LOCAL
client	id-0TRljvJEe3qKwJiXvy3IbjvcixfiiY1Q

The `client` attribute is a **Client ID** specified in the [OAuth 2.0 settings](#).

Non Standard Endpoints

OSP provides a non-standard OAuth 2.0 endpoint for signing additional data that can be passed during the grant request. The URL of the sign endpoint is: `https://<serverip>/osp/a/TOP/auth/oauth2/sign`.

The sign endpoint helps to create a signed and encrypted data packet that can be used to supply data to other endpoints. For more information, see the `Sign` class documentation.

The only endpoint with which the signed data is currently used is the grant endpoint when it is used with the authorization code grant and implicit grant types.

The signed data can be used to supply one or both of the following:

- ◆ **Username:** Supplying the username for a client application is useful when you already know the username.
- ◆ **Advanced Authentication chain:** An Advanced Authentication server (5.6 or later) can be used to supply one or more additional authentication factors by authenticating with Advanced Authentication OAuth 2.0 for a user who is already authenticated. The username and name of the desired authentication chain containing the factor(s) is supplied.

You must be able to resolve username in an Advanced Authentication repository and you must configure the chain in the Advanced Authentication event for the OAuth 2.0 client used.

Submitting the Data

The sign endpoint is used by submitting a string value to the endpoint. The output is returned in a JSON structure. The output can be used with the grant endpoint with the **parameters** attribute.

You can accomplish OAuth 2.0 client authentication with HTTP **Basic** or **Bearer** authorization header value.

Request parameters

- ♦ **data** (required): The data to be signed and encrypted.

See `OAuth2Constants.OAUTHX_REQUEST_PARAM_DATA`.

- ♦ **ttl** (optional): The time-to-live period of the result data in milliseconds. If no value is supplied, then the default value of 30 seconds is used.

6 RADIUS Server

The Advanced Authentication server provides a built-in RADIUS server that can authenticate any RADIUS client using one of the chains configured for the event.

IMPORTANT

- ♦ The built-in RADIUS server supports only the PAP method.
 - ♦ The RADIUS server supports all authentication methods except **Card**, **FIDO U2F**, **Notaris ID**, **Fingerprint**, and **PKI** methods.
 - ♦ By design, Advanced Authentication does not support the single-factor authentication with a **Smartphone** method for RADIUS. It is recommended to use it in a two-factor chain with the **LDAP Password** method.
-

To configure an authentication event for RADIUS, perform the following steps:

- 1 Click **Events**.
 - 2 Click **Edit** next to the **Radius Server** event.
 - 3 Ensure that **Is enabled** is set to **ON**.
 - 4 Select the chains that you want to assign to the event.
 - 5 Select **Radius** from **Endpoint whitelist**.
 - 6 Click **Add** to add and assign a RADIUS Client to the event:
 - 6a Specify the IP address of the RADIUS Client in **IP Address**.
 - 6b Specify the RADIUS Client name in **Name**.
 - 6c Specify the RADIUS Client secret and confirm the secret.
 - 6d Ensure that the RADIUS Client is set to **ON**.
 - 6e Click  next to the RADIUS Client.
 - 6f Add more RADIUS Clients if required.
- 7 Set **Return user groups** to **ON** to enable the RADIUS server to return all the groups of a user in the `filter-id` attribute in an authentication response to the RADIUS Client. To enable the RADIUS server to send only specific groups of a user in place of all the groups of a user in the `filter-id` attribute, specify the particular user groups in **User groups white list**. For example, `Bob\mydomain`.

By default the option is set to **OFF** and the RADIUS server does not return the `filter-id` attribute in the authentication response.

If you set the option to **ON** and the **User groups white list** is empty, all the groups of a user are returned in the `filter-id` attribute.

NOTE: It is recommended to enable the **Return user groups** option and specify the particular user groups because in large environments a user can be part of many groups and as a result, the list of all groups that are returned by the RADIUS server can be large. The size of RADIUS response exceeds the maximum size of RADIUS packet.

- 8 Click **Save**.

IMPORTANT: If you use more than one chain with the RADIUS server, follow one of the following ways:

1. Each chain assigned to the RADIUS event may be assigned to a different LDAP group. For example, **LDAP Password+Smartphone** chain is assigned to a **Smartphone** users group, **LDAP Password+HOTP** chain is assigned to a HOTP users group. If a RADIUS user is a member of both groups, the top group is used.
2. By default, the top chain specified in the **Radius Server** event in which all the methods are enrolled is used. But, you can authenticate with the RADIUS authentication using another chain from the list when specifying `<username>&<chain shortname>` in **username**. For example, `pjones&sms`. Ensure that you have specified the short names for chains. Some RADIUS clients such as FortiGate do not support this option.

NOTE: If you use the **LDAP Password+Smartphone** chain, you can use an offline authentication by specifying the following the password in the `<LDAP Password>&<Smartphone OTP>` format. For example, `Q1w2e3r4&512385`. This option is supported for **LDAP Password+OATH TOTP**, **Password+Smartphone**, **Password+OATH TOTP**, **Password+OATH HOTP**.

Challenge-Response Authentication

If you have configured a multi-factor chain such as **LDAP Password&SMS OTP** or any other combination chain, some users (during the authentication) might not be able to specify the `<Password>&<OTP>` in a single line (because of the Password length limit in RADIUS). In this case, you can configure the existing RADIUS Client by performing the following steps:

1. Specify an LDAP password in **Password** and send the authentication request.

Advanced Authentication server returns the access-challenge response with `State=<some value>` (example: `State=WWKNNLTTBxP6QYfiZIpvscyt7RYrYsGag4h8s0Rh8R`) and `Reply-Message=SMS OTP`. You will receive an SMS with a one-time password on the registered mobile.

2. Specify the OTP in **Password** and add an additional RADIUS attribute with `State=<value>` where, value is the value that is obtained in step 1.
3. Send the authentication request.

Advanced Authentication stores the RADIUS event settings only on a server where the administrator performs the configuration (typically this is the DB Master server). After the DB Slave server is converted to DB Master server, the configuration may be lost. Goto the **Radius Server** event settings and click **Save** to apply the configuration.

The following are the examples of integration with a RADIUS Server:

- ♦ [Configuring Integration with Barracuda](#)
- ♦ [Configuring Integration with Citrix NetScaler](#)
- ♦ [Configuring Integration with Dell SonicWall SRA EX-Virtual Appliance](#)
- ♦ [Configuring Integration with FortiGate](#)
- ♦ [Configuring Integration with OpenVPN](#)

7 SAML 2.0

SAML 2.0 is an XML-based protocol that uses security tokens containing assertions. The assertions are used for sending the information about a subject (an entity that is often a human user) from a SAML authority (Identity Provider) to a SAML consumer (Service Provider).

To integrate Advanced Authentication with the third-party solutions using SAML 2.0, perform the following steps

- 1 Click **Server Options** in the Advanced Authentication Administration portal.
- 2 : Enable the **WebAuth** option and confirm the selection.

NOTE: You must enable the **WebAuth** option separately for each server where ever required.

- 3 Click **Events > Add**.
- 4 Specify a name for the new event.
- 5 Change the **Event type** to **SAML2**.
- 6 Select the required chains for the event.
- 7 (Conditional) If you require Geo-fencing, enable **Geo-fencing**.

NOTE: Geo-fencing can be enabled only for the Smartphone method.

- 8 Copy and paste your Service Provider's SAML 2.0 metadata to **SP SAML 2.0 metadata**.
OR
Click **Browse** and select a Service Provider's SAML 2.0 metadata XML file to upload it.
- 9 Click **Policies > SAML 2.0 options**.
- 10 (Conditional) Specify the Identity Provider's URL in **External URL**.

