opentext™

Advanced Authentication 25.1 (v6.5)

Administration Guide

March 2025

Legal notice

Copyright 2014 - 2025 Open Text

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

Contents

About this book 13

1 Introduction to Advanced Authentication 15

1.1 How Advanced Authentication is better than other solutions 15
1.2 Key features 16
1.3 Advanced Authentication server components 16
1.3.1 Administration Portal 17
1.3.2 Self-Service Portal 18
1.3.3 Helpdesk Portal 18
1.3.4 Reporting Portal 18
1.4 Architecture 18
1.4.1 Basic architecture 18
1.4.2 Enterprise level architecture 19
1.4.3 Enterprise architecture with A Load Balancer 21
1.5 Terminology 22
1.5.1 Authentication method 22
1.5.2 Authentication chain 22
1.5.3 Authentication event 22
1.5.4 Endpoint 22

Part I Configuring Advanced Authentication 23

2 Logging in to the Advanced Authentication administration Portal 25

3 End to end configuration with examples 27

3.1 Implementing multi-factor authentication to VPN 27
3.1.1 Prerequisites 28
3.1.2 Considerations before configuration 28
3.1.3 Add a Repository 29
3.1.4 Configure methods 30
3.1.5 Create a chain 30
3.1.6 Configure public external URLs policy 30
3.1.7 Assign chain to RADIUS server event 31
3.1.8 Configure the OpenVPN server 31
3.1.9 End user tasks 31
3.2 Securing Windows workstation with multi-factor authentication 33
3.2.1 Prerequisites 34
3.2.2 Points to consider before configuration 34
3.2.3 Configure methods 35
3.2.4 Create a chain 36
3.2.5 Configure SMS sender policy 37
3.2.6 Assign chain to Windows Logon event 37
3.2.7 End user tasks 37
3.3 Configuring TOTP from OTP Desktop tool as one of the factors to access a corporate Portal 38
3.3.1 Prerequisites 39

3.3.2 Configure methods . . . 40
3.3.3 Create a chain . . . 40
3.3.4 Create a SAML2 event . . . 41
3.3.5 Configure web authentication policy . . . 41
3.3.6 Obtaining the signing certificate of Advanced Authentication . . . 41
3.3.7 Configure Google workspace . . . 41
3.3.8 Generate and send an enrollment link to users . . . 43
3.3.9 End user tasks . . . 43
3.4 Integrate Advanced Authentication and Office 365 without using AD FS . . . 45
3.4.1 Prerequisites . . . 45
3.4.2 Administrator tasks . . . 45
3.4.3 End user tasks . . . 49
3.5 Integrate Advanced Authentication and Office 365 using AD FS . . . 50
3.5.1 Prerequisites . . . 50
3.5.2 Administrator tasks . . . 51
3.5.3 End user tasks . . . 54

Part II Configuring the Advanced Authentication settings 55

4 Managing dashboard 57

4.1 Adding widgets . . . 57
4.1.1 Pie chart . . . 58
4.1.2 Stacked chart . . . 58
4.1.3 Activity stream . . . 58
4.1.4 Enroll activity stream . . . 58
4.1.5 Users . . . 58
4.1.6 Authenticators . . . 59
4.1.7 Event count line chart . . . 59
4.1.8 Events count line chart grouped by field . . . 59
4.1.9 Distinct events count line chart . . . 59
4.1.10 Distinct events count line chart grouped by field . . . 59
4.1.11 Server messages . . . 60
4.2 Customizing dashboard . . . 60
4.3 Updating dashboard to view real time or historical data . . . 60
4.4 Customizing the default widgets . . . 61
4.4.1 Server metrics . . . 62
4.4.2 Tenants . . . 62
4.4.3 Billing . . . 62
4.4.4 Logons per result . . . 62
4.4.5 Total users . . . 62
4.4.6 Total users per event . . . 62
4.4.7 Activity stream . . . 62
4.4.8 Successful/Failed Logons . . . 62
4.4.9 Top events with successful Logon per chain . . . 62
4.4.10 Top events with failed Logon per method . . . 63
4.4.11 Top 10 events . . . 63
4.4.12 Top 10 chains with successful result . . . 63
4.4.13 Top 10 servers . . . 63
4.4.14 Top 10 tenants . . . 63
4.4.15 Top 10 Repositories . . . 63
4.4.16 Top 5 events for Logons . . . 63
4.4.17 Top 5 users for Logons . . . 63
4.4.18 Top 10 users with failed Logon . . . 63

4.4.19 Top 10 users . . . 63
4.4.20 Top 10 methods with failed result . . . 63
4.5 Exporting widgets . . . 64

5 Managing Applications 65

6 Adding a Repository 67

6.1 Adding an LDAP Repository . . . 67
6.1.1 Advanced settings . . . 70
6.1.2 Adding an AD LDS Repository with the configured AD LDS proxy . . . 79
6.2 Adding an SQL database . . . 79
6.3 Adding a Cloud Bridge external Repository . . . 80
6.3.1 Advanced settings . . . 83
6.3.2 Health check settings . . . 90
6.3.3 Synchronizing Cloud Bridge Repository . . . 90
6.3.4 Testing Cloud Bridge . . . 90
6.3.5 Force configuring Cloud Bridge . . . 91
6.3.6 Enabling fast synchronization for eDirectory Repository . . . 91
6.4 Local Repository . . . 92
6.5 Adding a SCIM managed Repository . . . 93

7 Configuring methods 95

7.1 Customizing methods name . . . 96
7.2 Configuring tenancy settings . . . 96
7.3 Capabilities of authentication methods . . . 96
7.4 Apple Touch ID . . . 98
7.5 BankID . . . 98
7.6 Bluetooth . . . 99
7.7 Bluetooth eSec . . . 100
7.8 Card . . . 100
7.9 Denmark National ID . . . 102
7.10 Device authentication . . . 102
7.10.1 Windows Trusted Platform Module (TPM) . . . 103
7.10.2 Without using the Trusted Platform Module (Non-TPM) . . . 104
7.11 Email OTP . . . 104
7.11.1 Customizing Email settings for an event . . . 106
7.12 Emergency Password . . . 107
7.13 Facial Recognition . . . 107
7.13.1 Azure cognitive service . . . 108
7.13.2 Contactable KYC service . . . 109
7.14 FIDO2 . . . 110
7.15 Fingerprint . . . 113
7.16 Flex OTP . . . 116
7.17 HANIS Face . . . 117
7.18 HANIS Fingerprint . . . 118
7.19 LDAP Password . . . 121
7.20 OATH OTP . . . 122
7.20.1 HOTP . . . 123
7.20.2 TOTP . . . 124

7.20.3 Importing PSKC or CSV files . . . 126
7.20.4 CSV file format to import OATH compliant tokens . . . 127
7.21 Out-of-band . . . 127
7.21.1 Authentication agent for Windows . . . 128
7.21.2 Authentication agent for web . . . 129
7.22 Password . . . 130
7.23 PKI . . . 131
7.23.1 PKI device . . . 132
7.23.2 Virtual smartcard . . . 134
7.24 RADIUS client . . . 136
7.25 SAML service provider . . . 137
7.26 Security questions . . . 140
7.26.1 Adding questions . . . 141
7.27 Smartphone . . . 141
7.27.1 Configuring Smartphone method . . . 143
7.27.2 Configuring enrollment link . . . 148
7.27.3 Setting up geo-fence for Smartphone . . . 148
7.27.4 Priority vendor requirements . . . 149
7.28 SMS OTP . . . 149
7.29 Swisscom Mobile ID . . . 151
7.30 FIDO U2F . . . 152
7.30.1 Configuring the certificate settings . . . 153
7.30.2 Configuring Facets . . . 153
7.30.3 Configuring yubikey for Advanced Authentication server . . . 154
7.30.4 Configuring a web server to use the FIDO U2F authentication . . . 154
7.31 Voice . . . 156
7.32 Voice OTP . . . 158
7.33 Web authentication method . . . 159
7.33.1 SAML for Advanced Authentication . . . 160
7.33.2 OpenID connect for Advanced Authentication . . . 163
7.33.3 OAuth 2.0 for Advanced Authentication . . . 166
7.34 Windows Hello . . . 167

8 Creating a chain 169

9 Configuring events 173

9.1 Configuring an existing event . . . 173
9.1.1 ADFS event . . . 177
9.1.2 AdminUI event . . . 177
9.1.3 Authentication agent event . . . 178
9.1.4 Authenticators management event . . . 178
9.1.5 Desktop OTP tool event . . . 179
9.1.6 Helpdesk event . . . 179
9.1.7 Helpdesk user event . . . 179
9.1.8 Linux Logon event . . . 180
9.1.9 Mac OS Logon event . . . 180
9.1.10 Mainframe Logon event . . . 180
9.1.11 NAM event . . . 180
9.1.12 NCA event . . . 181
9.1.13 OOB UI Logon event . . . 181
9.1.14 RADIUS server event . . . 181
9.1.15 Report Logon event . . . 181

9.1.16 SCIM API event . . . 181
9.1.17 Search card event . . . 182
9.1.18 Smartphone enrollment event . . . 182
9.1.19 Tokens management event . . . 182
9.1.20 Windows Logon event . . . 182
9.2 Creating a customized event . . . 183
9.2.1 Creating a generic event . . . 183
9.2.2 Creating an OS Logon (Domain) event . . . 185
9.2.3 Creating an OAuth 2.0 / openid connect event . . . 185
9.2.4 Creating a SAML 2.0 event . . . 189
9.2.5 Creating a RADIUS event . . . 193

10 Managing endpoints 195

11 Configuring policies 197

11.1 Authenticator management options . . . 198
11.1.1 Enabling sharing of authenticators for the helpdesk administrators . . . 198
11.1.2 Disabling re-enrollment of the authenticators . . . 198
11.2 Cache options . . . 199
11.3 Custom branding . . . 200
11.3.1 Customizing the login page of web authentication events . . . 201
11.4 Custom messages . . . 208
11.4.1 Customizing messages in the custom localization file . . . 209
11.4.2 Customizing a specific message on the Portal . . . 210
11.4.3 Customizing authentication request message for Smartphone method . . . 210
11.4.4 Customizing prompt messages of the authentication methods for RADIUS event . . . 211
11.4.5 Customizing the messages for clients . . . 212
11.4.6 Localizing the web UI and messages . . . 213
11.5 Database options . . . 214
11.6 Delete me options . . . 214
11.7 Endpoint management options . . . 215
11.8 Event categories . . . 216
11.9 Geo fencing options . . . 216
11.10 Google reCAPTCHA options . . . 217
11.10.1 Registering the Google reCAPTCHA account . . . 217
11.10.2 Configuring Google reCAPTCHA for Advanced Authentication . . . 218
11.10.3 Enabling the Google reCAPTCHA options policy for events . . . 218
11.11 Help options . . . 218
11.12 Helpdesk options . . . 219
11.13 Linked chains . . . 219
11.14 Lockout options . . . 220
11.15 Login options . . . 221
11.16 Logon filter for Active Directory . . . 223
11.17 Mail sender . . . 223
11.18 OAuth2 Application . . . 224
11.19 Password filter for Active Directory . . . 225
11.20 RADIUS Options . . . 226
11.20.1 Input rule . . . 227
11.20.2 Event selection rule . . . 228
11.20.3 Chain selection rule . . . 229
11.20.4 Result specification rule . . . 230

Scenario 1: selecting an authentication chain based on NAS ID and display groups of the authenticated user 231
Scenario 2: mapping RADIUS requests to a specific RADIUS server event based on NAS ID and
display Email address of the authenticated user 234
11.20.5 Adding clients 236
11.21 Reporting options 237
11.22 SMS sender. 237
11.22.1 Generic 238
11.22.2 Twilio 242
11.22.3 MessageBird 244
11.23 Voice sender. 244
11.24 Web Authentication 246
11.24.1 Configuring the identity provider 246
11.24.2 Downloading the identity provider SAML metadata 247
11.24.3 Configuring timeout 247
11.24.4 Enabling the client event selection 248
11.24.5 Enabling the client chain selection 248
11.24.6 Customizing messages and authentication method names for the web authentication events 248
11.25 Windows Credential Providers 249

12 Configuring the server options 251

12.1 Generating OSP keystores 251

13 Adding a Report 253

14 Enrolling the authentication methods 261

15 Scripts option 263

15.1 Generating RADIUS script 263

Part III Configuring risk settings 265

16 Configuring Risk Service 267

16.1 Monitoring risk audit logs 267

17 Understanding how Risk Service works through scenarios 269

17.1 Assessing risks based on the IP Address 269
17.2 Allowing employees to access the human resources Portal outside the corporate network 271

18 Troubleshooting Risk Service configuration 275

18.1 An error in Syslog when the Risk Service license is not Applied 275
18.2 Cannot read the log file error in risk logs 275

Part IV Configuring integrations 277

19 OAuth 2.0 279

19.1 Building blocks of OAuth 2.0 . . . 279
19.1.1 OAuth 2.0 roles . . . 279
19.1.2 OAuth 2.0 grants . . . 280
19.2 Sample OAuth 2.0 Application integrated with Advanced Authentication . . . 282
19.2.1 Running the sample web Application . . . 288
19.3 OAuth 2.0 attributes . . . 289
19.4 Non standard endpoints . . . 290

20 RADIUS server 293

Customizing prompt messages for RADIUS event . . . 295
Challenge-Response authentication . . . 295

21 SAML 2.0 297

21.1 Integrating Advanced Authentication with SAML 2.0 . . . 297
21.1.1 Requesting Advanced Authentication methods and chains through a SAML authnrequest . . . 298

22 Examples of integrations 301

22.1 Configuring integration with Barracuda . . . 301
22.1.1 Configuring the Advanced Authentication RADIUS server . . . 302
22.1.2 Configuring the Barracuda SSL VPN Appliance . . . 303
22.1.3 Authenticating on Barracuda SSL VPN using Advanced Authentication . . . 303
22.2 Configuring integration with Citrix NetScaler . . . 303
22.2.1 Configuring the Advanced Authentication RADIUS server . . . 304
22.2.2 Configuring the Citrix NetScaler Appliance . . . 305
22.2.3 Authenticating on the Citrix NetScaler using Advanced Authentication . . . 305
22.3 Configuring integration with dell sonicwall SRA EX-Virtual Appliance . . . 305
22.3.1 Configuring the Advanced Authentication RADIUS server . . . 306
22.3.2 Configuring the dell sonicwall SRA Appliance . . . 307
22.3.3 Authenticating on Dell SonicWall Workspace using Advanced Authentication . . . 307
22.4 Configuring integration with FortiGate . . . 307
22.4.1 Configuring the Advanced Authentication RADIUS server . . . 308
22.4.2 Configuring the FortiGate Appliance . . . 308
22.4.3 Authenticating on FortiGate using Advanced Authentication . . . 309
22.5 Configuring integration with OpenVPN . . . 309
22.5.1 Configuring the Advanced Authentication RADIUS server . . . 310
22.5.2 Configuring the OpenVPN Appliance . . . 310
22.6 Configuring integration with Palo Alto GlobalProtect Gateway . . . 311
22.6.1 Adding the RADIUS server . . . 311
22.6.2 Adding an authentication profile . . . 312
22.6.3 Configuring globalprotect gateway . . . 312
22.7 Configuring integration with Salesforce . . . 312
22.7.1 Configuring the Advanced Authentication SAML 2.0 event . . . 312
22.7.2 Configuring to authenticate on Salesforce with SAML 2.0 . . . 313
22.7.3 Obtaining the signing certificate of Advanced Authentication . . . 313
22.7.4 Configuring the Salesforce domain name . . . 313

22.7.5 Configuring the SAML provider 314
22.7.6 Verifying Single Sign-On to Salesforce 315
22.8 Configuring integration with ADFS 315
22.8.1 Configuring the Advanced Authentication SAML 2.0 event 316
22.8.2 Making the corresponding changes in ADFS 317
22.9 Configuring integration with Google G suite 317
22.9.1 Obtaining the signing certificate of Advanced Authentication 318
22.9.2 Configuring Google G suite 318
22.9.3 Configuring the Advanced Authentication event 319
22.9.4 Configuring to authenticate on Google G-Suite with SAML 2.0 319
22.9.5 Verifying single sign-on to Google suite 319
22.10 Configuring integration with Office 365 320
22.10.1 Configuring Advanced Authentication SAML 2.0 event 320
22.10.2 Making the corresponding changes in ADFS 321
22.10.3 Authenticating on Office 365 322
22.11 Configuring integration with Sentinel 322
22.11.1 Configuring the CEF log forward policy on Advanced Authentication 322
22.11.2 Searching the events on Sentinel 323
22.12 Configuring integration with Office 365 without using ADFS 323
22.12.1 Configuring the Advanced Authentication SAML 2.0 event 324
22.12.2 Configuring the identity provider URL 324
22.12.3 Obtaining the signing certificate of Advanced Authentication 324
22.12.4 Enabling Single Sign-On to Office 365 324
22.12.5 Verifying Single Sign-On to Office 365 326
22.13 Configuring integration with Cisco AnyConnect 326
22.13.1 Configuring the Advanced Authentication RADIUS server 328
22.13.2 Enabling the Connection profile in cisco ASA 328
22.13.3 Creating a group policy in cisco ASA 328
22.13.4 Adding a RADIUS token server in cisco ISE 328
22.13.5 Configuring policy sets in cisco ISE 329
22.13.6 Authenticating to Cisco AnyConnect using Advanced Authentication 329
22.14 Configuring integration with GitLab 329
22.14.1 Configuring GitLab for Advanced Authentication 330
22.14.2 Creating the relying party trust on ADFS 331
22.14.3 Creating the claims party trust on ADFS 332
22.14.4 Configuring the SAML 2.0 event on Advanced Authentication 333
22.15 Configuring integration with Filr 334
22.16 Configuring integration with DUO authentication proxy 334
22.16.1 Configuring the Advanced Authentication RADIUS client 334
22.16.2 Configuring the DUO authentication proxy 335
22.17 Configuring integration with ArcSight 335
22.17.1 Configuring ArcSight 335
22.17.2 Configuring the SAML 2.0 event on Advanced Authentication 336
22.17.3 Authenticating on ArcSight with SAML 2.0 337
22.18 Configuring integration with Azure 337
22.18.1 Configuring Advanced Authentication SAML 2.0 event 337
22.18.2 Configuring ADFS 338
22.18.3 Authenticating on Azure 339
22.19 Configuring integration with Amazon Web Services Single Sign-On 339
22.19.1 Downloading the SAML metadata of Advanced Authentication 340
22.19.2 Setting-up AWS Single Sign-On 340
22.19.3 Configuring a SAML 2.0 event on Advanced Authentication 340
22.19.4 Verifying the integration 340

Part V Maintaining Advanced Authentication **343**

23 Logging **345**

23.1 Syslog 346
23.2 RADIUS logs 378
23.3 Async logs 378
23.4 Long tasks logs 379
23.5 Long scheduler logs 379
23.6 Fingerprint logs 379
23.7 Risk Service logs 379

24 Reporting **381**

25 Searching a card holder's information **383**

26 Troubleshooting **385**

26.1 Administration Portal is accessible without any authentication 385
26.2 The ON/OFF switch is broken if the screen resolution is 110% 385
26.3 Users can login using the old Password 386
26.4 Issue with authenticating on Office 365 386
26.5 Error while downloading logs package 386
26.6 Error while configuring SMS OTP method 386
26.7 Configuring the log rotation in Docker before deploying the Advanced Authentication server . . . 387
26.8 Error while logging in to Salesforce 387
26.9 Analyzing performance issue using the profiling tool 387
26.10 Validating JSON syntax in SLAnalyzer 388
26.11 Push messages does not Appear in Smartphone 388
26.12 Insufficient allocated disk space 389
26.12.1 Clearing the log files 389
26.12.2 Expanding the root partition 389
26.13 Issue with cluster synchronization 390
26.14 Users with very large usergroups attributes are being rejected by the NGINX reverse proxy 390
26.15 Error while loading the dashboard data 391

27 General best practices **393**

27.1 Recommendations to prevent phishing attacks 394

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 Introduction to Advanced Authentication

Advanced Authentication™ is a multi-factor authentication solution that goes beyond the typical username-and-password-based authentication, enabling you to protect your IT infrastructure and sensitive data more effectively and securely. Advanced Authentication allows you to authenticate on diverse platforms using various devices and methods such as Fingerprint, Card readers, Facial Recognition, and One-Time Passwords (OTP). In addition, Advanced Authentication's authentication framework is both comprehensive and intuitive, ensuring secure access to all your devices while minimizing administrative overhead.

Generalized Authentication comprises 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 (as determined by biometrics, such as via fingerprint readers or iris scanners).

Achieving Multi-factor (aka strong) authentication requires utilizing any two (or more) factors from this list in combination. For example, multi-factor authentication could be realized by including a password with a physical token, or by combining both a smartcard and a fingerprint.

This section contains the following topics:

- Section 1.1, "How Advanced Authentication is better than other solutions," on page 15
- Section 1.2, "Key features," on page 16
- Section 1.3, "Advanced Authentication server components," on page 16
- Section 1.4, "Architecture," on page 18
- Section 1.5, "Terminology," on page 22

1.1 How Advanced Authentication is 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, helping organizations to distribute their authentication services globally.

1.2 Key features

- **Multi-factor Authentication**: The solution provides the flexibility to choose from more than twenty authentication methods to create authentication chains. You can assign these chains to various events to configure authentication for different types of endpoints.
- **Supports Multiple Repositories:** Advanced Authentication supports Active Directory, Active Directory Lightweight Domain Services, eDirectory, and other RFC 2307 compliant LDAP repositories.
- **Supports Distributed Environments**: Advanced Authentication works on geographically distributed environments and under high utilization.
- **Multitenancy**: A single Advanced Authentication solution can support multiple tenants (multiple customers in differing 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 the Helpdesk or Security Officer role can manage authentication for end users through the Helpdesk portal.
- **Supports RADIUS Server**: Advanced Authentication contains a built-in RADIUS server and provides strong authentication for third-party RADIUS clients. It also act as a RADIUS client for use with third-party RADIUS servers.
- **Supports ADFS 3 and 4, 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 users who need to access third-party consumer applications.
- **Reporting**: Advance Authentication provides a Reporting portal that allows access to various security reports. You can also create customized reports based on your requirements.
- **Syslog support**: Advanced Authentication provides a central logging server that can be configured to forward logs to an external Syslog server.
- **FIPS 140-2 Compliant Encryption**: Advanced Authentication adheres to Federal Information Processing Standard (FIPS) 140-2.
- **Supports Localization**: Advanced Authentication supports several languages, such as Arabic, Canadian French, Chinese Simplified, Chinese Traditional, Danish, Dutch, English, French, German, Italian, Japanese, Polish, Portuguese (Brazilian), Russian, Spanish, Hebrew, and Swedish.

1.3 Advanced Authentication server components

The Advanced Authentication server comprises the following components:

- **Administration Portal**

  For more information, see Section 1.3.1, “Administration Portal,” on page 17
- **Self-Service Portal**

  For more information, see Section 1.3.2, “Self-Service Portal,” on page 18
- **Helpdesk Portal**

  For more information, see Section 1.3.3, “Helpdesk Portal,” on page 18

- **Reporting Portal**

  For more information, see Section 1.3.4, “Reporting Portal,” on page 18

1.3.1 Administration Portal

The Administration Portal is a centralized portal that helps you to configure and manage various authentication settings such as methods, events, and so on. You also can use it to configure various policies that have an effect on how authentication is performed. Use the Administration Portal to perform any of the following tasks:

- **Add repositories**: A repository is an internal representation of a database that contains user information. For example: An organization, Digital Airlines might store its user information in Active Directory to manage the information for each user, such as username, telephone, address, and so on. Advanced Authentication administrators can add this Active Directory instance to Advanced Authentication as a repository. This allows various departments in the organization (such as IT, finance, HR, and Engineering) to authenticate users described by the information from the database and to customize the authentication experience based on their department and/or organizational requirements. For more information about how to add repositories, see “Adding a Repository”.
- **Configure methods**: A method (also called an authenticator) helps to confirm the identity of a user (or in some cases, a machine) that is trying to log on or access resources. For example, you might want users to verify their identity by using a smart card or by providing a password. As an administrator, you can configure the settings for any of the supported methods. For more information about how to configure methods, see “Configuring methods”.
- **Create chains:** A chain is a specific combination of methods. To successfully authenticate, users must verify themselves with every methods in a chain. For example, a chain can be created with Fingerprint and Card methods for the IT department and a chain with the Smartphone, LDAP Password, and HOTP methods can be assigned to the Engineering department. In this example, the IT user must provide both a known card and a recognized fingerprint for authentication, while the engineering user must first authenticate using a smartphone application and then provide LDAP along with an additional one-time password. For more information about how to create chains, see “Creating a chain”.
- **Configure events**: Events enable a specific application or device (such as Windows machine, RADIUS client, third-party client, and so on.) to use Advanced Authentication functionality. Events provide the necessary protocol and policy for access and are triggered whenever a specific device or application needs to perform an authentication. After creating a chain, the 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**: Policies are rules and settings that are specific to users, devices, or locations. They are managed by the Administrator to customize 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”.

1.3.2 Self-Service Portal

The Self-Service Portal allows users to manage the available authentication methods. This portal consists of **Enrolled authenticators** and **Add authenticator**. T**he 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.

1.3.3 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.

1.3.4 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".

1.4 Architecture

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

- Basic Architecture

  For more information, see Section 1.4.1, "Basic architecture," on page 18
- Enterprise Level Architecture

  For more information, see Section 1.4.2, "Enterprise level architecture," on page 19
- Enterprise Architecture With A Load Balancer

  For more information, see Section 1.4.3, "Enterprise architecture with A Load Balancer," on page 21

1.4.1 Basic architecture

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



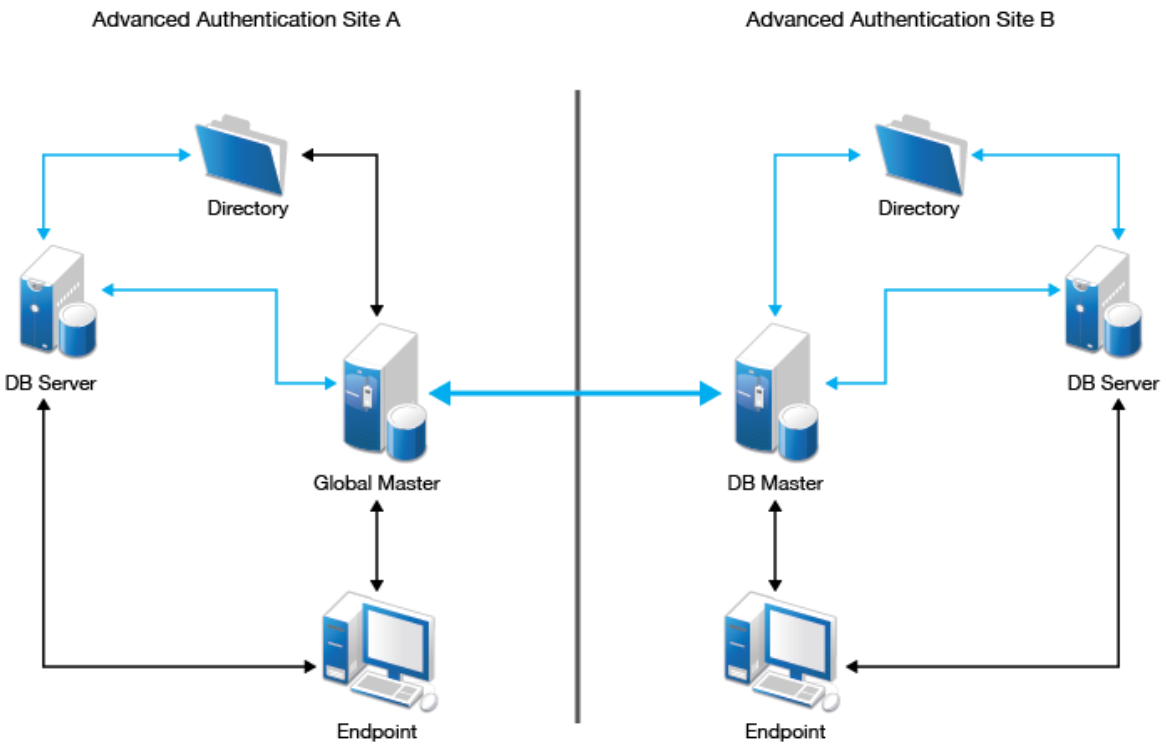
An Advanced Authentication server is connected to a directory such as Active Directory Domain Services, eDirectory, Active Directory Lightweight Directory Service or other compliant LDAP directories. An Event Endpoint can be Windows, Linux or Mac OS X machine, Access Manager, CloudAccess, or RADIUS Client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

1.4.2 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 19 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.

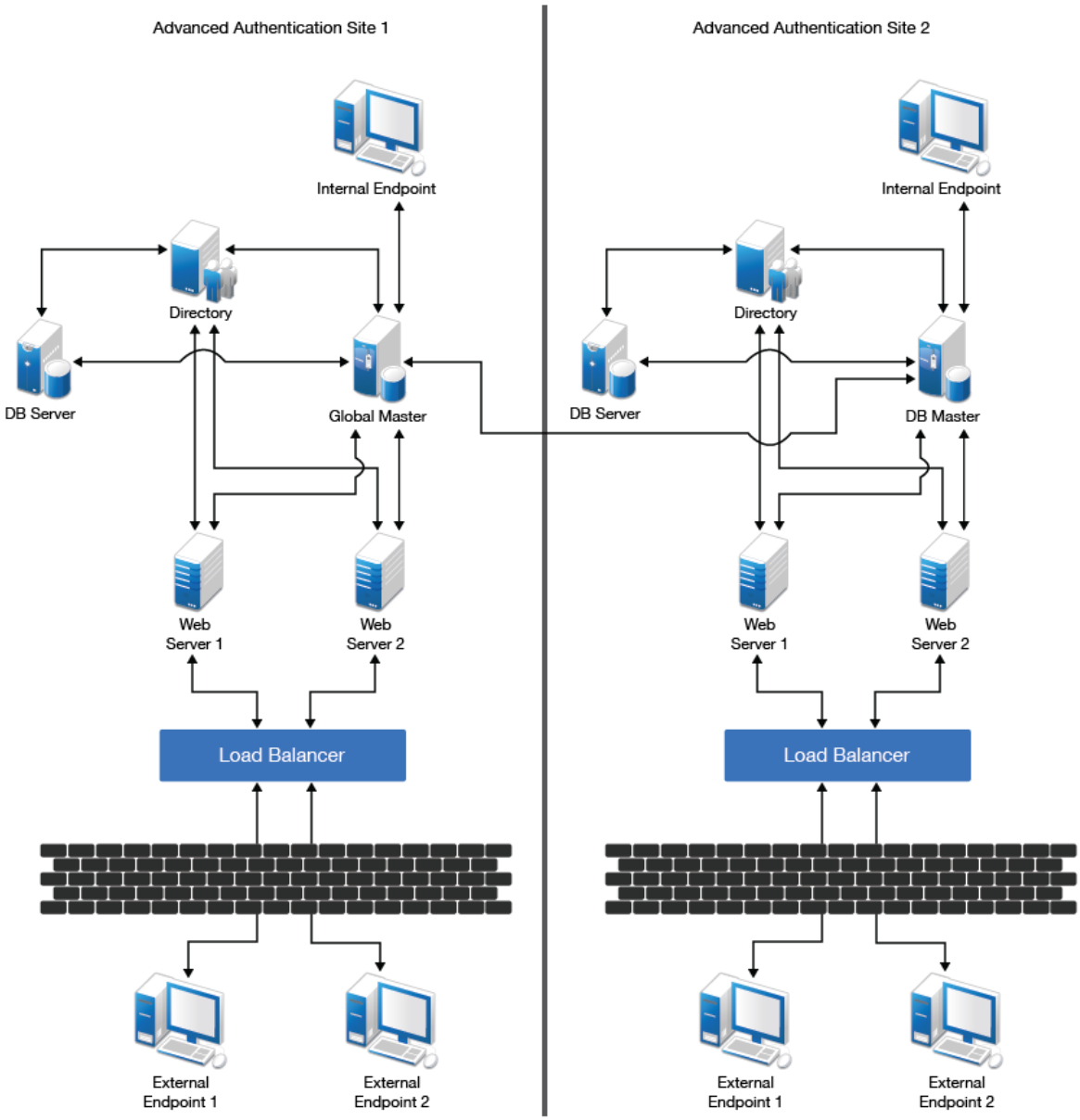
Master servers do not initiate a connection to the DB servers. Master servers initiate connection to Master servers only. DB servers initiate connection to the DB Master of the same site and Registrar only.

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.

1.4.3 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 21 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.

WARNING: Do not place the Advanced Authentication server in Demilitarized Zone (DMZ). It is recommended to use Load Balancer to process authentication requests from the external endpoints.

If a Global Master server (GMS) of a cluster goes down, the Web Servers of the primary site automatically communicate with the DB server of the primary site. When the GMS is up and running, Web Servers connects back to Global Master Server. This connection is established within a few of minutes. If the DB Master server of a secondary site goes down, the Web Servers of the same site communicates to the DB Server of the same site. When the DB Master is up, Web Servers connect back to it. While a GMS is down, the replication between sites fail. While a DB Master of a secondary site is down, the site does not replicate with the Global Master server.

1.5 Terminology

- Section 1.5.1, “Authentication method,” on page 22
- Section 1.5.2, “Authentication chain,” on page 22
- Section 1.5.3, “Authentication event,” on page 22
- Section 1.5.4, “Endpoint,” on page 22

1.5.1 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.

1.5.2 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.

1.5.3 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.