NOTE: To use multiple Advanced Authentication servers with SAML 2.0, you must do the following:

1. Enable the **WebAuth** option in the section for every Advanced Authentication server where this is required.
 2. Configure an external .
 3. Specify the address in **External URL** instead of specifying an address of a single Advanced Authentication server.
-
- 11 Click **Download IdP SAML 2.0 Metadata** to open a metadata. The metadata opens in a new browser page.
 - 12 Save the metadata (XML text) from the browser.
 - 13 (Conditional) If required, use the downloaded metadata file in your Service Provider.
 - 14 (Conditional) If required, use the Identity Provider certificate in your Service Provider.

```
-----BEGIN CERTIFICATE-----
MIIDkzCCAnugAwIBAgIESsmdMzANBgkqhkiG9w0BAQsFADB6MRAwDgYDVQQGEwdVbmtub3duMRAw
DgYDVQQQIEwdVbmtub3duMRAwDgYDVQQHEwdVbmtub3duMREwDwYDVQQKEwhBdXR0YXNhc3ESMBAG
A1UECXMjQXV0aGFzYXNhMRswGQYDVQQDEExJvc3AuYXV0aGFzYXMubG9jYWwwHhcNMTYwNTI2MDUz
NjI0WhcNMjYwNDA0MDUzNjI0WjB6MRAwDgYDVQQGEwdVbmtub3duMRAwDgYDVQQQIEwdVbmtub3du
MRAwDgYDVQQHEwdVbmtub3duMREwDwYDVQQKEwhBdXR0YXNhc3ESMBAGA1UECXMjQXV0aGFzYXNh
MRswGQYDVQQDEExJvc3AuYXV0aGFzYXMubG9jYWwwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEK
AoIBAQCw3YLz03qhSZPXjBc/Ws+cZ2/E5oogqKeJ3p4RR6USOoarjnmvQPq+maRfvexriwQjRDgS
OFRb58cert/misqzsHBVmQDnfMwicFVzuuKjDEbWFP9vLlgRkDzI1pCy13eNmBWuWXM49Z6mm8XS
fIwlAoydNp5DK0o0Yrk6FNOi0nOrnI5kHGVD0bd5SpDtvXSF1Wlfc5YT9UBUpfZneKsVPWskbeBX
F84hYJWBtdzcTEyjdso9Ra7UtxLIUW0UH3LWTgn9zS97nLkmhetmD1I3mEAEAE9SAmqTRyH1FNXZ
ZOfi/BJF4+sz86f6pBbwYM2KtVxAbgzSpZpJ1pQrZKPAgMBAAGjITAFMB0GA1UdDgQWBbTL8Pba
+e6YkBIk4yELTZ+Abfda6DANBgkqhkiG9w0BAQsFAAOCAQEAm87lNyAO8Ctn5jllLe3CupLAAbUWR
NY6av7LpPaillJRIw+uvddMyOz1vOS1IwpDDntcPtxGXsaZl1CKgNPBpLvSxepVUXNFgUCtu+bT
cuUtiQbkiDWWFLmAS6KeA+EBFOeqBiudEfkaZZT87DF9gKvM6VWdzJ7BvWi2YPbH/FRM82fLoyAd
RbphF215we3rvsfeWbwXw70UGNyBUTb3zUcAmB3sHbcZiXJZj3pJYgDaN9Ss60sz/yG1ZLEYlulL
R1T2PPEfEcA1Eij0R1A31Z5hJ3zDlXoCeNyLoMg4522QYekTwwQeWkeYeJBXEcxdl7VP6F91zmfZ
bm1A4PY5jw==
-----END CERTIFICATE-----
```

15 Change used hash to SHA-1 in your Service Provider, if the option is presented.

The following are the examples of integration with SAML 2.0.

- ◆ [Configuring Integration with Salesforce](#)
- ◆ [Configuring Integration with ADFS](#)

8 Examples of Integrations

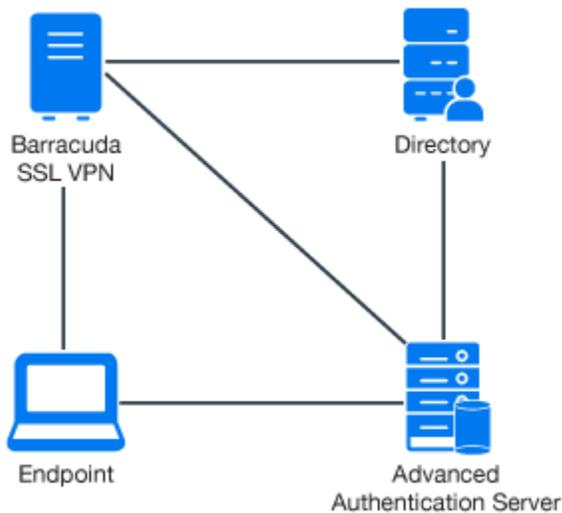
This chapter contains the following examples of third- party integrations.

- ◆ “Configuring Integration with Barracuda” on page 101
- ◆ “Configuring Integration with Citrix NetScaler” on page 103
- ◆ “Configuring Integration with Dell SonicWall SRA EX-Virtual Appliance” on page 105
- ◆ “Configuring Integration with FortiGate” on page 106
- ◆ “Configuring Integration with OpenVPN” on page 108
- ◆ “Configuring Integration with Palo Alto GlobalProtect Gateway” on page 109
- ◆ “Configuring Integration with Salesforce” on page 110
- ◆ “Configuring Integration with ADFS” on page 113

Configuring Integration with Barracuda

This section provides the configuration information on integrating Advanced Authentication with Barracuda SSL VPN virtual appliance. This integration secures the Barracuda SSL VPN connection.

The following diagram represents integration of Advanced Authentication with Barracuda SSL VPN.



To configure the Advanced Authentication integration with Barracuda SSL VPN, perform the following configuration tasks:

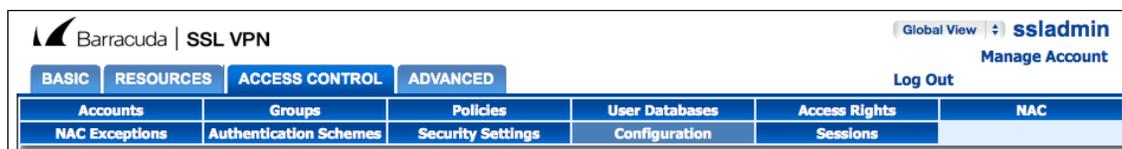
- ◆ “Configure the Advanced Authentication RADIUS server:” on page 102
- ◆ “Configure the Barracuda SSL VPN Appliance:” on page 102
- ◆ “Authenticate in Barracuda SSL VPN Using Advanced Authentication” on page 102

Configure the Advanced Authentication RADIUS server:

- 1 Open the Advanced Authentication Administration portal.
- 2 Click **Events > Radius Server**.
- 3 Set **Is enabled** to **ON**.
- 4 Move one or more chains from **Available** to **Used** list. Ensure that the chains are assigned to the appropriate group of users in **Roles & Groups** of the **Chains** section.
- 5 Click **Client > Add**.
- 6 Specify an **IP address** of the Barracuda SSL VPN appliance.
- 7 Specify a secret and confirm it.
- 8 Set **Enabled** to **ON**.
- 9 Click **Save** in **Client**.
- 10 Click **Save** in **Events**.

Configure the Barracuda SSL VPN Appliance:

- 1 Sign-in to the Barracuda SSL VPN Configuration portal as **ssladmin**.
- 2 Click **Access Control > Configuration**.



- 3 Scroll down to **RADIUS**.
- 4 Specify an Advanced Authentication appliance IP address in **RADIUS Server**.
- 5 Specify a shared secret in **Shared Secret**.
- 6 Set **Authentication Method** to **PAP**.
- 7 Set **Reject Challenge** to **No** to allow challenge response.
- 8 Click **Save Changes**.
- 9 Click **Access Control > User Databases**.
- 10 Create a user database using the same storage as you are using for Advanced Authentication.
- 11 Click **Access Control > Authentication Schemes**.
- 12 Click **Edit** for the **Password** scheme for the user database.
- 13 Move **RADIUS** from **Available modules** to **Selected modules**.
- 14 Remove the **Password** module from the **Selected modules**.
- 15 Apply the changes.

Authenticate in Barracuda SSL VPN Using Advanced Authentication

- 1 Specify the user's credentials.
- 2 Click **More** and select the configured user database (if the database is not selected by default).