1.5.4 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 administration Portal,” on page 25
- Chapter 3, “End to end configuration with examples,” on page 27

2 Logging in to the Advanced Authentication administration Portal

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

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

1. Specify the administrator's credentials in the format: `repository\user` (**local\admin** 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, Canadian French, Chinese Simplified, Chinese Traditional, Danish, Dutch, English, French, German, Italian, Japanese, Polish, Portuguese (Brazilian), Russian, Spanish, Hebrew, and Swedish.

NOTE: If you are unable to log in to the Administration portal and an error message `LDAP connect error: invalidCredentials (result=49) 80090308: LdapErr: DSID-0C0903D3, comment: AcceptSecurityContext error, data 52e, v3839 (AuError)` is displayed then clear cookies or use Incognito mode in the browser.

IMPORTANT: Password of **local\admin** 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 End to end configuration with examples

This section describes how to configure Advanced Authentication with the following example scenarios:

- Section 3.1, “Implementing multi-factor authentication to VPN,” on page 27
- Section 3.2, “Securing Windows workstation with multi-factor authentication,” on page 33
- Section 3.3, “Configuring TOTP from OTP Desktop tool as one of the factors to access a corporate Portal,” on page 38
- Section 3.4, “Integrate Advanced Authentication and Office 365 without using AD FS,” on page 45
- Section 3.5, “Integrate Advanced Authentication and Office 365 using AD FS,” on page 50

3.1 Implementing multi-factor authentication to VPN

Let us assume Reltic Data, Inc. wants to implement multi-factor authentication for its VPN (Virtual Private Network) connection to secure the Corporate network that is accessed from their employees who are in a remote location.

This section explains the prerequisites, flow of actions, and step-by-step configuration details to achieve this.

This example refers to the following user profiles:

- Thomas: An administrator of Reltic Data, Inc.
- Mark Jones: An employee of Reltic Data, Inc.

Thomas, an administrator wants to enforce Multi-factor authentication with the LDAP Password and Smartphone methods for OpenVPN to secure the corporate network. After multi-factor authentication is implemented, employees need to authenticate to both methods successfully to access the network through VPN.

Thomas must perform the following tasks to implement multi-factor authentication for OpenVPN:

1. Add a Repository
2. Configure methods
3. Create a chain
4. Configure public external URLs policy
5. Assign chain to RADIUS server event
6. Configure the OpenVPN server

To understand the sequential flow of configuration in the Advanced Authentication Administration portal, see Configuration flow in Advanced Authentication for RADIUS server event.

For information about how an end user enrolls the configured methods and authenticates to VPN client using Advanced Authentication, see End user tasks.

3.1.1 Prerequisites

Ensure that you meet the following prerequisites:

- An LDAP repository for Reltic Data, Inc is configured and the repository contains the information of all users.

  This example uses **Active Directory Domain Services** as an **LDAP repository**.
- A group named **Employees** is created in Active Directory Domain Services.
- The Advanced Authentication server is installed. For more information, see Installing Advanced Authentication.
- A VPN client is installed on all employees' system.

  This example uses **OpenVPN** as the VPN client.

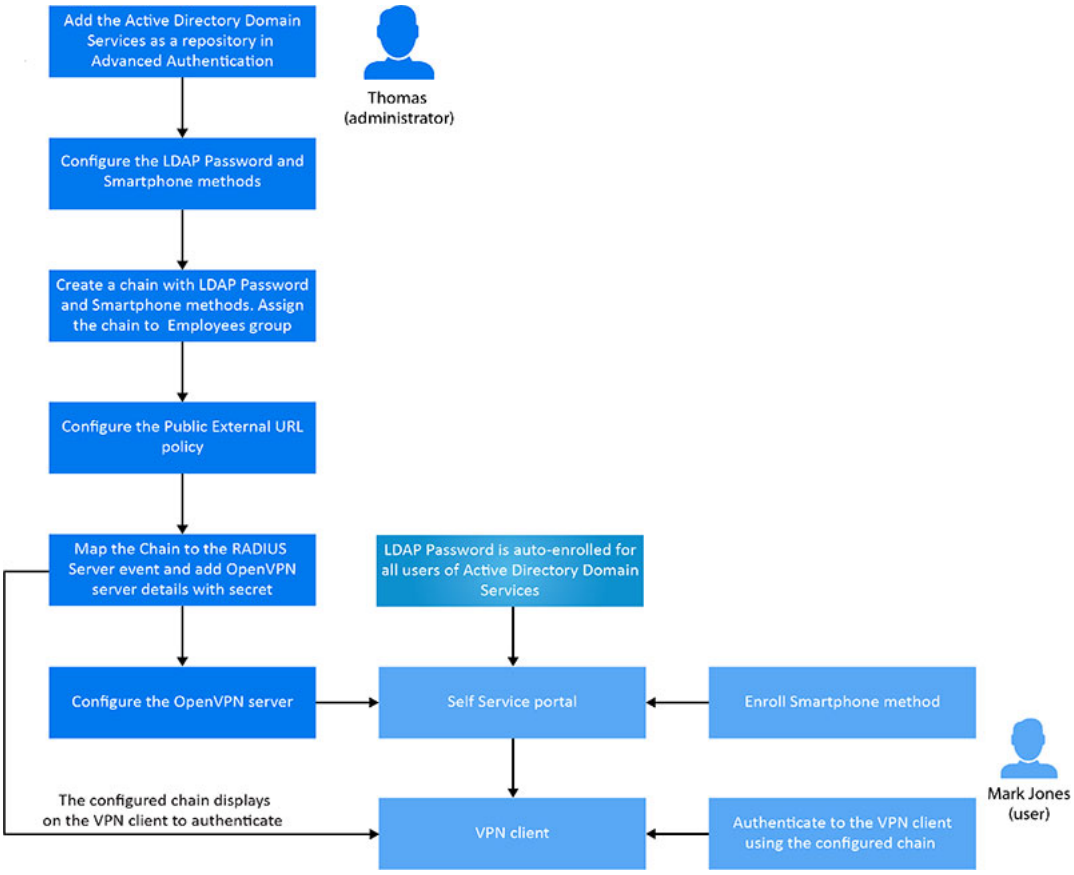
3.1.2 Considerations before configuration

Follow these guidelines to begin implementing multi-factor authentication for any event:

1. Identify the authentication methods that you want to configure.
2. Determine the order of methods in the chain. The methods are displayed to the end user in the order that you have configured.
3. Determine the policy that must be configured for the identified method.
4. Identify the user group for which you want to enforce this authentication chain.

Configuration flow in Advanced Authentication for RADIUS server event

The following diagram illustrates the sequential flow of actions required for securing the Open VPN client with multi-factor authentication:



3.1.3 Add a Repository

In Advanced Authentication, add Active Directory of Reltic Data, Inc. as a repository from where the user details are fetched for validation.

Perform the following steps to add Active Directory of Reltic Data, Inc. to Advanced Authentication:

1. Click **Repositories** on the Advanced Authentication Administration portal.
2. Click **Add LDAP repo**.
3. Select **AD** (Active Directory Domain Services) from the **LDAP type list**.
4. Specify a container for the users in **Base DN**. When you select the **Subtree** option, Advanced Authentication performs a search for the users in each child node. You can change the search scope by selecting the **Search one level only** option.
5. Specify a user account in **User** and specify the password of the user in **Password**.

   Ensure that the user's password has no expiry.
6. 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 each child node. You can change the search scope by selecting the **Search one level only** option.
7. Select **DNS discovery** to find LDAP servers automatically. Specify **DNS zone** and **Site name** (optional) and click **Perform DNS Discovery**.

   When the DNS discovery is done, the DNS servers list is updated every three hours.
8. Click **Save**.

3.1.4 Configure methods

The LDAP Password and Smartphone methods are configured with pre-defined values. These methods work as expected with the pre-defined values.

For more information, see LDAP Password and Smartphone.

3.1.5 Create a chain

Perform the following steps to create a chain with LDAP Password and Smartphone methods:

1. Click **Chains > Add** in the Advanced Authentication Administration portal.
2. Specify the following details:

Field	Description
Name	A name for the chain. **NOTE:** Ensure to remember the name of the chain for further use.
Short name	A name that is provided to end user for selecting a chain. For example, you configure a chain named SMS containing LDAP Password and SMS methods. A user can specify <username> sms and the user is required to use SMS as the chain. This is helpful in scenarios when the primary chain is not available.
Is enabled	Set to ON to enable the chain.
Methods	Select the LDAP Password and Smartphone methods to add to the chain.
Roles and Groups	Specify Employees. This enforces all users of this group to use this authentication chain for accessing the corporate network through VPN.

3. Click **Save**.
4. Continue with Configure public external URLs policy.

3.1.6 Configure public external URLs policy

The external URL manages the following activities for the Smartphone method:

- A push notification that is sent to the Advanced Authentication app.
- User responses from the Advanced Authentication app.

Perform the following steps to configure the external URL:

1. Click **Policies > Public External URLs** on the Advanced Authentication Administration portal.
2. Click **Edit** and specify the URL in **Public URL**.

   Ensure that the Public URL is accessible from users' smartphone.
3. Click **OK**.
4. Continue with Assign chain to RADIUS server event.

3.1.7 Assign chain to RADIUS server event

Perform the following steps to assign the chain to RADIUS Server event and configure details of the Open VPN server with a secret:

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 have created in Create a chain.
5. Click **Add** to add and assign a RADIUS Client to the event:
   - **5a** Specify the IP address of the OpenVPN server in **IP Address**.
   - **5b** Specify the OpenVPN server name in **Name**.
   - **5c** Specify the OpenVPN server secret and confirm the secret.

     NOTE: Ensure to make a note of this secret for future reference.

   - **5d** Ensure that the **RADIUS Client** is set to **ON**.
6. Click **Save**.
7. Continue with Configure the OpenVPN server.

3.1.8 Configure the OpenVPN server

Perform the following steps to configure the OpenVPN server and enable the server to connect with Advanced Authentication server:

1. Open the OpenVPN Access server site.
2. Click **Authentication > RADIUS**.
3. Enable the **RADIUS authentication**.
4. Select **PAP**.
5. Add an IP address of the Advanced Authentication server and specify the secret that is set while configuring the RADIUS Server event in the Advanced Authentication Administration portal.

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 appliance.

3.1.9 End user tasks

Mark, as an employee, must perform the following actions to access the corporate network of Reltic Data network through OpenVPN:

- “Enroll the Smartphone method” on page 32
- “Authenticate to OpenVPN using Advanced Authentication” on page 33

NOTE: The LDAP Password method enrolls automatically and users cannot remove it.

For more information, see LDAP Password.

Enroll the Smartphone method

Mark must ensure to install the OpenText Auth application on his smartphone to enroll the Smartphone method.

For more information about downloading and installing the OpenText Auth application, see Installing NetIQ Advanced Authentication App.

During the enrollment, Mark must scan a QR code that creates an authenticator on his mobile app. When Mark initiates the authentication, a push notification is sent to the app. Accept the request and get authenticated.

To enroll the Smartphone method with a QR code, perform the following steps:

1. Click the Smartphone icon under the **Add Authenticator** section of the Self-Service portal.
2. (Optional) Specify a comment related to the Smartphone authenticator.
3. (Optional) Select the required category from **Category**.
4. Click **Save**.

   A QR code is displayed.
5. Scan the QR code with the OpenText Auth app. To do this, perform the following steps:
   - **5a** Open the OpenText Auth app.
   - **5b** Specify a PIN if applicable.
   - **5c** Click the **+** (plus) icon in the **Enrolled Authenticators** screen.
   - **5d** The camera of your smartphone is launched.
   - **5e** Scan the QR code with the camera.

     A message `Authenticator "Smartphone" added` is displayed.
6. Specify the user name and an optional comment in the app.
7. Tap **Save**.

   The smartphone authenticator is created.

If Mark does not enroll the Smartphone method within few minutes, an error message `Enroll failed: Enroll timeout` is displayed. He can refresh the browser and try enrolling again.

TIP: If you users are unable to scan the QR code with the OpenText Auth app, they can do the following:

- Zoom the page to 125-150% and scan the zoomed QR code.
- Ensure that nothing overlaps the QR code (mouse cursor, text).

For more information, see Smartphone.

Authenticate to OpenVPN using Advanced Authentication

1 Launch the OpenVPN client.

The Login dialog is displayed.

2 Specify **Username** and **LDAP Password**.

3 Open the OpenText Auth app in the mobile phone.

Specify PIN or touch the enrolled finger that you registered for the app.

4 Tap **Accept** in the **Authentication Requests** screen to accept the authentication request.

A message `Accepted` is displayed and Mark authenticates to OpenVPN successfully.

3.2 Securing Windows workstation with multi-factor authentication

Let us assume Reltic Data, Inc. wants to implement multi-factor authentication for all Windows workstations to secure the data and provide authorized access to their employees.

This section explains the prerequisites, flow of actions, and step-by-step configuration details to achieve this.

This example refers to the following user profiles:

- Clarie Lee: An administrator of Reltic Data, Inc.
- Sussane Ross: An employee of Reltic Data, Inc.

Clarie, an administrator wants to enforce multi-factor authentication with the U2F and SMS OTP methods for the Windows login. After multi-factor authentication is implemented, employees must authenticate to both methods successfully to access the Windows workstation.

Clarie must perform the following tasks to implement multi-factor authentication for the Windows logon:

1. Add a Repository
2. Configure methods
3. Create a chain
4. Configure SMS sender policy
5. Assign chain to Windows Logon event

To understand the sequential flow of configuration in the Advanced Authentication Administration portal, see Configuration flow in Advanced Authentication for Windows Logon event.

For information about how an end user enrolls the configured methods and authenticates to the Windows workstation using Advanced Authentication, see End user tasks.

3.2.1 Prerequisites

Ensure that you meet the following prerequisites:

- An LDAP repository for Reltic Data, Inc is configured and the repository contains the information of all users.

  This example uses **Active Directory Domain Services** as an **LDAP repository**.
- A group named **Windows OS** is created in Active Directory Domain Services.
- The Advanced Authentication server is installed. For more information, see Installing Advanced Authentication.
- The Advanced Authentication Windows Client is installed on Windows workstation. For more information, see Installing Windows client.
- A DNS is configured to allow the Windows Client to discover and connect with the Advanced Authentication server. For more information, see Setting a DNS for Advanced Authentication Server Discovery.
- The Advanced Authentication Device Service is installed on the Windows workstation. For more information, see Installing and upgrading Device Service on Windows.
- An account for Reltic Data, Inc is registered with a SMS service provider that can deliver SMS OTP to users during authentication.

  This example uses **Twilio** as the SMS service provider.

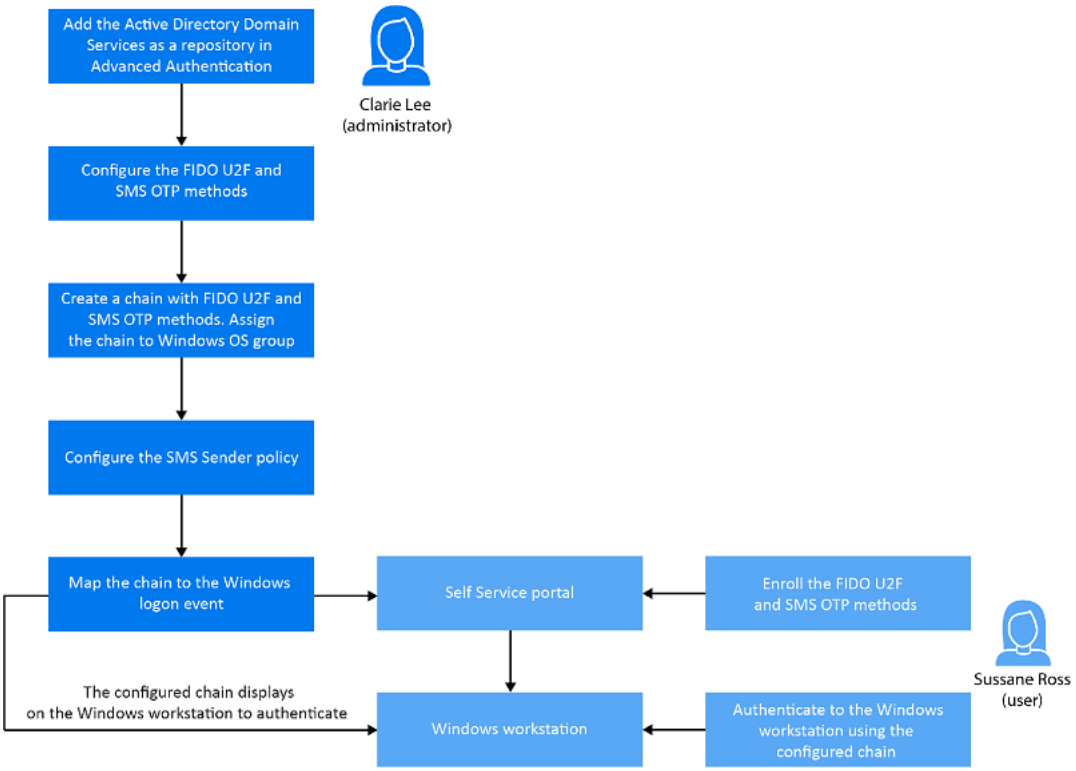
3.2.2 Points to consider before configuration

Consider the following guidelines before you begin implementing multi-factor authentication for the **Windows logon**:

1. Identify the authentication methods that you want to configure.
2. Determine the order of methods in the chain. The methods are displayed to the end user in the order that you have configured.
3. Determine the policy that must be configured for the identified methods.
4. Identify the user group for which you want to enforce this authentication chain.

Configuration flow in Advanced Authentication for Windows Logon event

The following diagram illustrates the sequential flow of actions required for securing the Windows workstation with multi-factor authentication:



3.2.3 Configure methods

Perform the following steps to configure the Password and SMS OTP methods:

1. Click **Methods** on the Advanced Authentication Administration portal.
2. Click the **Edit** icon corresponding to the U2F method.
3. Perform the following steps to configure the U2F method:
   - **3a** Set **Require attestation certificate** to **ON** to enable validation of the attestation certificate.
   - **3b** Select the attestation certificate:
     - **3b1** To use a default certificate, click **Add Default**.
     - **3b2** To use a custom certificate instead of predefined device manufacturer certificate, perform the following steps:
       - **3b2a** Click next to the default attestation certificate to remove the certificate.
       - **3b2b** Click **Add** to add a custom certificate.
       - **3b2c** Click **Browse** and select the custom certificate and click **Upload**.

         The certificate must be in the PEM format.
   - **3c** Click **Save**.
4. Configure the SMS OTP method.
   - **4a** Click the Edit icon corresponding to SMS OTP method.
   - **4b** Specify the following details to configure SMS OTP method:

Parameter	Description
OTP Period	The lifetime of an OTP in seconds. The default value is 120 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.
Allow overriding phone number	Set this option to **OFF** to prevent users to specify a different phone number during the enrollment. The option is set to ON by default.
Allow user enrollment without a phone	Set this option to **OFF** to ensure that a user does not enroll the SMS OTP authenticator without a phone. The user is prompted with an error message that you can specify in Error message.Set this option to **ON** to allow the user to enroll the SMS OTP authenticator without a phone.

4c Click **Save**.

5 Continue with Create a Chain.

3.2.4 Create a chain

Perform the following steps to create a chain with the U2F and SMS OTP methods:

1 Click **Chains > Add** in the Advanced Authentication Administration portal.

2 Specify the following details:

Field	Description
Name	A name for the chain. **NOTE:** Ensure to remember the name of the chain for further use.
Short name	This is not applicable for the Windows Client event. This is applicable only for the RADIUS Server event.
Is enabled	Set to **ON** to enable the chain.
Methods	Select the U2F and SMS OTP methods to add to the chain.
Roles and Groups	Specify Windows OS users. This enforces all users of this group to use this authentication chain for logging in to the Windows workstation.

3 Click **Save**.

4 Continue with Configure SMS Sender Policy.

3.2.5 Configure SMS sender policy

In Advanced Authentication, add Twilio details of Reltic Data, Inc. as a service provider that sends SMS OTP to the end users during authentication.

Perform the following steps to configure the details of Twilio in Advanced Authentication:

1. Click **Policies > SMS Sender** in the Advanced Authentication Administration portal.
2. Select **Twilio** in Sender service.
3. Specify the following details:
   - **Account sid** and **Authentication token**: In Twilio, the Account SID acts as a username and the Authentication Token acts as a password.
   - **Sender phone**: Sender's phone number.
4. 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!**.
5. Click **Save**.
6. Continue with Assign chain to Windows Logon event.

3.2.6 Assign chain to Windows Logon event

Perform the following steps to assign the chain to Windows logon event:

1. Click **Events**.
2. Click **Edit** next to the **Windows Logon** event.
3. Ensure that **Is enabled** is set to **ON**.
4. Select the chains that you have created in Create a Chain.
5. Click **Save**.

3.2.7 End user tasks

Sussane must perform the following tasks to authenticate to the Windows workstation with the configured methods:

- "Enroll the FIDO U2F method" on page 37
- "Enroll the SMS OTP methods" on page 38
- "Authenticate to the Windows workstation using Advanced Authentication" on page 38

Enroll the FIDO U2F method

1. Log in to the Advanced Authentication Self-Service portal.
2. Click the U2F icon in **Add Authenticator**.

   A message `Press button "Save" to begin enrolling.` is displayed.

3 (Optional) Specify a comment related to U2F in **Comment**.

4 (Optional) Select the preferred category from **Category**.

5 Click **Save**.

A message `Please touch the flashing U2F device now` is displayed. You may be prompted to allow the site permissions to access your security keys.

6 Touch the FIDO U2F button when there is a flash on the device.

A message `Authenticator "U2F" enrolled` is displayed. If there is no flash for more than 10 seconds, reconnect your token and repeat the steps.

Enroll the SMS OTP methods

NOTE: The SMS OTP method enrolls automatically if a phone number is specified in the user profile in Active Directory.

---

1 Click the SMS OTP icon in **Add Authenticator**.

2 (Optional) Specify a comment related to SMS OTP authenticator in **Comment**.

3 (Optional) Select the preferred category from **Category**.

4 Specify the mobile number in **Phone number**.

5 Click **Save**.

A message `Authenticator "SMS OTP" has been added` is displayed.

Authenticate to the Windows workstation using Advanced Authentication

1 Switch ON the Windows workstation.

The Sign in screen is displayed.

2 Specify **Username**.

Ensure the FIDO U2F device is plugged to the workstation.

3 Touch the FIDO U2F button when there is a flash on the device.

4 Specify the OTP that is sent to the phone.

Sussane gets authenticated to the Windows workstation successfully.

3.3 Configuring TOTP from OTP Desktop tool as one of the factors to access a corporate Portal

Let us assume an organization name Reltic Data, Inc. wants to use the Advanced Authentication OTP Desktop tool to generate time-based OTP. The generated OTP is used as one of the factors to access their corporate portal integrated with Advanced Authentication using SAML 2.0.

This section explains the prerequisites and step-by-step configuration details to achieve this.

This example uses the following user profiles:

- **Administrator**: Thomas is an administrator of Reltic Data, Inc.
- **End user**: Mark Jones is a software developer of Reltic Data, Inc.

Thomas, an administrator of Reltic Data, has identified the Card and TOTP methods for authenticating to the corporate portal. TOTP is generated using the Advanced Authentication OTP Desktop tool. This example uses Google Workspace as the corporate portal.

Thomas must perform the following tasks to integrate Google Workspace with Advanced Authentication and implement TOTP from OTP Desktop tool as one of the factors for Google Suite authentication:

1. Configure methods
2. Create a chain
3. Configure Google workspace
4. Create a SAML2 event
5. Configure web authentication policy
6. Generate and send an enrollment link to users

For information about how an end-user enrolls to configured methods, generates time-based OTP using Advanced Authentication OTP Desktop tool, and authenticates to the corporate portal, see "End user tasks" on page 43.

3.3.1 Prerequisites

Ensure that you meet the following prerequisites:

- An LDAP repository for Reltic Data, Inc is configured and the repository contains the information of all users.

  This example uses Active Directory Domain Services as an LDAP repository.
- A group, named **SAML Websites**, is created in Active Directory Domain Services. Add the users who must succeed the multi-factor authentication to log in to the corporate website to the group.
- The Advanced Authentication server is installed. For more information, see Installing Advanced Authentication.
- Add Active Directory of Digital Data, Inc. as a repository in Advanced Authentication from where the user details are fetched for validation. For more information, see Add a Repository.
- The Advanced Authentication OTP Desktop tool is installed on the Windows workstation. For more information, see Installing Desktop OTP tool.
- Identify and obtain ideal contactless card readers and cards for employees. Employee can use the card to enroll and authenticate to the Corporate Portal. For more information, see Supported card readers and cards.
- The Advanced Authentication Device Service is installed on the workstation. For more information, see Installing and upgrading Device Service.
- The parameters specific to the card reader are configured in the Device Service. For more information, see Configuring the card settings.

3.3.2 Configure methods

The Card method work as expected with the pre-defined value.

Perform the following steps to configure the TOTP methods:

1. Click **Methods > OATH OTP** on Advanced Authentication Administration portal.
2. Specify the following details in the TOTP section:

Parameter	Description
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 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 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 2*30 seconds before current time and 2*30 seconds after current time, which is ±2 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.

3. Click **Save**.
4. Continue with Create a chain.

3.3.3 Create a chain

Perform the following steps to create a chain with Card and TOTP methods:

1. Click **Chains > New Chain** in the Advanced Authentication Administration portal.
2. Specify the following details:

Field	Description
Name	A name for the chain. Note: Ensure to remember the name of the chain for further use.
Is enabled	Set to ON to enable the chain.
Methods	Select the Card and TOTP methods to add to the chain.
Roles and Groups	Specify SAML Websites. This enforces all users of this group to use this authentication chain for logging in to Google G Suite.

3. Click Save.
4. Continue with Create a SAML2 event.

3.3.4 Create a SAML2 event

1. Click **Events > New Event** in the Advanced Authentication Administration portal.
2. Perform the following to add a new event:
    1. Specify `Google` in **Name**.
    2. Select **SAML 2** in the **Event type**.
    3. Select the chain that you created in Create a chain.
    4. Click **Choose File** to upload the XML file that you fetched from Google.
    5. Set **Send E-Mail as NameID (suitable for G-Suite)** to **ON**. This is applicable for the Google Workspace.
    6. Click **Save**.
3. Continue with Configure web authentication policy.

IMPORTANT: By default, the OTP Desktop Event is set with either LDAP Password only and Password method. OTP Desktop Event supports a single-factor authenticator. User can use one of the methods to authenticate to the OTP Desktop tool.

3.3.5 Configure web authentication policy

1. Specify a valid DNS name of the Advanced Authentication server in the Identity Provider URL field for the SAML integration with Google Workspace.

    NOTE: You can download the SAML 2.0 metadata file only after specifying the Identity Provider's URL. The downloaded SAML 2.0 metadata file is used to configure the Service Provider.

    ---

2. Continue with Obtaining the signing certificate of Advanced Authentication.

3.3.6 Obtaining the signing certificate of Advanced Authentication

1. Click **Server Options** in the Advanced Authentication Administration portal.
2. Click **Signing Certificate** and save the certificate content in a notepad file for further use.
3. (Optional) To verify the integration, open the Google Sign-in page and specify an email address of the user from Basic information of the Google account (email address of Google account). Check whether Google redirects to the Advanced Authentication server, where the user must authenticate. After successful authentication, the Advanced Authentication server redirects the user back to Google.
4. Continue with Configure Google workspace.

3.3.7 Configure Google workspace

1. Login to the Google’s Administration console.

    NOTE: Sign in with an administrator account (doesn't end with gmail.com).

    ---

2 Open the **Security** section.

3 Expand **Set up single sign-on (SSO)**.

4 Enable **Setup SSO with third party identity provider**.

5 Specify the following parameters:

5a **Sign-in page URL**: `https://<AdvancedAuthenticationServerAddress>/osp/a/TOP/auth/saml2/sso`. Replace `AdvancedAuthenticationServerAddress` with the domain name or IP address of your Advanced Authentication server.

5b **Sign-out page URL**: `https://<AdvancedAuthenticationServerAddress>/osp/a/TOP/auth/app/logout`.

5c **Change password URL**: `https://<AdvancedAuthenticationServerAddress>` or Self-Service Password Reset URL.

5d Upload the Identity Provider Certificate that you saved in Step 2.

6 Clear **Use a domain specific issuer** if you have one domain in G Suite or select the option if you have more than one domain in G Suite.

Ensure that you have a user account in a repository that corresponds to a user account in Google. An email address specified in the **Contact information** for the Google account must be the same as an address from email attribute for the corresponding account of your repository.

NOTE: You cannot use the Google administrator account with SAML.

---

7 Create a new text file and add the Service Provider metadata to it. Following is the sample metadata:

```
<EntityDescriptor entityID="google.com"
xmlns="urn:oasis:names:tc:SAML:2.0:metadata"> <SPSSODescriptor
protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
<NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress</
NameIDFormat> <AssertionConsumerService index="1"
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://www.google.com/a/mycompany.com" /> </SPSSODescriptor>
</EntityDescriptor>
```

Replace `mycompany.com` in the Location URL to your primary domain from the **Domains** settings in Google.

NOTE: You must use the Service Provider metadata when one domain exists in the G Workspace. If you have more than one domain in G Suite, then every Service Provider metadata for each domain must have `google.com` as an entityID replaced with `google.com/mycompany.com`, where `mycompany.com` is your domain name.

---

8 Save the text file with a.`xml` extension.

9 Continue with Generate and send an enrollment link to users.

3.3.8 Generate and send an enrollment link to users

To generate an enrollment link, you can encode the server URL, tenant ID, and category name to the Base64 format using any online tool. The generated link is then sent to the users through the email to access the OTP Desktop tool and enroll the TOTP authenticator. The users can create an account on the tool to enroll the TOTP authenticator in the Self-Service portal.

To generate the enrollment link in the Base64 format, perform the following steps:

1. To encode use the details such as server URL, tenant ID and category name in the following JSON format:

   `{"server_url":"<domain-name>","tenant_name":"<tenant-name>","category_name": "HOME"}`

   For example, {"server_url": "aafserver.company.com", "tenant_name":"opentext", "category_name": "HOME"}

   You can specify the preferred category name for `category_name` parameter if you have added categories in the Event categories policy. You can remove the parameter `category_name`, if you have not added any category.

   In case of further problems with the enrollment link, please validate the syntax using Validating JSON syntax in SLAnalyzer.

2. Encode the value including {} to Base64 (charset: UTF-8) format.

   For example, the encoded link is displayed as:

   eyJzZXJ2ZXJfdXJsIjogImFhZnNlcnZlci5jb21wYW55LmNvbSIsICJ0ZW5hbnRfbmFtZSI6Im5ldGlx4oCdLCAiY2F0ZWdvcnlfbmFtZSI6ICJIT01FIn0=

3. Copy the encoded link for further use.

To send an enrollment link through email, perform the following steps:

1. Compose an email with the subject and body.

   For example, specify TOTP Enrollment Link in the Subject and body as follows:

   Hi Users, Click here to enroll for the TOTP authenticator using the OTP Desktop tool.

2. Right click on the preferred text and select **Hyperlink**.
3. Specify the encoded link and prefix `aaf-otp` in **Address**.

   For example, aaf-otp:eyJzZXJ2ZXJfdXJsIjogImFhZnNlcnZlci5jb21wYW55LmNvbSIsICJ0ZW5hbnRfbmFtZSI6Im5ldGlx4oCdLCAiY2F0ZWdvcnlfbmFtZSI6ICJIT01FIn0=

4. Specify the email address of the preferred users in **To** then click **Send**.

   User can click the hyperlink to open the OTP Desktop automatically.

3.3.9 End user tasks

Users must perform the following to authenticate to Google Suite with the configured methods:

- Enrolling card method
- Enrolling TOTP method using the OTP Desktop tool
- Authenticate to Google workspace

Enrolling card method

Before enrolling the Card authenticator, ensure that the card reader is connected to the computer.

1. Log in to the Advanced Authentication Self-Service portal.
2. Click the Card icon  in **Add Authenticator**.

   A message `Click "Save" to begin` is displayed.
3. (Optional) Specify a comment related to the Card authenticator in **Comment**.
4. (Optional) Select the preferred category from the **Category**.
5. Click **Save**.

   A message `Waiting for the card` is displayed.
6. Tap a card on the reader.

   A message `Authenticator "Card" has been added` is displayed.

Enrolling TOTP method using the OTP Desktop tool

Before enrolling the TOTP authenticator using the link, ensure that OTP Desktop tool is installed on your system.

1. Check your registered email or phone for the enrollment link.
2. Click on the link.

   You are directed to the OTP Desktop tool.
3. Specify your LDAP repository or local username, password and optional comment in the **OpenText AA OTP Desktop Tool** window.
4. Click **OK**.

   The TOTP authenticator is created in the OTP Desktop tool and enrolled in the Self-Service portal.

Authenticate to Google workspace

1. Open the Google Sign-in page in a browser.
2. Specify the email address of the Google account.

   The page redirects to the Advanced Authentication server authentication screen.

   Ensure the card reader is plugged into the workstation.
3. Tap the card on the reader.
4. Open the OTP Desktop tool.
5. Specify LDAP repository username and password.
6. Copy the OTP from the OTP Desktop tool.
7. Paste the OTP in **Password** of authentication screen.

   You are authenticated to Google Workspace successfully.

3.4 Integrate Advanced Authentication and Office 365 without using AD FS

Let us assume Reltic Data, Inc. wants to implement multi-factor authentication for their Office 365 without using Active Directory Federation Services (AD FS). Their employees must use the corporate email address and succeed the multi-factor authentication to access Microsoft Office 365 suite.