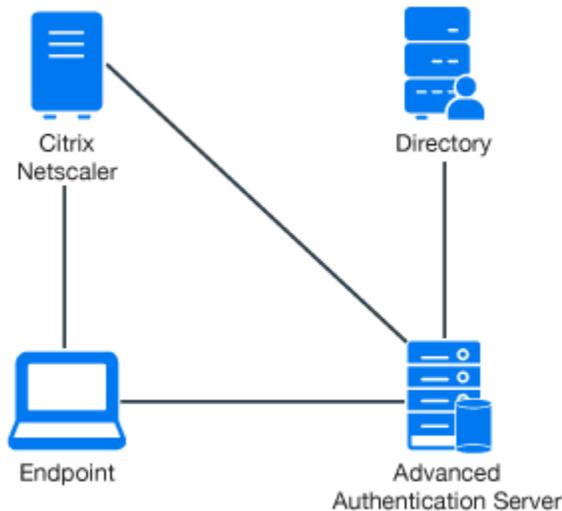
- 3 Click **Log In** and approve the authentication on the user's smartphone.

NOTE: Advanced Authentication can be configured with the other authentication chains.

Configuring Integration with Citrix NetScaler

This section provides the configuration information on integrating Advanced Authentication with Citrix NetScaler VPX. This integration secures the Citrix NetScaler VPX connection.

The following diagram represents Advanced Authentication in Citrix NetScaler.



To configure the Advanced Authentication integration with Citrix NetScaler VPX, perform the following configuration tasks:

- ♦ [“Configure the Advanced Authentication RADIUS Server” on page 103](#)
- ♦ [“Configure the Citrix NetScaler Appliance” on page 104](#)
- ♦ [“Authenticate in Citrix NetScaler Using Advanced Authentication” on page 104](#)

Ensure that the following requirements are met:

- ♦ Citrix NetScaler VPX (version NS11.0 has been used to prepare these instructions) is installed.
- ♦ Advanced Authentication 5 appliance is installed.

Configure the Advanced Authentication RADIUS Server

- 1 Open the Advanced Authentication Administration portal.
- 2 Click **Events > Radius Server**.
- 3 Set **Is enabled** to **ON**.
- 4 Move one or more chains from **Available** to **Used** list. Ensure that the chains are assigned to the appropriate group of users in **Roles & Groups** of the **Chains** section.
- 5 Click **Client > Add**.
- 6 Specify an **IP address** of the Citrix NetScaler appliance.

- 7 Specify a secret and confirm it.
- 8 Set **Enabled** to **ON**.
- 9 Click **Save** in **Client**.
- 10 Click **Save** in **Events**.

Configure the Citrix NetScaler Appliance

- 1 Sign-in to the Citrix NetScaler configuration portal as **nsroot**.
- 2 Click **Configuration > Authentication > Dashboard**.
- 3 Click **Add**.
- 4 Select **RADIUS** for **Choose Server Type**.
- 5 Specify **Name** of the Advanced Authentication server, **IP Address**, **Secret Key**, and **Confirm Secret Key**.
- 6 Change **Time-out (seconds)** to 120-180 seconds if you are using the Smartphone, SMS, Email or Voice methods.
- 7 Click **More** and ensure that **PAP** is selected in **Password Encoding**.
- 8 Click **Create**.

If the connection to the RADIUS server is valid, the **Up** status is displayed.

- 9 Click **Configuration > System > Authentication > RADIUS > Policy**.
- 10 Click **Add**.
- 11 Specify **Name** of the Authentication RADIUS Policy.
- 12 Select the created RADIUS server from **Server** and select **ns_true** from the **Saved Policy Expressions** list.
- 13 Click **Create**.
- 14 Select the created policy and click **Global Bindings**.
- 15 Click **Select Policy**.
- 16 Select the created policy.
- 17 Click **Bind**.
- 18 Click **Done**.

A check mark is displayed in the **Globally Bound** column.

Authenticate in Citrix NetScaler Using Advanced Authentication

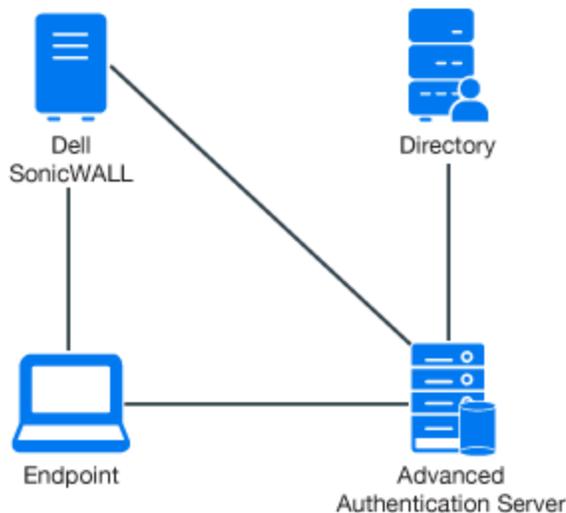
- 1 Specify the user's credentials then click **Login**.
- 2 Accept the authentication on your smartphone.

NOTE: Advanced Authentication can be configured with other authentication chains.

Configuring Integration with Dell SonicWall SRA EX-Virtual Appliance

This section provides the configuration information on integrating Advanced Authentication with Dell SonicWall SRA EX-virtual appliance. This integration secures the Dell SonicWall SRA connection.

The following diagram represents Advanced Authentication in Dell SonicWall.



To configure the Advanced Authentication integration with Dell SonicWall SRA, perform the following configuration tasks:

- ◆ [“Configure the Advanced Authentication RADIUS Server” on page 105](#)
- ◆ [“Configure the Dell SonicWall SRA Appliance” on page 106](#)
- ◆ [“Authenticate in Dell SonicWall Workspace Using Advanced Authentication” on page 106](#)

Ensure that the following requirements are met:

- ◆ Dell SonicWall SRA EX-Virtual appliance v11.2.0-258 is installed.
- ◆ Advanced Authentication v5 appliance is installed.

Configure the Advanced Authentication RADIUS Server

- 1 Open the Advanced Authentication Administration portal.
- 2 Click **Events > Radius Server**.
- 3 Set **Is enabled** to **ON**.
- 4 Move one or more chains from **Available** to **Used** list. Ensure that the chains are assigned to the appropriate group of users in **Roles & Groups** of the **Chains** section.
- 5 Click **Client > Add**.
- 6 Specify an **IP address** of the Dell SonicWall appliance.
- 7 Specify a secret and confirm it.
- 8 Set **Enabled** to **ON**.

- 9 Click **Save** in **Client**.
- 10 Click **Save** in **Events**.

Configure the Dell SonicWall SRA Appliance

1. Sign-in to the Dell SonicWall SRA Management console as **admin**.
2. Click **User Access > Realms**.
3. Click **New realm**.
4. Create a **New Authentication Server** and set the **Radius** authentication directory.
5. Set **Radius Server** and **Shared key**.
6. Save and apply the configuration.
7. Click **User Access > Realms**.
Review the realm diagram.

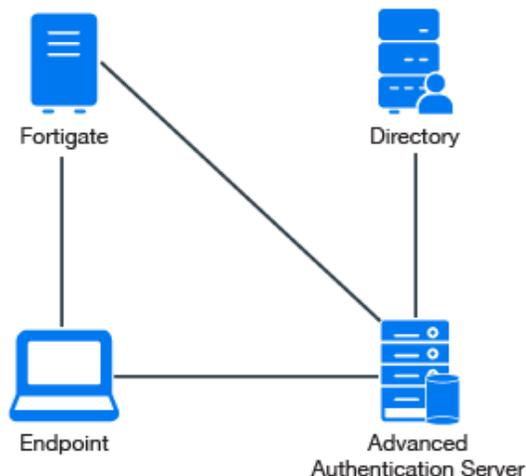
Authenticate in Dell SonicWall Workspace Using Advanced Authentication

- 1 Open a browser and navigate to the workplace.
- 2 Specify your username and LDAP password.
- 3 Specify the **SMS OTP** and click **OK**.

Configuring Integration with FortiGate

This section provides the configuration information on integrating Advanced Authentication with FortiGate. This integration secures the FortiGate connection.

The following diagram represents Advanced Authentication in FortiGate.



To configure the Advanced Authentication integration with FortiGate perform the following configuration tasks:

- ◆ [“Configure the Advanced Authentication RADIUS Server” on page 107](#)
- ◆ [“Configure the FortiGate Appliance” on page 107](#)
- ◆ [“Authenticate in FortiGate Using Advanced Authentication” on page 107](#)

Ensure that the following requirements are met:

- ◆ Fortinet virtual appliance v5 (Firmware version 5.2.5, build 8542 has been used to prepare these instructions) is installed.
- ◆ Advanced Authentication v5 appliance is installed.

Configure the Advanced Authentication RADIUS Server

- 1 Open the Advanced Authentication Administration portal.
- 2 Click **Events > Radius Server**.
- 3 Set **Is enabled** to **ON**.
- 4 Move one or more chains from **Available** to **Used** list. Ensure that the chains are assigned to the appropriate group of users in **Roles & Groups** of the **Chains** section.
- 5 Click **Client > Add**.
- 6 Specify an **IP address** of the FortiGate appliance.
- 7 Specify a secret and confirm it.
- 8 Set **Enabled** to **ON**.
- 9 Click **Save** in **Client**.
- 10 Click **Save** in **Events**.

Configure the FortiGate Appliance

1. Sign-in to FortiGate configuration portal as **admin**.
2. Check which **Virtual Domain** is bound to the network interface.
3. Open the RADIUS Server configuration for an appropriate **Virtual Domain** and setup the required settings.
4. Click **Test Connectivity** and specify the credentials of Advanced Authentication administrator to test the connection.
5. Create a user group and bind it to a remote authentication server.
6. Create user and place in the created group.

Authenticate in FortiGate Using Advanced Authentication

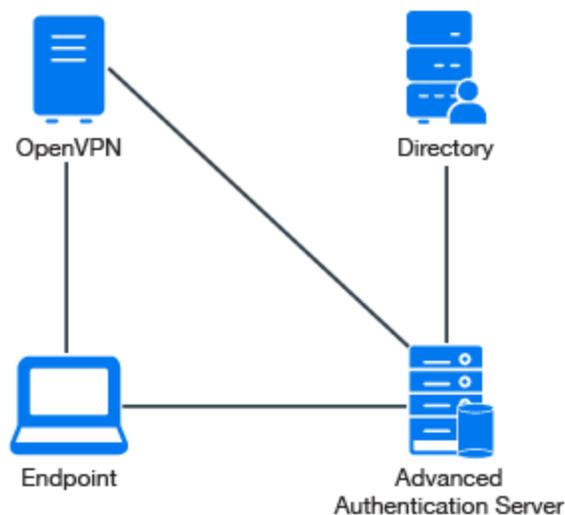
- 1 Specify the user’s credentials and click **Login**.
- 2 Specify the OTP and click **Login**.

NOTE: The **Token Code** field has a limitation of 16 digits. Therefore, you may face issues when using the YubiKey tokens with 18-20 digits code.

Configuring Integration with OpenVPN

This section provides the configuration information on integrating Advanced Authentication with OpenVPN virtual appliance. This integration secures the OpenVPN connection.

The following diagram represents Advanced Authentication in OpenVPN.



To configure the Advanced Authentication integration with OpenVPN perform the following configuration tasks:

- ◆ [“Configure the Advanced Authentication RADIUS Server” on page 108](#)
- ◆ [“Configure the OpenVPN Appliance” on page 109](#)

Ensure that the following requirements are met:

- ◆ OpenVPN v2 appliance (version 2.0.10 was used to prepare these instructions) is installed.
- ◆ Advanced Authentication v5 appliance with a configured repository is installed.

Configure the Advanced Authentication RADIUS Server

- 1 Open the Advanced Authentication Administration portal.
- 2 Click **Events > Radius Server**.
- 3 Set **Is enabled** to **ON**.
- 4 Move one or more chains from **Available** to **Used** list. Ensure that the chains are assigned to the appropriate group of users in **Roles & Groups** of the **Chains** section.
- 5 Click **Client > Add**.
- 6 Specify an **IP address** of the OpenVPN appliance.
- 7 Specify a secret and confirm it.
- 8 Set **Enabled** to **ON**.
- 9 Click **Save** in **Client**.
- 10 Click **Save** in **Events**.

Configure the OpenVPN Appliance

- 1 Open the **OpenVPN Access Server** site.
- 2 Click **Authentication > RADIUS**.
- 3 Enable the **RADIUS** authentication.
- 4 Select **PAP** authentication method.
- 5 Add an IP address of the Advanced Authentication v5 appliance and specify the secret.

You must specify the `<repository name>\<username>` or only `<username>`, if you have set the following configurations:

- ◆ You have selected a chain from the **Used** section in the **Radius Server** settings for connecting to OpenVPN.
- ◆ You have set the default repository name in **Policies > Login options** of the Advanced Authentication v5 appliance.

You must specify a **Short name** of the chain in the username after the `<username>` and space (you can specify the **Short name** in the **Chains** section of the Advanced Authentication v5 appliance), if you have set the following configurations:

- ◆ You have selected multiple chains from the **Used** section for connecting to OpenVPN.

NOTE: For some authentication methods, the correct time must be configured on the OpenVPN appliance. You can sync the time of the OpenVPN appliance using the following commands:

```
/etc/init.d/ntp stop  
  
/usr/sbin/ntpdate pool.ntp.org
```

User Account Locks After Three Successful Authentications with SMS AP to OpenVPN

Issue: While authenticating with the SMS method to connect to OpenVPN, after three successful authentications the user account is locked by OpenVPN.

Workaround: OpenVPN assumes each attempt of the challenge response (request of additional data in chain) as an error.

To resolve the issue, you must change the number of failures that can be accepted. For more information, see [Authentication failure lockout policy](#).

Configuring Integration with Palo Alto GlobalProtect Gateway

This section provides the configuration information on integrating Advanced Authentication with Palo Alto GlobalProtect Gateway. This integration secures the Palo Alto GlobalProtect Gateway connection.

NOTE: This configuration has been tested with PAN-OS 6.1.5 to 7.1.x and GlobalProtect 2.1x.

To configure the Advanced Authentication integration with Palo Alto GlobalProtect Gateway, perform the following configuration tasks:

- ◆ [“Adding the RADIUS Server” on page 110](#)
- ◆ [“Adding an Authentication Profile” on page 110](#)
- ◆ [“Configuring GlobalProtect Gateway” on page 110](#)

Adding the RADIUS Server

- 1 Log in to the Palo Alto administrative interface.
- 2 Click **Device > Server Profiles > RADIUS**.
- 3 Click **Add** to add a new RADIUS server profile.
- 4 Specify **NetIQ RADIUS** in **Name**.
- 5 Specify 30 in **Timeout**.
- 6 In the **Servers** section, click **Add** to add a RADIUS server and specify the following information:
 - ◆ **Profile Name**
 - ◆ Set **Timeout and Retries** in **Server Settings**
 - ◆ Details in the **Servers** section
- 7 Click **Add** and configure a connection to the RADIUS server built-in to the Advanced Authentication server.
- 8 Click **OK**.

Adding an Authentication Profile

- 1 Click **Device > Authentication Profile**.
- 2 Click **New** to add a new authentication profile.
- 3 Specify the Authentication Profile details such as the server type and user domain.

Configuring GlobalProtect Gateway

- 1 Click **Network > GlobalProtect > Gateways**.
- 2 Click on your configured GlobalProtect Gateway to open the properties window.
- 3 In the **Authentication** section of the **GlobalProtect Gateway General properties** tab, select the **NetIQ authentication profile** created in [Add an Authentication Profile](#) from the list.
- 4 Click **OK** to save the GlobalProtect Gateway settings.