This section explains the prerequisites, flow of actions, and step-by-step configuration details to achieve this. This example refers to the following user profiles:

- Susan: An administrator of Reltic Data, Inc.
- Sam: An employee of Reltic Data, Inc.

Susan, the administrator, needs to enforce multi-factor authentication with the Card and Email OTP methods for Office 365. After multi-factor authentication is implemented, Sam the employee, needs to authenticate both methods to access Office 365.

- Section 3.4.1, “Prerequisites,” on page 45
- Section 3.4.2, “Administrator tasks,” on page 45
- Section 3.4.3, “End user tasks,” on page 49

3.4.1 Prerequisites

Ensure that you meet the following prerequisites:

- The Advanced Authentication server is installed. For more information, see Installing Advanced Authentication .
- Add Active Directory of Reltic Data, Inc. as a repository in Advanced Authentication from where the user details are fetched for validation. For more information, see Adding a Repository.
- Download Office 365 SAML metadata from Microsoft Online Service.
- Identify and obtain ideal contactless card readers and cards for employees. The employee can use the card to enroll and authenticate to the Office 365. For more information, see Supported card readers and cards.
- The Advanced Authentication Device Service is installed on the workstation. For more information, see Installing and Upgrading Device Service.

3.4.2 Administrator tasks

Susan, the administrator, needs to perform the following tasks:

- “Configure methods” on page 46
- “Create chain” on page 46
- “Create SAML2 event” on page 46
- “Configuring policies” on page 46
- “Configuring server option” on page 47
- “Enabling Single Sign-On to Microsoft Office 365” on page 47

Configure methods

1. Log in to Advanced Authentication Administration Portal as an Administrator.
2. The Card and Email OTP methods work as expected with the pre-defined value. For more information, see Card and Email OTP.

Create chain

Perform the following steps to create a chain with Card and Email OTP methods:

1. Click **Chains > New Chain** in the Advanced Authentication Administration portal.
2. Specify the following details:

Field	Action
Name	Specify the name for the chain. **NOTE:** Ensure to remember the name of the chain for further use. In this example, we named the chain Card+ Email OTP.
Methods	Select the Card and Email OTP methods to add to the chain.

3. Click **Save**.

Create SAML2 event

1. Click **Events > New Event** to add a new event.
2. Specify the following details:

Field	Action
Name	Specify a name for the event.
Event type	Select **SAML2**.
Chains	Select the required chains. In this example, we select Card+ Email OTP.

3. In **Upload SP SAML 2.0 metadata file**, click **Choose File** and upload the saved XML file.
4. Set **Send Immutable Id (User object Id) as Name ID (required for Microsoft Office 365)** to **ON**.
5. Click **Save**.

Configuring policies

Policies contain configuration settings for the Advanced Authentication methods, events, and so on. Perform the following steps to configure the policy:

Configuring web authentication policy

1 Click **Policies > Web Authentication**.

2 Specify a valid DNS name of an Advanced Authentication server in the Identity Provider URL field.

For example, `https://caf.realticsol.cf/`

3 Click **Save**.

Configuring mail sender policy

1 Click **Policies > Mail sender** to add a configure the Email OTP method

2 Specify the following details:

Field	Action
Host	Specify the outgoing mail server name.
Port	Specify the port number.
Username	Specify the username of an account that is used to send the authentication email messages.
Password	Specify the password for the specified account.
Sender email	Specify the email address of the sender.

3 Click **Save**.

Configuring server option

1 Open **Server Option**.

2 Click **Signing Certificate**.

3 Click Signing Certificate and save the certificate content for further use.

Enabling Single Sign-On to Microsoft Office 365

To enable single sign-on to Office 365, perform the following tasks:

Enabling directory synchronization in Office 365

1 Log in to the domain-joined computer where you have installed the following components:

- Microsoft Online Services Sign-in Assistant.
- Microsoft Azure Active Directory Module for Windows PowerShell.
- Azure AD Connect tool.

2 Launch **Azure AD Connect** on the domain-joined computer.

3 In **Express Settings**, click **Use express settings**.

4 In **User Sign-in**, select **Do not Configure**.

5 Click **Next**.

6 Specify the Azure AD global administrator credentials in **Connect to Azure AD**.

7 Click **Next**.

8 In **Identifying users**, select **Choose a specific attribute**.

9 Select **objectGUID**.

10 Verify the Active Directory Synchronization and activate the Office 365 licensing for the unlicensed but synchronized user

Federating the custom domain using Advanced Authentication

1 Launch Windows PowerShell.

2 Run the following command to connect to your Office 365 tenant:

```
Connect-MsolService
```

3 Specify the tenant administrator credentials of your office 365 domain.

4 Click **Sign in**.

5 Run the following command to verify whether your Office 365 domain is federated:

```
get-msoldomain -domain samplecompany.com
```

In this example, `get-msoldomain -domain realticsol.com`

In case the authentication type of your Office 365 domain is set to Federated, you must convert the authentication type to Managed using the following command:

```
Set-MsolDomainAuthentication –DomainName realticsol.com -Authentication
Managed
```

6 Run the following commands:

- `$dom="fully_qualified_domain_name"`

  In this example, `$dom="realticsol.cf"`.

- `$uri="https://AdvancedAuthenticationServerAddress/osp/a/TOP/auth/saml2/metadata"`

  In this example, `$uri="https://caf.realticsol.cf/osp/a/TOP/auth/saml2/metadata"`

- `$url="https://AdvancedAuthenticationServerAddress/osp/a/TOP/auth/saml2/sso"`

  In this example, `$url"https://caf.realticsol.cf/osp/a/TOP/auth/saml2/sso"`

- `$logoutUrl="https://AdvancedAuthenticationServerAddress/osp/a/TOP/auth/saml2/slo"`

  In this example, `$logoutUrl="https://caf.realticsol.cf/osp/a/TOP/auth/saml2/slo"`

- `$protocol="SAMLP"`

- `$cert=`"paste the signing certificate copied from **Server options** of Advanced Authentication."

7 Run the following command to convert your Office 365 domain to Federated authentication:

```
Set-MsolDomainAuthentication –DomainName $dom -Authentication Federated
-PassiveLogOnUri $url -IssuerUri $uri -LogOffUri $logoutUrl -
PreferredAuthenticationProtocol SAMLP -SigningCertificate $cert
```

8 Run the following command to verify the federation settings of your Office 365 domain:

```
Get-MsolDomainFederationSettings -domain samplecompany.com
```

In this example, `Get-MsolDomainFederationSettings -domain realticsol.cf`

3.4.3 End user tasks

Sam, the employee, must perform the following actions to access Office 365.

NOTE: The Email OTP method enrolls automatically. If you need to enroll with another email ID, see Email OTP.

Enrolling card method

Before enrolling the Card authenticator, ensure that the card reader is connected to the computer.

1 Log in to the Advanced Authentication Self-Service portal.

2 Click the Card icon in **Add Authenticator**.

A message `Click "Save" to begin` is displayed.

3 (Optional) Specify a comment related to the Card authenticator in **Comment**.

4 (Optional) Select the preferred category from the **Category**.

5 Click **Save**.

A message `Waiting for the card` is displayed.

6 Tap a card on the reader.

A message `Authenticator "Card" has been added` is displayed.

Authenticating on Office 365

1 Launch `http://office.com/`.

2 Click **Sign In**.

3 Specify the email address of the Office 365 account.

The page redirects to the Advanced Authentication server authentication screen.

NOTE: Ensure the card reader is plugged into the workstation.

---

4 Tap the card on the reader.

5 Check your email. You will receive an email with an OTP.

6 Specify the OTP from Email in **Password**.

7 Click **Login**.

3.5 Integrate Advanced Authentication and Office 365 using AD FS

Let us assume Reltic Data, Inc. wants to implement multi-factor authentication for their Office 365 using Active Directory Federation Services (AD FS). Their employees must use the corporate email address and succeed the multi-factor authentication to access Microsoft Office 365 suite.

This section explains the prerequisites, flow of actions, and step-by-step configuration details to achieve this.

This example refers to the following user profiles:

- Susan: An administrator of Reltic Data, Inc.
- Sam: An employee of Reltic Data, Inc

Susan, an administrator, wants to enforce multi-factor authentication with the Card and Email OTP methods for Office 365. After multi-factor authentication is implemented, Sam needs to authenticate both methods to access Office 365 successfully.

- Section 3.5.1, “Prerequisites,” on page 50
- Section 3.5.2, “Administrator tasks,” on page 51
- Section 3.5.3, “End user tasks,” on page 54

3.5.1 Prerequisites

Ensure that you meet the following prerequisites:

- The Advanced Authentication server is installed. For more information, see Installing Advanced Authentication.
- Add Active Directory of Reltic Data, Inc. as a repository in Advanced Authentication from where the user details are fetched for validation. For more information, see Add a Repository.
- Download the Office 365 SAML metadata from `https://<adfs_hostname>/FederationMetadata/2007-06/FederationMetadata.xml`.

  In this example, `https://adfs.saml.aaf-o365-int-tk/FederationMetadata/2007-06/FederationMetadata.xml`.
- Identify and obtain ideal contactless card readers and cards for employees. The employee can use the card to enroll and authenticate to the O365. For more information, see Supported Card Readers and Cards.
- The Advanced Authentication Device Service is installed on the workstation. For more information, seeInstalling and Upgrading Device Service.
- The parameters specific to the card reader are configured in the Device Service. For more information, see Configuring the Card Settings.

3.5.2 Administrator tasks

Susan, the administrator, needs to perform the following tasks:

- "Configure methods" on page 51
- "Create a chain" on page 51
- "Create SAML2 event" on page 51
- "Configuring policies" on page 52
- "Enable multi-factor authentication to Microsoft Office 365" on page 52

Configure methods

1. Log in to Advanced Authentication Administration Portal as an Administrator.
2. The Card and Email OTP methods work as expected with the pre-defined value.

   For more information, see Card and Email OTP.

Create a chain

Perform the following steps to create a chain with Card and Email OTP methods:

1. Click **Chains > New Chain** in the Advanced Authentication Administration portal.
2. Specify the following details:

Field	Action
Name	Specify a name for the chain. **NOTE:** Ensure to remember the name of the chain for further use. In this example, we named the chain as Card+ Email OTP.
Methods	Select the Card and Email OTP methods to add to the chain.

3. Click **Save**.

Create SAML2 event

1. Click **Events > New Event** to add a new event.
2. Specify the following details:

Field	Action
Name	Specify a name for the event.
Event type	Select **SAML2**.
Chains	Select the required chains. In this example, we select Card+ Email OTP chain.

3. Click **Choose File** and upload the saved XML file.

4 Set **Send Immutable Id (User object Id) as Name ID (required for Microsoft Office 365)** to **ON**.

5 Click **Save**.

Configuring policies

Policies contain configuration settings for the Advanced Authentication methods, events, and so on. Perform the following steps to configure the policy:

Configuring web authentication policy

1 Click **Policies > Web Authentication**.

2 Specify a valid DNS name of an Advanced Authentication server in the Identity Provider URL field.

For example, `https://caf.realticsol.cf/`

3 Click **Save**.

Configuring mail sender policy

1 Click **Policies > Mail sender** to add a configure the Email OTP method

2 Specify the following details:

Field	Action
Host	Specify the outgoing mail server name.
Port	Specify the port number.
Username	Specify the username of an account that is used to send the authentication email messages.
Password	Specify the password for the specified account.
Sender email	Specify the email address of the sender.

3 Click **Save**.

Enable multi-factor authentication to Microsoft Office 365

To enable single sign-on to Office 365, perform the following tasks:

Enabling directory synchronization in Office 365

1 1. Log in to the domain-joined computer where you have installed the following components:

- Microsoft Online Services Sign-in Assistant.
- Microsoft Azure Active Directory Module for Windows PowerShell.
- Azure AD Connect tool.

2 Launch Azure AD Connect.

3 In **Express Settings Wizard**, click **Use express settings**.

4 In **User sign-in**, select **Federation with AD FS**.

5 Click **Next**.

6 Specify the Azure AD global administrator credentials in **Connect to Azure AD**.

Wait to connect to Microsoft Online

7 Click **Add Directory**.

8 Select **Create new AD account**.

9 Specify the enterprise credentials and click **OK**.

10 In **Domain/OU Filtering**, select the following and click **Next**.

10a Select **Sync selected domains and OUs**.

10b Select only **O365**.

11 In **Credentials**, specify the domain administrator credentials and click **Next**.

12 In **AD FS Farm**, perform the following steps and click **Next**:

12a Click **Browse** and select the SSL certificate file from the local drive.

12b Specify the password for certificate.

13 In **Federation server,** add the server where to install AD FS click **Next**.

14 In **Service account**, specify the AD FS account credentials and click **Next**.

15 In **Azure AD Domain**, select your domain and click **Next**.

16 In **Ready to Configure**, click **Install**.

17 Verify the Active Directory synchronization.

Making the corresponding changes in ADFS

1 Open the ADFS management console.

2 Click **Claims Provider Trusts > Add Claims Provider trust**.

3 Click **Start**.

4 Click **Import data about the claims provider published online or on a local network**.

5 Specify federation metadata address.

In this example, `https://caf.realticsol.cf/osp/a/TOP/auth/saml2/metadata`.

6 Click **Next**.

7 Specify the **Display name**.

8 Click **Next**.

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

10 Click **Close**.

11 **Right-click the Display name** and select **Edit Claim Rules**.

12 Click **Add Rule**.

13 In **Claim rule template**, select **Send Claims Using a Custom**.

14 Click **Next**.

15 Specify the **Claim rule name**.

16 Paste the following in Custom rule:

```
c:[Type == "netbiosName"] => issue(Type = "http://
schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname",
Issuer = c.Issuer, OriginalIssuer = c.OriginalIssuer, Value = c.Value,
ValueType = c.ValueType);
```

17 Click **OK**.

3.5.3 End user tasks

Sam, an employee, must perform the following actions to access Office 365.

NOTE: The Email OTP method enrolls automatically. If you need to enroll with another email ID, see Email OTP.

- "Enrolling card method" on page 54
- "Authenticating on Office 365" on page 54

Enrolling card method

Before enrolling the Card authenticator, ensure that the card reader is connected to the computer.

1 Log in to the Advanced Authentication Self-Service portal.

2 Click the Card icon  in **Add Authenticator**.

A message `Click "Save" to begin` is displayed.

3 (Optional) Specify a comment related to the Card authenticator in **Comment**.

4 (Optional) Select the preferred category from the **Category**.

5 Click **Save**.

A message `Waiting for the card` is displayed.

6 Tap a card on the reader.

A message `Authenticator "Card" has been added` is displayed.

Authenticating on Office 365

1 Launch `http://office.com/`.

2 Click **Sign In**.

3 Specify the email address of the Office 365 account.

The page redirects to the Advanced Authentication server authentication screen.

NOTE: Ensure the card reader is plugged into the workstation.

---

4 Tap the card on the reader.

5 Check your email. You will receive an email with an OTP.

6 Specify the OTP from Email in **Password**.

7 Click **Login**.

II Configuring the Advanced Authentication settings

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.

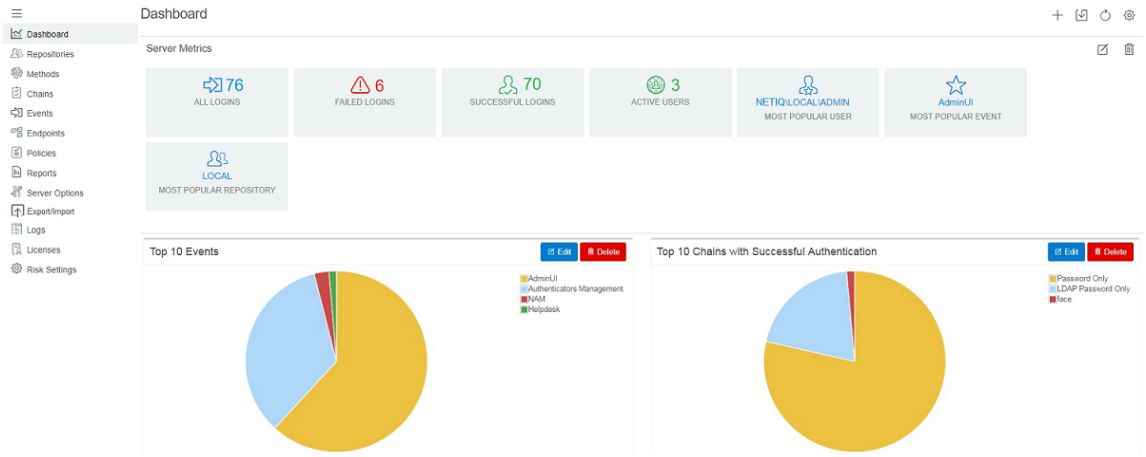
This section contains the following sections:

- Chapter 4, “Managing dashboard,” on page 57
- Chapter 5, “Managing Applications,” on page 65
- Chapter 6, “Adding a Repository,” on page 67
- Chapter 7, “Configuring methods,” on page 95
- Chapter 8, “Creating a chain,” on page 169
- Chapter 9, “Configuring events,” on page 173
- Chapter 10, “Managing endpoints,” on page 195
- Chapter 11, “Configuring policies,” on page 197
- Chapter 12, “Configuring the server options,” on page 251
- Chapter 13, “Adding a Report,” on page 253
- Chapter 14, “Enrolling the authentication methods,” on page 261
- Chapter 15, “Scripts option,” on page 263

4 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
- Export Widgets

4.1 Adding widgets

To add widgets, perform the following steps:

1. Click the **Add widget** icon + 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.
4. Click **OK**.

You can add the following types of widgets:

- Section 4.1.1, “Pie chart,” on page 58
- Section 4.1.2, “Stacked chart,” on page 58

- Section 4.1.3, “Activity stream,” on page 58
- Section 4.1.4, “Enroll activity stream,” on page 58
- Section 4.1.5, “Users,” on page 58
- Section 4.1.6, “Authenticators,” on page 59
- Section 4.1.7, “Event count line chart,” on page 59
- Section 4.1.8, “Events count line chart grouped by field,” on page 59
- Section 4.1.9, “Distinct events count line chart,” on page 59
- Section 4.1.10, “Distinct events count line chart grouped by field,” on page 59
- Section 4.1.11, “Server messages,” on page 60

4.1.1 Pie chart

This widget displays the information collected on a specific parameter and represents information in the Pie chart format. You can also sort the parameter is ascending and descending order.