Configuring Integration with Salesforce

This section provides the configuration information on integrating Advanced Authentication with Salesforce. This integration secures the Salesforce connection.

The following diagram represents Advanced Authentication in Salesforce.



To configure the Advanced Authentication integration with Salesforce, perform the following configuration tasks:

- ♦ “Configure Salesforce Domain Name” on page 111
- ♦ “Configure the SAML Provider” on page 111
- ♦ “Configure the Advanced Authentication SAML 2.0 Event” on page 112
- ♦ “Configuring to Authenticate on Salesforce with SAML 2.0” on page 113

Configure Salesforce Domain Name

- 1 Login to your Salesforce account.
- 2 Create a domain. If the domain is not created, then perform the following tasks:
 - 2a Click **Gear** and select **Setup Home** in the **Lightning Experience** interface.
 - 2b Scroll down the setup toolbar and navigate to **Company Settings**.
 - 2c Click **My Domain**.
 - 2d Specify your domain name and click **Save**.

The domain is activated. Use your domain name to open Salesforce. For example, <https://CompanyName.my.salesforce.com/>. SAML provider requires the domain name.

Configure the SAML Provider

- 1 Click **Settings > Identity > Single Sign-On Settings**.
- 2 Create a text file and add the following Identity Provider certificate to the file.

```

-----BEGIN CERTIFICATE-----
MIIDkzCCAnugAwIBAgIESsmdMzANBgkqhkiG9w0BAQsFADB6MRAwDgYDVQQGEwdVbmtub3duMRAw
DgYDVQQQIEwdVbmtub3duMRAwDgYDVQQHEwdVbmtub3duMREwDwYDVQQKEWhBdXRoYXNhc3ESMBAG
A1UECXMJQXV0aGFzYXNhc3ESMBAGA1UECXMJQXV0aGFzYXNhc3ESMBAGA1UECXMJQXV0aGFzYXNhc3ES
NjI0WhcNMjYwNDA0MDUzNjI0WjB6MRAwDgYDVQQGEwdVbmtub3duMRAwDgYDVQQQIEwdVbmtub3du
MRAwDgYDVQQHEwdVbmtub3duMREwDwYDVQQKEWhBdXRoYXNhc3ESMBAGA1UECXMJQXV0aGFzYXNhc3ES
MRswGQYDVQQDEdXjvc3AuYXV0aGFzYXNhc3ESMBAG9jYXNhc3ESMBAG9jYXNhc3ESMBAG9jYXNhc3ES
AoIBAQCw3YLz03qhSZPXjBc/Ws+cZ2/E5oogqKeJ3p4RR6USOoarjnmvQPq+maRfvexriwQjRDgS
OFRb58cert/misqzsHBVmqDnfMwicFVzUuKjDEbWFP9vLlgRkDzIlpCy13eNmBWuWXM49Z6mm8XS
fIw1AoydNp5DK0o0Yrk6FNOi0nOrnI5kHGVD0bd5SpDt vXSf1Wlfc5YT9UBUpfZneKsVPWskbeBX
F84hYJWBtdzctEYjds09Ra7UtXLIUW0UH3LWTgn9zS97nLkmhetmD1I3mEAeAE9SAmqTRyH1FNXZ
ZOfi/BJF4+sz86f6pBbwYM2KtVXaABgzSpZpJlpQrZKPAgMBAAGjITAfMB0GA1UdDgQWBbTL8PbA
+e6YkBIk4yELTZ+AbfdA6DANBgkqhkiG9w0BAQsFAAOCAQEAm87lNyA08CtN5jllLe3CupLAAbUWR
NY6av7LpPaillJRlW+uvddMy0zlvOS1IwpDDntcPtxGXsaZi1CKgNPBpLvSxePVUXnfFgUctu+bt
cuUtiQbkiDwWFLmAS6KeA+EBFOeqBiudEfkaZZT87DF9gKvM6VWdzJ7BvWi2YPbH/FRM82fLoyAd
RbphF215we3rvsfewBwXw70UGNyBUTb3zUcAmB3ShbcZiXJZj3pJYgDaN9Ss60sz/yG1ZLEYlUvL
R1T2PPEfEcA1Eij0R1A31Z5hJ3zDlXoCeNyLoMg4522QYekTwvQeWkeYeJBXEcxdL7VP6F91zmfZ
bmlA4PY5jw==
-----END CERTIFICATE-----
  
```

- 3 In **Single Sign-On Settings**, click **New** and specify the following details:
 1. **Name:** Advanced Authentication.
 2. **API Name:** AAF.
 3. **Issuer:** `https://AdvancedAuthenticationServerAddress/osp/a/TOP/auth/saml2/metadata`, where you must replace `AdvancedAuthenticationServerAddress` with the domain name or IP address of your Advanced Authentication server.
 4. **Entity ID:** `https://CompanyName.my.salesforce.com/`.
 5. Click **Choose File** to open the Identity Provider certificate.
 6. **SAML Identity Type:** Select **Assertion contains the Federation ID from the User object**.
 7. **SAML Identity Location:** Select **Identity is in an Attribute element**.
 8. **Attribute Name:** upn.
 9. **Service Provider Initiated Request Binding:** Select **HTTP Redirect**.
 10. **Identity Provider Login URL:** `https://AdvancedAuthenticationServerAddress/osp/a/TOP/auth/saml2/sso`.
 11. Select **User Provisioning Enabled**.
 12. Click **Save**.
- 4 Click **Edit** for Federated Single Sign-On Using SAML.
- 5 Select **SAML Enabled**.
- 6 Click **Save**.
- 7 Click **Settings > Users**.
- 8 Click **Edit** for the required Salesforce users by adding **Federation ID** for the user accounts. The Federation ID corresponds to `userPrincipalName` attribute in Active Directory. For example, `pjones@company.com`.

NOTE: The name that you specify in **Federation ID** is case sensitive. The following error may occur, if you ignore the case:

We can't log you in. Check for an invalid assertion in the SAML Assertion Validator (available in Single-Sign On Settings) or check the login history for failed logins.

- 9 Click your profile icon and click **Switch to Salesforce Classic**.

This mode is required to tune the domain options.
- 10 Click **Setup Administrator > Domain Management > My Domain > Edit** to access the **Authentication Configuration** screen.
- 11 Select **Login Page** and **osp** options.
- 12 Click **Save**.

Configure the Advanced Authentication SAML 2.0 Event

- 1 Click **username > Switch to Lightning Experience**.
- 2 Click **Gear** and select **Setup Home**.
- 3 Navigate to **Identity > Single Sign-On Settings**.
- 4 Click the created configuration (not for Edit).
- 5 Click **Download Metadata**.

- 6 Open the Advanced Authentication Administration portal.
- 7 Click **Events > Add** to add a new event.
- 8 Create an event with the following parameters.
 - ◆ Name: Salesforce
 - ◆ Chains: select the required chains.
 - ◆ Click **Browse** to Upload SP SAML 2.0 metadata file. Open the Salesforce metadata file and click **Save**.

Configuring to Authenticate on Salesforce with SAML 2.0

- 1 Click **Policies > SAML 2.0 options**.
- 2 Set **External URL** to `https://AdvancedAuthenticationServerAddress/` and replace `AdvancedAuthenticationServerAddress` with domain name or IP address of your Advanced Authentication server.

NOTE: To use multiple Advanced Authentication servers with SAML 2.0, you must do the following:

1. Enable the **WebAuth** option in the section for every Advanced Authentication server where this is required.
 2. Configure an external .
 3. Specify the address in **External URL** instead of specifying an address of a single Advanced Authentication server.
-

IMPORTANT: You must use the server name or IP address specified in the **Issuer** field of Salesforce.

- 3 Open the URL `https://CompanyName.my.salesforce.com/` and click **Advanced Authentication** to check the SAML 2.0 authentication.

Configuring Integration with ADFS

This section provides the configuration information on integrating Advanced Authentication with ADFS. This integration secures the ADFS connection.

The following diagram represents Advanced Authentication in ADFS.



To configure the Advanced Authentication integration with ADFS (Active Directory Federation Services) using SAML 2.0 perform the following configuration tasks:

- ◆ [“Configure the Advanced Authentication SAML 2.0 Event” on page 114](#)
- ◆ [“Make the Corresponding Changes in ADFS” on page 114](#)

NOTE: These instructions are valid only for ADFS 3 and 4.

Configure the Advanced Authentication SAML 2.0 Event

- 1 Open the Advanced Authentication Administration portal.
- 2 Click **Server Options**.
- 3 Enable **WebAuth**.
- 4 Click **Events > Add** to add a new event.
- 5 Create an event with the following parameters:
 - ◆ Name: ADFS_SAML.
 - ◆ Event Type: **SAML 2**.
 - ◆ Chains: Select the required chains.
 - ◆ Paste the content of the file `https://<adfs_hostname>/FederationMetadata/2007-06/FederationMetadata.xml` to **SP SAML 2.0 meta data**.Or
 - ◆ Click **Browse** and upload the saved XML file.
 - ◆ Click **Save**.

NOTE: Verify that you can access the file in your browser. If the file is not displayed, you have an issue on ADFS that you must resolve.

- 6 Click **Policies > SAML 2.0 Options**.
- 7 Set **External URL** to `https://AdvancedAuthenticationServerAddress/` and replace `AdvancedAuthenticationServerAddress` with the domain name or IP address of your Advanced Authentication server.

NOTE: To use multiple Advanced Authentication servers with SAML 2.0, you must do the following:

1. Enable the **WebAuth** option in the section for every Advanced Authentication server where this is required.
2. Configure an external .
3. Specify the address in **External URL** instead of specifying an address of a single Advanced Authentication server.

-
- 8 Click **Download IdP SAML 2.0 Metadata**.

You must open the file as an XML file.

NOTE: If `{"Fault":{...}}` is displayed, you must verify the configuration.

Make the Corresponding Changes in ADFS

- 1 Open the ADFS management console.
- 2 Expand **Trust Relationships**.
- 3 Click **Add Claims Provider trust**.

4 Paste OSP metadata URL `https://<AAF_server_hostname>/osp/a/TOP/auth/saml2/metadata`.

It may not work for self-signed certificate. You can copy metadata from OSP URL to an XML file and provide the file name.

5 Specify the **Display name**.

6 Select **Open the Edit Claim Rules dialog for this claims provider when the wizard closes**.

7 In **Edit Claims Rules**, click **Add Rule**.

8 Select **Send Claims Using a Custom Rule**.

9 Click **Next**.

10 Specify **Claim rule name**.

11 Paste Custom rule and click **Finish**.

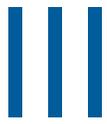
```
c:[Type == "upn"]  
  
=> issue(Type = "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn",  
Issuer = c.Issuer, OriginalIssuer = c.OriginalIssuer, Value = c.Value,  
ValueType = c.ValueType);
```

12 In **ADFS snap-in**, double click on the provider name.

13 Click **Advanced**.

14 Move the hash algorithm from SHA-256 to SHA1.

15 Click **OK**.



Maintaining Advanced Authentication

This chapter contains the following sections:

- ♦ [Chapter 9, “Reporting,” on page 119](#)
- ♦ [Chapter 10, “Logging,” on page 121](#)
- ♦ [Chapter 11, “Searching a Card Holder’s Information,” on page 135](#)

9 Reporting

The Advanced Authentication provides a reporting functionality. To log in to the Advanced Authentication Reporting Portal, open the following address: <https://<NetIQServer>/report> and sign-in using your account.

NOTE: It is required to assign chains to the **Report logon** event in the **Events** section.

The Reporting portal does not work behind a load balancer. You need to open it directly.

The following data is displayed:

Failed authentications per event

- ◆ Logon failed per event - 1
- ◆ Logon failed per event - 2
- ◆ Logon failed (total)
- ◆ Events failed
- ◆ Logon failed per user for the top 25 failed users in the **AuCore stats 2** dashboard

Successful authentications per event

- ◆ Logon succeeded per repo
- ◆ Events succeeded
- ◆ Logon succeeded per user for the top 25 successful users in the **AuCore stats 2** dashboard

List of endpoints connecting to an event

- ◆ Endpoints activity for the top 50 most active in the **AuCore stats 2** dashboard

System

- ◆ CPU load in the AuCore stats 3 dashboard
- ◆ Memory load in the AuCore stats 3 dashboard

You can select **Last N minutes** in the top-right corner to change the period of the report. To switch dashboard, click **Load saved dashboard** icon in the toolbar and select a required dashboard.

FULL ADMINS can view the reports from all tenants. To view the reports of a specific tenant, you must login to the Reporting portal as the tenant admin.

10 Logging

Advanced Authentication provides the logging functionality. All the administrative and user actions and events are logged.

Advanced Authentication supports the following types of logs:

- ♦ Syslog
- ♦ RADIUS
- ♦ Async

You can change a time zone in the upper-right section that displays your local time zone. The changes are applied for only the logs displayed and are not applied for the exported logs. Advanced Authentication resets the time zone when you switch from the **Logs** section or close the Administration portal.

1. Click **Export** in the **Logs** page.
2. Specify a **Start date** and **End date** to determine the required logging period.
3. Click **Export**. A **File Name** block appears.
4. Click on a name of the logs package (`aucore-logs_<logging_period>.tar`) to download it.

You can clear all the logs on the server that you are currently logged on. To clear the logs, perform the following steps:

1. In the **Logs** page, click **Clear**.

A message appears to confirm that you want to continue clearing the logs.

NOTE: It is a good practice to export the logs to save as backup before you delete them.

2. Click **OK** to clear the logs.

Syslog

These logs contain information about the system events and actions.

The Syslogs are classified as follows:

- ♦ 0 - 100: Maintenance
- ♦ 100 - 200: Access
- ♦ 200 - 300: App data
- ♦ 300 - 400: Endpoints
- ♦ 400 - 500: Repositories
- ♦ 500 - 600: Local users
- ♦ 600 - 700: Repository users
- ♦ 700 - 800: User templates
- ♦ 800 - 900: Policies

- ♦ 900 - 1000: Licenses
- ♦ 1000 - 1100: Settings
- ♦ 1100 - 1200: Password filter
- ♦ 1201 - 1300: Background logon

Code	Name	Class	Severity	Optional Parameters	Example
1	New Request	Operational	1	None	June 10 20:10:11 host CEF:0 AAA Core 5.0 1 New Request 1
2	Request failed	Operational	1	None	June 10 20:10:11 host CEF:0 AAA Core 5.0 1 Request failed 1
10	Server started	Operational	4	None	June 10 20:10:11 host CEF:0 AAA Core 5.0 1 Server started 4
12	Server stopped	Operational	7	None	June 10 20:10:11 host CEF:0 AAA Core 5.0 2 Server stopped 7
13	Server unexpectedly stopped	Operational	10	None	June 10 20:10:11 host CEF:0 AAA Core 5.0 3 Server unexpectedly stopped 10
50	Server Message	Operational	5	Message	June 10 20:10:11 host CEF:0 AAA Core 5.0 4 Server Message 4 This is my message
100	User logon started	Security	4	Username Ep Ep_addr Sid Unit_id Session_id Event Tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 4 User logon started 4 username=Mycompany\\demo sid=S-1-5-XXX session_id=123 event=Windows Logon ep=aaadev1.Mycompany.local ep_addr=192.168.91.1 tenant_name=Mycompany
101	User was successfully logged on	Security	7	Username Ep Ep_addr Sid Session_id method_name method_comment method_infoEvent Tenant_name Template_owner	June 10 20:10:11 host CEF:0 AAA Core 5.0 5 User was successfully logged on 7 username=Mycompany\\demo sid=S-1-5-XXX session_id=123 method_name=card method_comment=white card method_info=YYY password ep=aaadev1.Mycompany.local ep_addr=192.168.91.1 event=Windows Logon template_owner=Mycompany\\demo tenant_name=Mycompany