4.1.2 Stacked chart

This widget displays a stacked bar chart that classifies and compares different categories of **Field 1** and **Field 2** parameters to track the maximum and minimum number of logons. X-axis represents categories of the **Field 2** parameter. Y-axis represents logon count. Segments in each vertical bar represents categories of **Field 1** parameter. Different colors are used to depict different categories and label for each category is displayed in upper-right corner of the widget.

4.1.3 Activity stream

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

4.1.4 Enroll activity stream

This widget displays information about enrolled users: last log on time, tenant, user, method used for authentication, and event type.

NOTE: The Enroll Activity Stream widget retrieves information about users who have authenticated at least once after the auto-enrollment and those who have never authenticated through any method.

4.1.5 Users

This widget displays information about the enrolled users: tenant name, user name, enrollment status and last log on time.

NOTE: The Users widget retrieves information about users who have authenticated at least once after the auto-enrollment and those who have never authenticated through any method.

4.1.6 Authenticators

This widget displays information about the enrolled authenticators: tenant name, user name, event category, method, comment and owner of the account.

NOTE: The Authenticators widget retrieves information about users who have authenticated at least once after the auto-enrollment and those who have never authenticated through any method.

4.1.7 Event count line chart

This widget tracks and displays logon count of all events in the appliance. The X-axis (horizontal) represents time and Y-axis (vertical) represents logon count. Each data point on the chart represents numbers of user logged on at a specific time. All the data points are plotted and connected with a line to track the maximum and minimum number of logons.

4.1.8 Events count line chart grouped by field

This widget tracks and displays logon count of specific parameter. The X-axis (horizontal) represents time and Y-axis (vertical) represents logon count. Data points of different colors represent specific category of the selected parameter. The label for each category is displayed in upper-right corner of the widget. All the data points are plotted and connected with a line to track the maximum and minimum number of logons.

4.1.9 Distinct events count line chart

This widget tracks and displays distinct count of all categories in the selected parameter (Distinct values by field). X-axis (horizontal) represents time and Y-axis (vertical) represents distinct logon count. Each data point on the chart represents unique logon count at a specific time. All the data points are plotted and connected with a line to track the maximum and minimum number of distinct logons.

For example: If the Distinct events count line chart widget is customized as follows:

- **Interval** set to **1 hour**.
- **Distinct values by field** is set to **User name**.

The widget displays number of unique users logged in to all events for the time duration of 1 hour.

4.1.10 Distinct events count line chart grouped by field

This widget displays and classifies distinct logon count of each event. The X-axis (horizontal) represents time and Y-axis (vertical) represents distinct logon count. Each data point on the chart represents unique logon count of particular event at a specific time. All the data points are plotted and connected with a line to track the maximum and minimum number of distinct logons to particular event.

4.1.11 Server messages

This widget displays a message describing the low disk space condition along with the severity.

The Advanced Authentication records the event and checks for insufficient disk space. If the disk space is less than 5% of its disk capacity, the widget displays the following message:

```
AA events are not being recorded due to insufficient disk space. Please
free up disk space and event recording will resume within 5 minutes. Free
disk space must be greater than 5 percent of disk capacity.
```

4.2 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 the **Delete** icon on the top of each widget.

To edit the widget, click the **Edit** icon on the top of each widget.

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

4.3 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, perform the following steps:

1. Click the **Dashboard Settings** icon on the upper-right corner of the dashboard.
2. Ensure **Relative time interval** is set to **ON** in the **Dashboard Settings** window.
3. Select the time interval from **Relative interval**. By default, time interval is set to **Last 15 minutes**.
4. Click **Update**.

Viewing dashboard for previous records

To view previous records, perform the following steps:

1. Click the **Dashboard Settings** icon on the upper-right corner of the dashboard.
2. Set **Relative time interval** to **OFF** in the **Dashboard Settings** window.
3. Select the **Date range**.
4. Click **Update**.

4.4 Customizing the default widgets

To customize the widget, click **Edit** and select the appropriate filters. You can customize the display based on the following filter factors:

- **Title**: Specify preferred title for the widget.
- **Event type**: Select preferred event type. Options available are **All logon events**, **Failed logon events** and **Successful logon events**.
- **Interval**: Select Time interval.
- **Size**: Select number of records.
- **Sort**: Select sorting order. Options available are ascending or descending order.
- **Field**: Select the parameter based on which the data must be collected to display on the widget. Options available are **Event Name**, **Chain Name**, **Method Name**, **Endpoint Name** and so on.
- **Users**: Select specific user.
- **Events**: Select specific event.
- **Chains**: Select specific chain.

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

- Section 4.4.1, “Server metrics,” on page 62
- Section 4.4.2, “Tenants,” on page 62
- Section 4.4.3, “Billing,” on page 62
- Section 4.4.4, “Logons per result,” on page 62
- Section 4.4.5, “Total users,” on page 62
- Section 4.4.6, “Total users per event,” on page 62
- Section 4.4.7, “Activity stream,” on page 62
- Section 4.4.8, “Successful/Failed Logons,” on page 62
- Section 4.4.9, “Top events with successful Logon per chain,” on page 62
- Section 4.4.10, “Top events with failed Logon per method,” on page 63
- Section 4.4.11, “Top 10 events,” on page 63
- Section 4.4.12, “Top 10 chains with successful result,” on page 63
- Section 4.4.13, “Top 10 servers,” on page 63
- Section 4.4.14, “Top 10 tenants,” on page 63
- Section 4.4.15, “Top 10 Repositories,” on page 63
- Section 4.4.16, “Top 5 events for Logons,” on page 63
- Section 4.4.17, “Top 5 users for Logons,” on page 63
- Section 4.4.18, “Top 10 users with failed Logon,” on page 63
- Section 4.4.19, “Top 10 users,” on page 63
- Section 4.4.20, “Top 10 methods with failed result,” on page 63

4.4.1 Server metrics

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

- **All Logins**: Total number of logins.
- **Failed Logins**: Total number of failed logins by the users.
- **Successful Logins**: Total number of successful logins by the users.
- **Active Users:** 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 Repository**: The repository that users have used the most.

4.4.2 Tenants

This widget displays information about the tenants and their login.

4.4.3 Billing

This widget displays the unique user logon count in the selected period.

4.4.4 Logons per result

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

4.4.5 Total users

This widget displays the total number of logged in users for time interval.

4.4.6 Total users per event

This widget displays the total number of logged in users for each event.

4.4.7 Activity stream

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

4.4.8 Successful/Failed Logons

This widget displays information about the successful or failed users login.

4.4.9 Top events with successful Logon per chain

This widget displays the top events based on the successful logon for each chain.

4.4.10 Top events with failed Logon per method

This widget displays the top events based on the failed logon for each chain.

4.4.11 Top 10 events

This widget displays the top ten events the user has performed.

4.4.12 Top 10 chains with successful result

This widget displays the top ten chains the user has successfully authenticated with.

4.4.13 Top 10 servers

This widget displays the top ten servers the user has used to authenticate.

4.4.14 Top 10 tenants

This widget displays the top ten tenants.

4.4.15 Top 10 Repositories

This widget displays the top ten repositories.

4.4.16 Top 5 events for Logons

This widget displays the top five events for login.

4.4.17 Top 5 users for Logons

This widget displays the top five users for login.

4.4.18 Top 10 users with failed Logon

This widget displays the top ten users who have failed in the login attempt.

4.4.19 Top 10 users

This widget displays the top ten users.

4.4.20 Top 10 methods with failed result

This widget displays the top ten methods with failed authentication results.

4.5 Exporting widgets

When you export a widget, Advanced Authentication creates a copy of the selected widget in the **Reports** section. You must navigate to **Reports** page to download the exported file on your local drive.

To export a widget, perform the following steps:

1. Select the preferred widget on the **Dashboard** page.
2. Click the **Export** icon  and select preferred format. Formats available are:
   - .csv
   - .json
3. Click **Reports**.
4. Click the exported file name in the **Exported reports** section, to download on the local drive.

5 Managing Applications

Advanced Authentication employs Single Sign-on to facilitate users a secure SSO access to different federated applications. OpenText Single Sign-on is a comprehensive access management solution that provides single sign-on and secure access to web-based applications, cloud applications, and federated business-to-business interactions. This service provides authorized users with adaptive, context-aware secure access to web and cloud applications from anywhere and from any device.

Single Sign-on is an add-on solution for Advanced Authentication. Single Sign-on creates a module, **Applications** in the Advanced Authentication Administration Portal. Using industry standards such as Security Assertion Markup Language (SAML) and OAuth, you can create a trusted connection. This trusted connection between your organization's systems and the applications or resource simplifies the end-user's log-in experience.

NOTE: The **Applications** module appears if you have a valid subscription for Advanced Authentication.

For information about...	See
Single Sign-on and its benefits	Welcome to Single Sign-on (https://www.microfocus.com/documentation/single-sign-on/help/single-sign-on-admin/welcome.html)
To create an Application	Creating an Application (https://www.microfocus.com/documentation/single-sign-on/help/single-sign-on-admin/walk-through.html)

6 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, eDirectory, Active Directory Lightweight Directory Services, OpenLDAP, and OpenDJ. Advanced Authentication also supports the MSSQL database.

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.

You can add the following repositories:

- Section 6.1, "Adding an LDAP Repository," on page 67
- Section 6.2, "Adding an SQL database," on page 79
- Section 6.3, "Adding a Cloud Bridge external Repository," on page 80
- Section 6.4, "Local Repository," on page 92
- Section 6.5, "Adding a SCIM managed Repository," on page 93

6.1 Adding an LDAP Repository

IMPORTANT: The LDAP Repository is not available in Advanced Authentication as a Service (SaaS) version.

To add a repository, perform the following steps:

1. Click **Repositories > New LDAP repo**.
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 eDirectory

     NOTE: When eDirectory is used as the LDAP repository then ensure Linux PAM Client's realm name matches the repository name for SSH logins to work properly.
   - **Other** for OpenLDAP, OpenDJ and other types

     To add the AD LDS repository with the AD LDS proxy, see Adding an AD LDS Repository with the configured AD LDS proxy.

   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.

NOTE: Make sure that you must re-enter the password every time you make changes.

---

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**, you can perform the DNS discovery either automatically or manually.

- Automatically performing the DNS discovery
- Manual DNS Discovery

Automatically performing the DNS discovery

1. Select **DNS discovery** in the **LDAP servers** option.
2. Specify the **DNS zone**.
3. Specify the **Site name** (optional).
4. The **Use SSL** option is set to **OFF** by default. This indicates that the DNS discovery is done on a non-SSL mode for the port 389. An `_ldap` SRV record is retrieved from the DNS server when this option is disabled. For example, `_ldap._tcp.test2.local2`.

   To use SSL for DNS discovery on port 636, turn **Use SSL** to **ON**. An `_ldaps` SRV record is retrieved from the DNS server. For example, `_ldaps._tcp.test2.local2`. However, administrators must create the SRV record on the DNS server before using the SSL option.
5. Click **Perform DNS Discovery**.

   When the DNS discovery is done, the DNS servers list is updated every three hours.

Manually performing the DNS discovery

1. Select the **Manual setting** option in the **LDAP servers** option to add LDAP servers manually.
2. 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.
3. Specify an LDAP server's **Address** and **Port**.
4. Turn **SSL** to **ON** to use SSL (if applicable).

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.

NOTE: If you have a domain per-site architecture, the Global Master Server must have a connection at least to one LDAP server from each site. This is required because the Global Master Server must have access to all domains. In the secondary sites, ensure that the LDAP servers list contains only local LDAP servers to prevent an Advanced Authentication server to communicate to a remote LDAP server. This is because communication to servers that are located far may result in delays.

For example, suppose you have the `company.com` domain at the primary site. Also, there are few child domains, located at other sites such as `my1.company.com` and `my2.company.com`. If you will put only LDAP servers from `company.com` to repository configuration, this will mean there is no sync possible with LDAP servers that belong to the child domains.

It is necessary to put the local and at least one LDAP server from each child domain on the Global Master Server to allow synchronization with those child domains.

---

5. Click the save icon next to server's credentials.

   Add additional servers (if applicable).

7 (Conditional) To configure custom attributes, expand Advanced settings. The Advanced Settings are required for OpenDJ, OpenLDAP, and in some cases for eDirectory.

8 Click **Save**.

NOTE: If you use 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 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.

---

9 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.

10 To verify the synchronization of a repository, click **Edit** and you can view the information in **Last sync**.

11 Click **Full synchronization** to perform a complete synchronization of the repository.

NOTE: Full synchronization must be initiated only on the Global Master server.

Advanced Authentication performs automatic synchronization of only the modified user attributes (fast synchronization) on an hourly basis. The fast sync is supported for AD repositories only.

6.1.1 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 70
- "User name attributes" on page 70
- "User mail attributes" on page 70
- "User cell phone attributes" on page 71
- "User ID/Passport number attributes" on page 71
- "User social security number attribute" on page 71
- "Group lookup attributes" on page 71
- "Group name attributes" on page 71
- "Verify SSL certificate" on page 72
- "Enable paged search" on page 72
- "Enable nested groups support" on page 73
- "Framed IPv4 address attribute" on page 73
- "Custom attributes to fetch" on page 73
- "Custom attributes to return" on page 74
- "Used attributes" on page 74

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.

User ID/Passport number attributes

Advanced Authentication validates the specified attributes to retrieve the national ID or passport number of users. These attributes are used for the HANIS method.

User social security number attribute

Advanced Authentication validates the specified attributes to retrieve the Social Security number of users. These attributes are used for the Danish National ID method. As the AD repository does not have a Social Security number attribute, you can specify the Social Security number in any other attribute and specify the same attribute name in this field.

For example, If you register your Social Security number in the Pager number attribute in AD, you can specify **Pager** in this field. Then, the Advanced Authentication checks the Pager attributes to retrieve the Social Security number.

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
PKI card certificate id attribute	altSecurityIdentities **NOTE:** The altSecurityIdentities attribute must be mapped to certificate mapping type **X509IssuerSerialNumber** to auto-enroll PKI method for a user.	
User Object Class	user	OpenDJ and OpenLDAP: person
Group Object Class	group	OpenDJ: groupOfNames OpenLDAP: posixGroup
Group Member Attribute	member	OpenDJ: member OpenLDAP: memberUid. If a required group contains groupOfNames class, disable **POSIX style groups**. If the group contains posixGroup, enable **POSIX style groups.** ◆ User UID attribute This attribute is available only when **POSIX style groups** is **ON**.Default value: uid.
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 **Browse** 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 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.

Custom attributes to fetch

Custom Attributes to Fetch contains the list of attributes that should be requested from the repository during authentication. In the case of RADIUS Server events, it's possible to configure rules based on the RADIUS Options.You can use custom attributes for SAML 2.0 integrations also. In this case, every attribute must be also added to Custom Attributes to Return list. If an attribute contains multiple values, the top value is used.

Custom attributes to return

Custom Attributes to Return contains the list of attributes that should be returned to the REST API clients after successful authentication. For SAML 2.0 integrations, only the LDAP attributes are supported. In this case, every attribute must be also added to Custom Attributes to Fetch list. If an attribute contains multiple values, the top value is used.

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 (https://docs.microsoft.com/en-us/windows/win32/adschema/s-enumeration?redirectedfrom=MSDN)	×	×	?
Object Class	`Object Class`	An unordered list of object classes	String	?	?	?
Member	`Member`	A list that indicates the objects associated with a group or list	String	?	?	?