Code	Name	Class	Severity	Optional Parameters	Example
102	User was failed to authenticate	Security	9	Username Ep Ep_addr Sid Session_id Method_name Tenant_name Template_owner	June 10 20:10:11 host CEF:0 AAA Core 5.0 6 User was failed to authenticate 9 Username=My company\\demo sid=S-1-5-XXX session_id=123 method_name=card ep=aaadev1.Mycompany.local ep_addr=192.168.91.1 template_owner=Mycompany\ \demo tenant_name=Mycompany
103	User was switched to different method	Security	2	Username Ep Ep_addr Sid Session_id New_method_ name Tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 7 User was switched to different method 2 username=Mycomp any\\demo sid=S-1-5-XXX new_method_name=fingerprin t session_id=123 ep=aaadev1.Mycompany.local ep_addr=192.168.91.1 tenant_name=Mycompany
104	User logon session was ended	Security	2	Username Ep Ep_addr Sid Session_id Tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 8 User logon session was ended 2 username=Mycompa ny\\demo sid=S-1-5-XXX session_id=123 ep=aaadev1.Mycompany.local ep_addr=192.168.91.1 tenant_name=Mycompany
105	User logon unwanted	Security	9	Username Ep Ep_addr Method_name Tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 8 User logon session was ended 9 username=Mycompa ny\\demo sid=S-1-5-XXX session_id=123 ep=aaadev1.Mycompany.local ep_addr=192.168.91.1 method_name=voice tenant_name=Mycompany
106	User was failed to authenticate method in the middle of a chain	Security	2	Username Ep Ep_addr Method_name Tenant_name	June 10 20:10:11 (UTC+0530) host CEF:0 AAA Core 5.0 106 User was failed to authenticate method in the middle of a chain 2 ep_addr=164.99.137.1 93 method_name=PASSWORD: 1 tenant_name=TOP user_name=MFA\\topvisu p=3147

Code	Name	Class	Severity	Optional Parameters	Example
200	User read app data	Security	3	Username Ep Ep_addr Sid Session_id Data_id Record_id Tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 9 User read app data 3 username=Mycompany \\demo sid=S-1-5-XXX session_id=123 data_id=Windows Logon record_id=password ep=aaadev1.Mycompany.local ep_addr=192.168.91.1 tenant_name=Mycompany
201	User write app data	Security	4	Username Ep Ep_addr Sid Session_id Data_id Record_id Tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 10 User write app data 4 username=Mycompany \\demo sid=S-1-5-XXX session_id=123 data_id=Windows Logon record_id=password ep=aaadev1.Mycompany.local ep_addr=192.168.91.1 tenant_name=Mycompany
300	Endpoint joined	Security	4	Ep_name Ep_addr Ep_id Username Tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 11 Endp oint joined 4 ep_name=xp_client ep_id=123 username=Mycompany\Admin ep_addr=192.168.91.1 tenant_name=Mycompany
301	No rights to join endpoint	Security	7	Ep_name Ep_addr Ep_id Username Tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 12 No rights to join endpoint 7 ep_name=xp_client ep_id=123 username=Mycompany\Admin ep_addr=192.168.91.1 tenant_name=Mycompany
302	Failed to join endpoint	Operational	7	Ep_name Ep_addr Ep_id Username Reason Tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 13 Failed to join endpoint 7 ep_name=xp_client ep_id=123 username=Mycompany\Admin ep_addr=192.168.91.1 reason=Duplicated tenant_name=Mycompany

Code	Name	Class	Severity	Optional Parameters	Example
303	Endpoint remove	Security	4	Ep_name Ep_addr Ep_id Username Tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 14 Endpoint remove 4 ep_name=xp_client ep_id=123 username=Mycompany\Admin ep_addr=192.168.91.1
304	No rights to remove endpoint	Security	7	Ep_name Ep_addr Ep_id Username Tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 15 No rights to remove endpoint 7 ep_name=xp_client ep_id=123 username=Mycompany\Admin ep_addr=192.168.91.1 tenant_name=Mycompany
305	Failed to remove endpoint	Operational	7	Ep_name Ep_addr Ep_id Username Reason Tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 16 Failed to remove endpoint 7 ep_name=xp_client ep_id=123 username=Mycompany\Admin ep_addr=192.168.91.1 reason=Duplicated tenant_name=Mycompany
306	Endpoint session started	Operational	2	Ep_name Ep_addr Ep_id Tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 17 Endpoint session started 2 ep_name=xp_client ep_id=123 ep_addr=192.168.91.1 tenant_name=Mycompany
307	Endpoint session ended	Operational	2	Ep_name Ep_addr Ep_id Tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 18 Endpoint session ended 2 ep_name=xp_client ep_id=123 ep_addr=192.168.91.1 tenant_name=Mycompany
308	Invalid endpoint secret	Security	7	Ep_name Ep_addr Ep_id Tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 17 Invalid endpoint secret 2 ep_name=xp_client ep_id=123 ep_addr=192.168.91.1 tenant_name=Mycompany

Code	Name	Class	Severity	Optional Parameters	Example
309	Failed to create endpoint session	Operational	7	Ep_name Ep_addr Ep_id Reason Tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 18 Failed to create endpoint session 7 ep_name=xp_client ep_id=123 ep_addr=192.168.91.1 reason=No memory tenant_name=Mycompany
310	Failed to end endpoint session	Operational	7	Ep_name Ep_addr Ep_id Reason Tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 18 Failed to create endpoint session 7 ep_name=xp_client ep_id=123 ep_addr=192.168.91.1 reason=No memory tenant_name=Mycompany
401	New repository was added	Operational	4	repo_name repo_type session_id tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 19 New repository was added 4 repo_name=Mycompany repo_type=LDAP session_id=123 tenant_name=Mycompany
402	Failed to add repository	Operational	7	repo_name repo_type session_id reason tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 20 Failed to add repository 7 repo_name=Myco mpany repo_type=LDAP session_id=123 reason=repo already exists tenant_name=Mycompany
403	Repository was removed	Operational	4	repo_name repo_type session_id tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 21 Repo sitory was removed 4 repo_name=Myco mpany repo_type=LDAP session_id=123 tenant_name=Mycompany
404	Failed to remove repository	Operational	7	repo_name repo_type session_id reason tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 22 Failed to remove repository 7 repo_name=Mycompany repo_type=LDAP session_id=123 reason=not empty tenant_name=Mycompany

Code	Name	Class	Severity	Optional Parameters	Example
405	Repository configuration was changed	Operational	4	repo_name repo_type session_id reason tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 23 Repository configuration was changed 4 repo_name=Mycompany repo_type=LDAP session_id=123 tenant_name=Mycompany
501	Local user was created	Operational	4	user_name session_id tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 24 Local user was created 4 user_name=admin session_id=123 tenant_name=Mycompany
502	Local user was removed	Operational	5	user_name session_id tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 25 Local user was removed 5 user_name=admin session_id=123 tenant_name=Mycompany
503	Failed to create local user	Operational	4	user_name session_id reason tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 26 ailed to create local user 4 user_name=admin session_id=123 reason=already exists tenant_name=Mycompany
504	No rights to remove local user	Security	7	user_name session_id tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 26 ailed to create local user 4 user_name=admin session_id=123 reason=already exists tenant_name=Mycompany
505	Failed to remove local user	Operational	5	user_name session_id reason tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 28 Failed to remove local user 5 user_name=admin session_id=123 reason=can't remove currently logged on user tenant_name=Mycompany
506	No rights to create local user	Security	7	user_name session_id tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 29 Failed to create local user 7 user_name=admin session_id=123 tenant_name=Mycompany

Code	Name	Class	Severity	Optional Parameters	Example
601	User was created	Operational	4	user_name session_id repo_name tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 30 User was created 4 username=Someon e session_id=123 repo_name=Mycompany tenant_name=Mycompany
602	No rights to create user	Security	7	user_name session_id repo_name tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 31 No rights to create user 7 username=Someone session_id=123 repo_name=Mycompany tenant_name=Mycompany
603	Failed to create user	Operational	4	user_name session_id repo_name reason tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 32 Failed to create user 4 user_name=someone session_id=123 repo_name=123 reason=already exists tenant_name=Mycompany
604	User was removed	Operational	5	user_name session_id repo_name tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 33 User was removed 5 username=Someo ne session_id=123 repo_name=Mycompany tenant_name=Mycompany
605	No rights to remove user	Security	7	user_name session_id repo_name tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 34 No rights to remove user 7 username=Someone session_id=123 repo_name=Mycompany tenant_name=Mycompany
606	Failed to remove user	Operational	5	user_name session_id repo_name reason tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 35 Failed to remove user 5 user_name=someone session_id=123 repo_name=123 reason=not found tenant_name=Mycompany
701	Template was assigned to the user	Security	7	user_name session_id ap_name comment tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 36 Templ ate was assigned to the user 7 user_name=Mycompan y\some session_id=123 ap_name=Card comment=white card tenant_name=Mycompany

Code	Name	Class	Severity	Optional Parameters	Example
702	Template was enrolled for the user	Security	7	user_name session_id ap_name comment tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 37 Template was enrolled for the user 7 user_name=Mycompany\some session_id=123 ap_name=hand 3D comment=left hand tenant_name=Mycompany
703	User enroll the assigned template	Security	7	user_name session_id ap_name comment tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 38 User enroll the assigned template 7 user_name=Mycompany\some session_id=123 ap_name=hand 3D comment=left hand tenant_name=Mycompany
704	Template is linked	Security	8	user_name target_user_name session_id ap_name comment tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 39 Template is linked 8 user_name=Mycompany\some target_user_name=Mycompany\boss session_id=123 ap_name=hand 3D comment=left hand tenant_name=Mycompany
705	Failed to assign template to the user	Security	7	user_name session_id ap_name comment reason tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 40 Failed to assign template to the user 7 user_name=Mycompany\some session_id=123 ap_name=Card comment=white card reason=no license tenant_name=Mycompany
706	Failed to enroll template for the user	Security	7	user_name session_id ap_name comment reason tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 41 Failed to enroll template for the user 7 user_name=Mycompany\some session_id=123 ap_name=hand 3D comment=left hand reason=ap error tenant_name=Mycompany

Code	Name	Class	Severity	Optional Parameters	Example
707	User can't enroll the assigned template	Security	7	user_name session_id ap_name comment reason tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 41 User can't enroll the assigned template 7 user_name=Myco mpany\some session_id=123 ap_name=hand 3D comment=left hand reason=AP not installed on client side tenant_name=Mycompany
709	Failed to link template	Security	8	user_name target_user_name session_id ap_name comment reason tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 42 Failed to link template 8 user_name=Myco mpany\some target_user_name=Mycompa ny\boss session_id=123 ap_name=hand 3D comment=left hand reason=target user can't be found tenant_name=Mycompany
709	Template link was removed	Security	6	user_name target_user_name session_id ap_name comment tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 43 Templ ate link was removed 6 user_name=Myco mpany\some target_user_name=Mycompa ny\boss session_id=123 ap_name=hand 3D comment=left hand tenant_name=Mycompany
710	Failed to remove template link	Security	6	user_name target_user_name session_id ap_name comment reason tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 44 Failed to remove template link 6 user_name=Mycompany \some target_user_name=Mycompa ny\boss session_id=123 ap_name=hand 3D comment=left hand reason=too small carma tenant_name=Mycompany
711	Template was removed	Security	6	user_name ap_name comment session_id tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 45 Templ ate was removed 6 user_name=Myco mpany\some session_id=123 ap_name=hand 3D comment=left hand tenant_name=Mycompany

Code	Name	Class	Severity	Optional Parameters	Example
712	Failed to remove template	Security	6	user_name ap_name comment session_id reason tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 46 Failed to remove template 6 user_name=Mycompany\ some session_id=123 ap_name=hand 3D comment=left hand reason=only owner can remove template tenant_name=Mycompany
713	Template was changed	Security	7	user_name ap_name comment session_id tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 47 Template was changed 7 user_name=Mycompany\ some session_id=123 ap_name=hand 3D comment=left hand tenant_name=Mycompany
714	Failed to change template	Security	6	user_name ap_name comment session_id reason tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 48 Failed to change template 6 user_name=Mycompany\ some session_id=123 ap_name=hand 3D comment=left hand reason=only owner can change template tenant_name=Mycompany
715	Template was changed during logon	Security	5	user_name ap_name comment session_id tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 49 Template was changed during logon 7 user_name=Mycompany\ some session_id=123 ap_name=TOTP comment=ASA (iPhone) tenant_name=Mycompany
801	Policy was changed	Security	7	session_id scope comp_name policy_name old_value new_value	June 10 20:10:11 host CEF:0 AAA Core 5.0 50 Policy was changed 7 session_id=123 scope=global comp_name=password poliices policy_name=minimal password length old_value=4 new_value=8
802	No rights to change policy	Security	8	session_id scope comp_name policy_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 51 No rights to change policy 8 session_id=123 scope=global comp_name=password poliices policy_name=minimal password

Code	Name	Class	Severity	Optional Parameters	Example
803	Failed to change policy	Operational	7	session_id scope comp_name policy_name reason	June 10 20:10:11 host CEF:0 AAA Core 5.0 52 Failed to change policy 7 session_id=123 scope=global comp_name=password poliices policy_name=minimal password reason=policy not found
901	New license was added	Operational	3	session_id license_id users_count enabled_features expire_date	June 10 20:10:11 host CEF:0 AAA Core 5.0 53 New license was added 3 session_id=123 license_id=111 users_count=101 enabled_features=client,rte,np s expire_date=31/12/2014
902	Failed to add license	Operational	8	session_id license_id users_count enabled_features expire_date reason	June 10 20:10:11 host CEF:0 AAA Core 5.0 54 Failed to add license 8 session_id=123 license_id=111 users_count=101 enabled_features=client,rte,np s expire_date=31/12/2013 reason=already expired
1001	Global setting was changed	Security	9	session_id setting_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 55 Globa l setting was changed 9 session_id=123 setting_name=syslog_server
1002	No rights to change global setting	Security	9	session_id setting_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 56 No rights to change global setting 9 session_id=123 setting_name=syslog_server
1003	Failed to change global setting	Operational	9	session_id setting_name reason	June 10 20:10:11 host CEF:0 AAA Core 5.0 57 Failed to change global setting 9 session_id=123 setting_name=syslog_server reason=server is unavailable
1101	Password was changed	Security	5	user_name ep_addr tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 15 Pass word was changed 5 ep=xp_client user_name=Mycompany\Admin ep_addr=192.168.91.1 tenant_name=Mycompany

Code	Name	Class	Severity	Optional Parameters	Example
1102	Password was reset	Security	8	user_name ep ep_addr tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 15 Password was reset 8 ep=xp_client user_name=Mycompany\Admin ep_addr=192.168.91.1 tenant_name=Mycompany
1201	User successfully logged on using local cache	Security	8	user_name ep_addr event chain_name logon_time tenant_name	June 10 20:10:11 host CEF:0 AAA Core 5.0 15 User successfully logged on using local cache 8 ep=xp_client user_name=Mycompany\Admin ep_addr=192.168.91.1 event=windows logon chain_name=LDAP+SMS logon_time=2017-11-05 08:10:03 tenant_name=Mycompany

RADIUS Logs

These logs contain information about the logs that are recorded for the RADIUS client or server.

Async Logs

These logs contain information about the asynchronous delivery of OTP messages for the SMS, Email, and Voice methods.

11

Searching a Card Holder's Information

With the Search Card portal, you can get a card holder's contact information by tapping the card on the card reader. Information such as name of the card holder, repository information, email address, and mobile number of the user can be obtained.

You must assign chains to the **Search card** event in the **Events** section.

IMPORTANT: To use this feature, you must have the Device Service installed on the computer.

To get the user information from the card, perform the following steps:

1. Log in to the Advanced Authentication Search Card portal (<https://<AdvancedAuthenticationServer>/search-card>).
2. Tap a card on the card reader. The card holder's user name, repository information, email address, and mobile number are displayed.

NOTE: If the card was not enrolled before, a message `No user was found for this card` is displayed.