Attribute Name	LDAP Name	Description	Type	Supported in Active Directory	Supported in LDS	Supported in eDirectory
User-Account-Control	`userAccountControl`	Flags that control the behavior of a user account	Enumeration (https://docs.microsoft.com/en-us/windows/win32/adschema/s-enumeration?redirectedfrom=MSDN)	?	×	×
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 (https://docs.microsoft.com/en-us/windows/win32/adschema/s-enumeration?redirectedfrom=MSDN)	?	?	×
Primary-Group-ID	`primaryGroupID`	A relative identifier (RID) for the primary group of a user	Enumeration (https://docs.microsoft.com/en-us/windows/win32/adschema/s-enumeration?redirectedfrom=MSDN)	?	×	×
Object-Guid	`objectGUID`	A unique identifier for an object	Octet String (https://docs.microsoft.com/en-us/windows/win32/adschema/s-string-octet?redirectedfrom=MSDN)	?	?	×

Attribute Name	LDAP Name	Description	Type	Supported in Active Directory	Supported in LDS	Supported in eDirectory
object-Sid	`objectSid`	A Binary value that specifies the security identifier (SID) of the user	Octet String (https://docs.microsoft.com/en-us/windows/win32/adschema/s-string-octet?redirectedfrom=MSDN)	?	?	×
Logon-Hours	`logonHours`	Hours that the user is allowed to logon to the domain	Octet String (https://docs.microsoft.com/en-us/windows/win32/adschema/s-string-octet?redirectedfrom=MSDN)	?	×	×
USN-Changed	`uSNChanged`	An update sequence number (USN) assigned by the local directory for the latest change including creation	Interval (https://docs.microsoft.com/en-us/windows/win32/adschema/s-interval?redirectedfrom=MSDN)	?	?	×

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\bf\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']
```

Search groups

```
(&(objectClass=group)(GUID=<group_GUID>))
```

Requested attributes:

```
['cn', 'objectClass', 'GUID', 'loginDisabled', 'loginExpirationTime',
'lockedByIntruder', 'radiusFramedIPAddress']
```

6.1.2 Adding an AD LDS Repository with the configured AD LDS proxy

1. Click **Repositories > New LDAP repo**.
2. Select **Other** from the **LDAP type** list.
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. Under Advanced Settings, specify `objectGUID` as object ID attribute and `userProxy` as user class.
7. Click **Save**.

NOTE: The drawback of this solution is that Advanced Authentication server does not validate the user attributes (account disabled, account locked out, and so on). This solution is beneficial when the users log on using the chain that does not include the LDAP password method (for example, CARD + PIN). However, LDS validates the user attributes in both the above scenarios when the LDAP password method is in use.

6.2 Adding an SQL database

IMPORTANT: The SQL Database is not available in Advanced Authentication as a Service (SaaS) version.

You can add an MSSQL database to be consumed as a repository by Advanced Authentication. The following version of SQL servers are supported:

- Microsoft SQL Server 2016

To add an SQL database, perform the following steps:

1. Click **Repositories > New SQL repo**.
2. Specify the following details of the SQL database:
   - **Name**: Name of the repository.
   - **Database type**: Select the preferred database from the list:
     - **MSSQL**
     - **MYSQL**
     - **POSTGRESQL**
   - **DB host**: Specify host in one of the following syntax based on the type of DB:
     - MSSQL: `<IP address>:<Port number>`

       IP address of the database host with the port number used for remote sharing
     - POSTGRESQL and MYSQL: `<IP address>`
   - **DB name:** Name of the database.
   - **DB user**: Name of the database user.
   - **Password**: Password of the database.
   - **Table or view name**: Name of the table or view in the database.
   - **User's id column** and **User's id type**: User's id column and id type in the database.
   - **User's name column** and **User's name type**: The username column and the type in which the name is specified.
   - **User's phone column** and **User's email column**: The phone and email column in the database.

IMPORTANT: Remember the following points, while configuring a SQL database:

- The LDAP Password method is not applicable for the users in SQL repository. The Password method for the users is not enrolled automatically and can be enrolled manually by the Helpdesk administrator only.
- You must disable the **Ask credentials of management user** in the Helpdesk Options policy for the SQL repository. This enables the helpdesk administrator to set an authenticator for a user, without getting authenticated with the user's password on the **User to Manage** page of the Helpdesk portal.
- The SQL repository supports auto enrollment of Email OTP, SMS OTP, and Voice OTP methods. If you use only these methods, you can create a chain with one or some of these methods. You do not need the Helpdesk administrator's assistance for the enrollment of these methods. It is not recommended to use a single factor chain with only one of these methods as it is not secure.

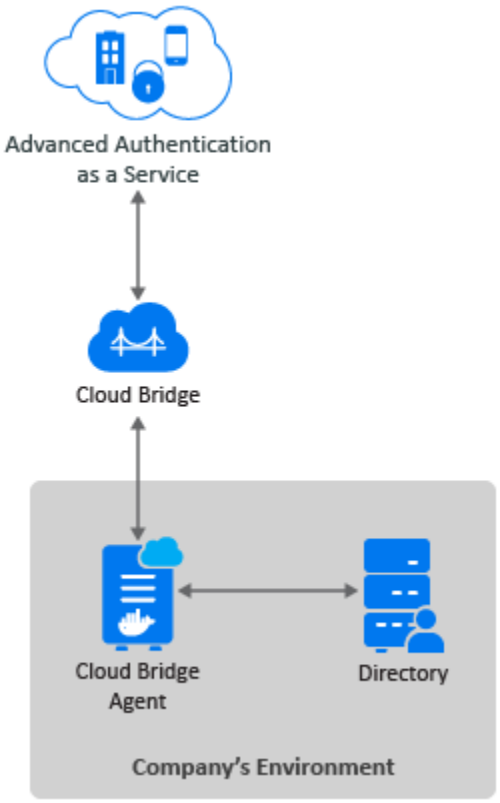
6.3 Adding a Cloud Bridge external Repository

IMPORTANT: The Cloud Bridge external repository is applicable only for Advanced Authentication as a Service (SaaS).

In Advanced Authentication as a Service environments, Cloud Bridge act as an identity transfer bridge between Advanced Authentication in the cloud and data sources in on-premises environments. The Cloud Bridge retrieves the identify information from the on-premises repositories and makes this data available periodically or on-demand requests to Advanced Authentication.

The Cloud Bridge Agent is the entity that responds to the Advanced Authentication collection and fulfillment commands and directs them to the proper data source for execution. To collect data from multiple on-premises repositories, you need to install a Cloud Bridge Agent in each on-premises repositories.

To learn the benefits of Cloud Bridge, see Understanding the Benefits of Cloud Bridge (https://www.microfocus.com/documentation/identity-and-access-management/iam-services/cloud-bridge-agent-admin/).



Repositories

To add a Cloud Bridge external repository, perform the following steps:

Before adding a new Cloud Bridge repository, ensure that you have the privilege to use the Cloud Bridge. Once you have the privilege to use Cloud Bridge, the Cloud Bridge Client(s) list will be displayed in the **Client** as in Step 6b. If you do not have the privilege to use Cloud Bridge, wait for the Open Text operations team (Center of Excellence or CoE) to entitle you or contact CoE.

1. Click **Repositories > New Cloud Bridge External repo**.
2. Select an applicable repository type from the **LDAP type** list. The supported options are:
   - **AD** for Active Directory Domain Services.
   - **eDirectory** for eDirectory.

3 Specify the name of the repository in **Name**.

NOTE: For AD repositories, the name of the repository must correspond to the domain NetBIOS name.

---

4 Specify a container for the users in **Base DN**. When you select the **Search full 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.

5 You can specify a container for the groups in **Group DN (optional)**. When you select the **Search full 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 Select the preferred **High Availability Mode** for distributing the load and to achieve uninterrupted synchronization of user identities from LDAP servers. The default option is **Round Robin**. Available options are:

- **Round Robin**: This approach allows Advanced Authentication Server to distribute the incoming request with available LDAP servers. This provides load balancing and if an LDAP server is not available then it will be skipped.
- **Failover**: In this approach, first available LDAP Server in the list gets all incoming request, in case the first LDAP Server is unavailable, then all requests are directed to the subsequent LDAP server in the list.

7 Add external server configurations:

7a Click **Add Server**.

7b Select the required client URL from **Client**.

7c Select the required data center from **Data Center**.

This list allows to add multiple domains.

7d Specify a local IP address of the LDAP server in **LDAP server**.

7e Specify the port number of the server in **Port**. For example, 389.

7f (Optional) Enable **SSL** to ensure that the LDAP connection to the appliance is secured with a valid self-signed SSL certificate. This helps to prevent any attacks on the LDAP connection and ensures safe authentication.

NOTE: If **SSL** is enabled, you need to upload the LDAP CA certificate.

7g Click the save  icon next to the server credentials.

8 If **SSL** is enabled for the external server, click **Choose File** in **LDAP CA certificate**, and select the certificate file from the local drive.

9 Open **Agents and Clients** if you need to view the following details:

- **Agent ID**: It lists the available `datacenter.json` files. You can select the required Agent ID to view the corresponding `datacenter.json` information.
- **Client URL:** Select the required Client URL to view the available Client URL.

10 Click **Save**.

To understand the prerequisites and install procedure of installing Cloud Bridge Agent, see Installing the Cloud Bridge Agent (https://www.microfocus.com/documentation/identity-and-access-management/iam-services/cloud-bridge-agent-admin/install-cba.html).

You can perform the following to manage the Cloud Bridge external repository:

- Synchronizing Cloud Bridge Repository
- Testing Cloud Bridge
- Force configuring Cloud Bridge
- Enabling fast synchronization for eDirectory Repository

6.3.1 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:

- "Fast sync enabled" on page 83
- "Time between fast syncs" on page 84
- "User lookup attributes" on page 84
- "User name attributes" on page 84
- "User mail attributes" on page 84
- "User cell phone attributes" on page 84
- "Group lookup attributes" on page 85
- "Group name attributes" on page 85
- "Custom attributes to fetch" on page 86
- "Custom attributes to return" on page 86
- "Cloud Bridge attributes" on page 86
- "Used attributes" on page 87

Fast sync enabled

This option allows you to disable the automatic initialization of fast synchronization of repository that might impact regular functioning of other dependent components. By default, this option is set to **ON** indicating the fast synchronization happens at intervals that is set in **Time between fast syncs**.

Time between fast syncs

This option allows you to set the required synchronization interval between the fast synchronization of repositories. By default, the interval is set to 5 minutes.

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.

NOTE: Points to consider about the otherMailbox attribute:

- Advanced Authentication 6.4 Service Pack 2 Patch 1 includes the `otherMailbox` attribute as default attribute in User lookup attributes.
- When you upgrade to Advanced Authentication 6.4 Service Pack 3 then the `otherMailbox` attribute is supported as default attribute in the existing repository. However, for a new repository, the `otherMailbox` attribute is not supported as default to improve search and performance.

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.

NOTE: Points to consider about the otherMailbox attribute:

- Advanced Authentication 6.4 Service Pack 2 Patch 1 includes the `otherMailbox` attribute as default attribute in User Name attributes.
- When you upgrade to Advanced Authentication 6.4 Service Pack 3 then the `otherMailbox` attribute is supported as default attribute in the existing repository. However, for a new repository, the `otherMailbox` attribute is not supported as default to improve search and performance.

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
`PKI card certificate id attribute`	`altSecurityIdentities`	`X509IssuerSerialNumber`
`User Object Class`	`user`	OpenDJ and OpenLDAP: `person`
`Group Object Class`	`group`	OpenDJ: `groupOfNames` OpenLDAP: `posixGroup`
`Group Member Attribute`	`member`	`OpenDJ: member` OpenLDAP: `memberUid`. If a required group contains `groupOfNames` class, disable **POSIX style groups**. If the group contains `posixGroup`, enable **POSIX style groups.** ◆ `User UID attribute` This attribute is available only when **POSIX style groups** is **ON**.Default value: `uid`.

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

Custom attributes to fetch

This attribute is applicable for the RADIUS Server event. This attribute displays additional information (for example, pager number) on the RADIUS client.

Custom attributes to return

This list show attributes which should be returned to the REST API clients on successful authentication

Cloud Bridge attributes

The following table describes the batch attributes that the appliance uses:

Attribute	Description
CB page size limit	The maximum number of users per groups to fetch from LDAP server. The value should not exceed 1000.
CB chunk request timeout	The number of seconds of idle time before a batched collection session is terminated. The default value is 600 seconds.
CB ldap read timeout	The number of minutes the LDAP client will wait for the server response before a connection is terminated (0=no timeout). The default value is 5 minutes.
CB users_page size limit	The maximum number of users to be processed in one batch. The value should not exceed 1000.
CB groups page size limit	The maximum number of groups to be processed in one batch. The value should not exceed 1000.
User repository alias	Alias name for the repository for ease of identification among numerous repositories.

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 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	?	×

Attribute Name	LDAP Name	Description	Type	Supported in Active Directory	Supported in eDirectory
GUID	`GUID`	An assured unique value for any object	Octet String (https://www.novell.com/documentation/developer/ndslib/schm_enu/data/sdk5652.html)	×	?
Object Class	`Object Class`	An unordered list of object classes	String	?	?
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 (https://docs.microsoft.com/en-us/windows/win32/adschema/s-enumeration?redirectedfrom=MSDN)	?	×
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 (https://docs.microsoft.com/en-us/windows/win32/adschema/s-enumeration?redirectedfrom=MSDN)	?	×
Primary-Group-ID	`primaryGroupID`	A relative identifier (RID) for the primary group of a user	Enumeration (https://docs.microsoft.com/en-us/windows/win32/adschema/s-enumeration?redirectedfrom=MSDN)	?	×

Attribute Name	LDAP Name	Description	Type	Supported in Active Directory	Supported in eDirectory
Object-Guid	`objectGUID`	A unique identifier for an object	Octet String (https://docs.microsoft.com/en-us/windows/win32/adschema/s-string-octet?redirectedfrom=MSDN)	?	×
object-Sid	`objectSid`	A Binary value that specifies the security identifier (SID) of the user	Octet String (https://docs.microsoft.com/en-us/windows/win32/adschema/s-string-octet?redirectedfrom=MSDN)	?	×
Logon-Hours	`logonHours`	Hours that the user is allowed to logon to the domain	Octet String (https://docs.microsoft.com/en-us/windows/win32/adschema/s-string-octet?redirectedfrom=MSDN)	?	×
USN-Changed	`uSNChanged`	An update sequence number (USN) assigned by the local directory for the latest change including creation	Interval (https://docs.microsoft.com/en-us/windows/win32/adschema/s-interval?redirectedfrom=MSDN)	?	×

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

6.3.2 Health check settings

Advanced Authentication server performs a background task to determine its connection with Repositories through the Cloud Bridge Agent. The connection status is indicated in the **Current Configuration** section of **Edit Cloud Bridge External Repository** page. You must define the following settings based on which the overall status is measured:

- **Success Threshold (times)**: Specify the number of consecutive connections that must succeed to recognize the connection as healthy. By default, success threshold is 1.
- **Failure Threshold (times)**: Specify the number of consecutive connections that must fail to recognize the connection as unhealthy. By default, failure threshold is 3.

For example, the below table details the configured threshold value and equivalent health status:

Option with value	Health Status
Success Threshold is set to 5	Five success messages indicates the connection is Healthy.
Failure Threshold is set to 7	Seven error messages indicates the connection is Unhealthy.

6.3.3 Synchronizing Cloud Bridge Repository

To synchronize the Cloud Bridge repository, perform the following steps:

1. Click **Repositories** and click **Edit** to edit the Cloud Bridge repository.
2. Click **Full Synchronization** to perform a full synchronization.

   Check the synchronization status after some time.

NOTE: The `YOURREPOSITORYNAME (fast): Users processed=0... Groups processes=0` message is acceptable.

NOTE: Fast synchronization occurs every five minutes. Full synchronization occurs once in seven days. However, every three hours a check is performed to confirm whether the full synchronization happened or not.

6.3.4 Testing Cloud Bridge

Advanced Authentication enables you to test Cloud Bridge configuration while creating, updating, or troubleshooting Cloud Bridge. When you test Cloud Bridge, you either ensure that the Cloud Bridge is properly configured, or you can change the Cloud Bridge configuration and quickly test again to check the results.

Perform the following steps to test Cloud Bridge:

1. Click **Repositories** and click **Edit** to edit the Cloud Bridge repository.
2. Click **Test Configuration** to test the configuration.

   A message `Your configuration looks OK` is displayed.

6.3.5 Force configuring Cloud Bridge

Advanced Authentication enables you to change the settings in Repository and apply the changes by saving the settings. If you do not want to wait until the background task to apply the changes in Cloud Bridge configuration, you can break the Cloud Bridge operations and enforces the new configuration by using the **Force Configuration** button.

Perform the following steps to force configure Cloud Bridge:

1. Click **Repositories** and click **Edit** to edit the Cloud Bridge repository.
2. Change the configuration.
3. Click **Force Configuration**.

   A message `Your configuration has been forced` is displayed.

6.3.6 Enabling fast synchronization for eDirectory Repository

To enable fast synchronization for the eDirectory repository that is configured as the Cloud Bridge external repository, ensure to install the change-log module on the eDirectory server. The change-log manages to log all the LDAP changes and enables fast sync for the eDirectory server.

Prerequisites

- The change-log module 4.0.8.1 is supported on eDirectory 9.2.
- To install the change-log module, you must have full rights to the root of eDirectory container.
- For synchronizing changes, ensure that you have the following rights to the base container of eDirectory:
  - **Entry Rights**: Read permission to collect attributes
  - **Attributes Rights**: Read permission on the attribute that are collected
  - **ACL**: Read

NOTE: If you have Identity Manager (IDM), Advanced Authentication, and a change-log module dedicated for IDM, you cannot point Advanced Authentication to the existing change-log module to achieve fast synchronization for the eDirectory repository. However, it is recommended to install a change-log module specifically for Advanced Authentication.

First, obtain the change-log module installer from here (https://download.microfocus.com/protected/Summary.jsp?buildid=vjmCGDYjbdA~). The required files are available in the `IDM_Changelog_4081.zip`

Perform the following steps to extend the schema and install the change-log module:

1. Create a remote eDirectory schema file (`clschema.sch`) with the following content:

```
NDSSchemaExtensions DEFINITIONS ::=
BEGIN

"DirXML-ServerKeys" ATTRIBUTE ::=
{
        Operation               ADD,
        Flags                   {DS_READ_ONLY_ATTR, DS_HIDDEN_ATTR},
        SyntaxID                SYN_OCTET_STRING,
        ASN1ObjID               {2 16 840 1 113719 1 14 4 1 65}
}

END
```

2 Extend the connected remote eDirectory schema to introduce a new attribute `DirXMLServerKeys`. You must perform an eDirectory heath check to ensure that the tree is ready to accept the new schema.

   To extend the `clschema.sch` schema file, use the ice utility (https://www.netiq.com/documentation/edirectory-92/edir_admin/data/a5hgmnu.html).

   For example:

   ```
   ice -S SCH -f clschema.sch -D LDAP -s <remote eDirectory server> -d
   <Admin DN> -w <password>
   ```

3 Stop eDirectory.

4 Navigate to the directory containing the change-log RPM and perform one of the following actions:

   - To install the change-log RPM, run the following command:

     ```
     rpm -ivh <rpm name>.rpm
     ```

     Example: `rpm -ivh ./novell-DXMLChlgx.rpm`

   - To upgrade the change-log RPM, run the following command:

     ```
     rpm -Uvh --noscripts ./novell-DXMLChlgx.rpm
     ```

5 Start eDirectory.

6.4 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 an additional privilege to share the authenticators to the Helpdesk administrators as **SHAREAUTH ADMINS**.

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

NOTE: By default the helpdesk administrator cannot share the authenticators. Only when the helpdesk administrator is added in **Members** in the **SHAREAUTH ADMINS**, the helpdesk administrator is allowed share the authenticators. However, the **Enable sharing of authenticators** in "Authenticator management options" policy must be enabled to share authenticators.

NOTE: The Reporting Portal is accessible only to the FULL ADMIN role.

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 perform the following: you can edit the name of the Local repository.

- Edit the name of the Local repository in **Name**.
- Specify alias name for the repository in **User repository alias** for ease of identification among numerous repositories.

NOTE: **User repository alias** is available only on Advanced Authentication as a Service model.

6.5 Adding a SCIM managed Repository

IMPORTANT: The SCIM managed repository is applicable only for Advanced Authentication as a Service (SaaS).

You can add a SCIM (System for Cross-domain Identity Management) managed repository. The main objective of adding a SCIM managed-repository is to employ an API that simplifies the user identity management for the cloud deployments. With SCIM managed-repository, Advanced Authentication as a Service can accept SCIM push from external identity providers such as Azure directory, Google, and so on. A token is issued when you create the SCIM managed repository, using which administrators can manage the users.

The SCIM API has read only access to any external repository, therefore it is possible to use the SCIM API calls to validate the accuracy of a Cloud Bridge managed repository.

For more information about the SCIM API calls, see SCIM (https://www.netiq.com/documentation/advanced-authentication-64/apidoc/data/apidoc.html#scim).

To add a SCIM managed-repository, perform the following steps:

1 Click **Repositories > New SCIM managed repo**.

2 Specify the name of the repository in **Name**.

3 Specify alias name for the repository in **User repository alias** for ease of identification among numerous repositories.

4 Click Copy token to Clipboard icon  to copy the token for further use.

5 Click **OK**.

7 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**.

You can configure the following methods in Advanced Authentication:

- Apple Touch ID
- BankID
- Bluetooth
- Bluetooth eSec
- Card
- Denmark National ID
- Device authentication
- Email OTP
- Emergency Password
- Facial Recognition
- FIDO2
- Fingerprint
- Flex OTP
- HANIS Face
- HANIS Fingerprint
- LDAP Password
- OATH OTP
- Out-of-band
- Password
- PKI
- RADIUS client
- SAML service provider
- Security questions
- Smartphone

- SMS OTP
- Swisscom Mobile ID
- FIDO U2F
- Voice
- Voice OTP
- Web authentication method
- Windows Hello

7.1 Customizing methods name

You can translate the method name to a preferred language in the **Custom names** section. The translated method name will appear in the following portals, clients, and events:

- **Portals**: Administration, Helpdesk, Self-Service, and Reporting
- **Clients**: Windows, Linux PAM, and Mac OS X
- **Events**: OSP, RADIUS, and custom events

To customize and translate the method name to a specific language, perform the following steps:

1. Open the method for which you want to localize the method name.
2. Specify the method name in a specific language field in the **Custom names** section.
3. Click **Save**.

7.2 Configuring tenancy settings

IMPORTANT: The Tenancy Settings are not available in Advanced Authentication as a Service (SaaS) version

---

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.

7.3 Capabilities of authentication methods

Authentication method is an approach validate the identity of users and prevent unauthorized users from accessing sensitive information.

The following table lists various authentication methods of Advanced Authentication with their capabilities:

Authentication Method	Strengths
Password	Passwords are widely used and familiar to users. They can be complex and unique, it is secure when combined with other device-based authentication factors.
OTP-based authentication: ◆ Email OTP ◆ SMS OTP ◆ Voice OTP	◆ OTP is valid for a short duration and single-use. ◆ OTP is generated based on algorithms and it is not a static value. Therefore, there is no need to remember the PIN or password.
Time-based OTP (TOTP)	◆ The trusted device generates the OTP and the server validates the token. ◆ Devices that generate and accept TOTP codes can be used offline without an internet connection. ◆ Hardware tokens generate unique OTPs and are not tied to a specific device. They are resistant to attacks targeting software-based OTP generators and provide an additional physical layer of security.
Hash-based OTP (HOTP)	◆ The counter is synchronized between the server and the client. ◆ Soft token
Facial Recognition	◆ Verifies identity using biometric ◆ Most secure ◆ Difficult to replicate ◆ Convenient to use
Fingerprint	◆ Verifies identity using biometric ◆ Most secure ◆ Difficult to replicate ◆ Convenient to use
PKI	◆ Digital certificate-based authentication. ◆ Protects confidential data and provides unique identities to the users and system.
Smartphone (Push Notification)	It involves sending a verification request to the registered device. Users can confirm or deny the authentication attempt, providing an additional layer of involvement and control.
Windows Hello	◆ Passwordless, however, it is biometric-based (Fingerprint & Face).Convenient to use.Provides anti-spoofing feature. ◆ Supports TPM-based authentication.

7.4 Apple Touch ID

Apple Touch ID is an electronic fingerprint recognition feature, available in Mac operating system devices, that allows the users to authenticate to Mac OS workspace. Users can authenticate with methods such as something you know (LDAP Password, Password) or something you have (Card, Smartphone) and **Apple Touch ID** (something you are) for multi-factor authentication. Users need to place their finger on the Touch ID scanner to enroll and authenticate.

To configure this method, add Apple Touch ID method to an authentication chain.

NOTE: You must install the Device Service on the Mac workstation to use this method.

NOTE: You cannot use Touch ID for the initial authentication after boot.

Enrolling Apple Touch ID and authenticating to Mac OS with Apple Touch ID

Consider an administrator performed the following steps to enforce users to enroll the Apple Touch ID method in Mac OS device.

1. Created a chain with the **Apple Touch ID** method and added another method such as **LDAP password**.
2. Assigned the chain to the **Mac OS Logon** event.

Paul, an end user, logs in to the Self Service portal and enrolls the Apple Touch ID method using his fingerprint. After enrollment, Paul authenticates to his Mac OS workstation by specifying the LDAP password and placing fingers on the Touch ID scanner.

7.5 BankID

Advanced Authentication provides the BankID method that facilitates users to authenticate with their personal identification number. Advanced Authentication supports both the desktop and the mobile versions of BankID. In this method, the user must configure the BankID app with the personal identification number, activation, and security code. The security code is mapped with the personal identification number.

NOTE: The user must ensure to set the security code with six digits in non-sequential format (for example: 221144) in the BankID app.

While enrolling the user, the specified identification number is saved as a template in the Advanced Authentication database. This method allows the users to get authenticated by specifying their secret code configured on the BankID app.

When a user wants to authenticate on an endpoint such as a laptop or a website with the BankID method. In this scenario, the authentication flow is as follows:

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 request to the BankID app.

4 User opens the BankID app, specifies the **Security Code**.

- Click **Identify** on the Mobile app.
- Click **Verify my identity** on the Desktop app.

5 The Security code is sent to the BankID server to validate.

6 The BankID server validates the authentication and the endpoint gets authenticated.

To configure the BankID method, perform the following steps:

NOTE: Ensure that you have the BankID client SSL certificate as a pre-requisite.

---

1 Click **Browse** then select the client SSL certificate from the local drive.

The certificate must be in `PKCS12` format.

2 Specify **Private key password**.

3 Set **Enable Test Mode** to **ON**, to allow the user to test the authenticator with valid test BankID.

If you set this option to **OFF**, users must use valid production BankID to enroll the authenticator.

4 Click **Save**.

7.6 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.

NOTE: The Bluetooth method is not available from Advanced Authentication 6.4 Service Pack 1. It is recommended to remove or replace the Bluetooth method from existing chains. If not, Advanced Authentication 6.4 Service Pack 1 blocks login from chains that include the Bluetooth Method. Also, users cannot enroll and log in with the Bluetooth method.

By default, the **Enable reaction on device removal** option is enabled. When this option is enabled and a user logs in to Windows using Bluetooth, Windows gets locked automatically or performs an action defined in Interactive logon: Smart card removal behavior policy (https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-R2-and-2012/jj852235(v=ws.11)?redirectedfrom=MSDN) in the following scenarios:

- 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 enhance the security.

7.7 Bluetooth eSec

Bluetooth eSec method identifies and connects with the mobile device when within the range to authenticate a user. It provides effortless user authentication.

NOTE: The Bluetooth eSec method is supported only on Windows Client and all Linux Platforms.

You can configure the Bluetooth eSec method with the following options:

- **Enable reaction on device removal** option is enabled by default. When this option is enabled and a user logs in to Windows Client using Bluetooth eSec, Windows gets locked automatically or performs an action defined in Interactive logon: Smart card removal behavior policy (https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-R2-and-2012/jj852235(v=ws.11)?redirectedfrom=MSDN) in the following scenarios:
  - Bluetooth is not turned ON on the device
  - The device is not in the range

  NOTE: The option is only applicable for Windows Client.
- **Required paired device (more secure)** option is enabled by default. When this option is enabled users are restricted to authenticate to Client with paired devices only. When this option is disabled users can authenticate to Client machines without pairing a mobile device.

7.8 Card

The **Card** authentication happens in the following cases:

- When a contactless card is placed on a card reader.
- When a Near Field Communication (NFC) tag is placed near a smartphone which supports NFC.

  IMPORTANT: The authentication using the NFC tag works only on the NFC supported Android smartphones and tablets.

  NOTE: Advanced Authentication supports NFC tag for authenticating to OAuth 2.0/ OpenID Connect, SAML 2.0 events, and Advanced Authentication portals. The user must have the Android smartphone and tablet that supports NFC and the Google Chrome browser to enroll and authenticate using this method.
- When a smart card with an integrated token supporting PKCS#11 library is inserted into the card reader.

  The PKCS#11 library provides a standardized interface for obtaining basic token information and is not used for encryption.

  NOTE: The authentication using the card with an integrated token supporting PKCS#11 libraries is supported only on Windows Client.

To use this type of card as a Card method, the smart card must be equipped with an integrated token compatible with PKCS#11 libraries. Additionally, ensure that your card reader adheres to the PKCS#11 standards.

Furthermore, to use this reader, you must configure the below parameter in the device service:

- `card.pkcs11Enabled`
- `pki.vendorModule` and associated PKI settings

For more information see, Configuring the card settings and Configuring smart card with token supporting PKCS#11 library in the Advanced Authentication - Device Service guide.

For more information about the supported cards and card readers, see Supported card readers and cards in the Advanced Authentication - Device Service guide.

NOTE: It is recommended to combine the **Card** method with another stronger authentication method in a chain to enhance the security. However, it is not advisable to combine the **Card** method with the **PKI** method in a chain because the **PKI** method already contains card serial number tracking.

To configure the Card method with the NFC tag as second-factor authenticator to secure OAuth2 / OpenID Connect based smartphone application, see the below video:

 http://www.youtube.com/watch?v=L85CAipxfns

Advanced Authentication supports the Microsoft policy Interactive logon: Smart card removal behavior (https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-R2-and-2012/jj852235(v=ws.11)?redirectedfrom=MSDN) 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 (https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-R2-and-2012/jj852235(v=ws.11)?redirectedfrom=MSDN). 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 (https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-R2-and-2012/jj852235(v=ws.11)?redirectedfrom=MSDN)) 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.

When you enable Single-sign on (SSO) for Remote Desktop, the Interactive logon: Smart card removal behavior policy (https://docs.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/interactive-logon-smart-card-removal-behavior) is ignored. You need to disable SSO to make it work.

7.9 Denmark National ID

Denmark National ID is an electronic personal identification system used in Denmark to communicate with public sectors, online banking, and online purchases. Denmark National ID authentication consists of a Danish Personal Identification number, a password, and a pin from the provided code card.

Advanced Authentication facilitates citizens of Denmark to authenticate using the CPR (Danish Social Security number), a password, and the pin that has been enrolled with the Social Security number. The Denmark National ID method is implemented to authenticate to the Advanced Authentication portals such as Self-Service (Enrollment), Helpdesk, Reporting, and OAuth 2.0/ OpenID Connect/ SAML 2.0 events.

To configure the Denmark National ID method, specify the following details and save it:

Parameter	Description
Service Provider ID	Specify the third-party service provider identification number that verifies the Social Security Number.
Service Provider VOCES certificate	Click **Browse**, then select the VOCES certificate from the local drive. The certificate has been issued as part of the Denmark National ID enrollment process. **NOTE:** The certificate file must be in `PKCS12` format.
Use test environment	Keep this option **OFF**.
User Social Security number attribute	The user's Social Security number against which the validation takes place. You can use the custom attribute workforce ID of the repository.
Allow overriding Social Security Number	The option is to prevent users from providing a Social Security number that is not registered in the LDAP repository. The option is set to **ON** by default. Set to **OFF** to prevent users from specifying the Social Security number during enrollment.

7.10 Device authentication

In the **Device Authentication** method, a device stores the private key and secures it with a PIN. It uses the trusted module or the file system of the device to store unique details of a user, such as private key and PIN.

Advanced Authentication supports the following two forms of Device Authentication method:

- Windows Trusted Platform Module (TPM)
- Without using the Trusted Platform Module (Non-TPM)

NOTE: Ensure users enroll the Device Authentication method using the workstation where they would perform further authentication. Enrollment on one machine and authentication on another machine is not supported.

7.10.1 Windows Trusted Platform Module (TPM)

The TPM chip is a crypto-processor available in Windows workstation to achieve actions, such as generating, storing, and limiting the use of cryptographic keys. Device Authentication supports authentication to Windows workstation and makes use of information available in the chip to authenticate users.

NOTE: Advanced Authentication cannot manage the TPM management. It is possible to manage the TPM virtual smart card and unlocking the same with the `tpmsvcmgr` command. For more information, see Tpmvscmgr (https://docs.microsoft.com/en-us/windows/security/identity-protection/virtual-smart-cards/virtual-smart-card-tpmvscmgr)

NOTE: The Virtual Smartcard Module that is part of the operating system manages the lock status of the virtual smart card. With the below pre-conditions if the virtual smart card in the Advanced Authentication Windows Client gets locked after six failed attempts, you can use the `tpmvscmgr` command to destroy the instance to remove the virtual smart card from the system:

- The Lockout options policy is not configured in the Advanced Authentication Server.
- The Standard User Individual Lockout Threshold (https://docs.microsoft.com/en-us/windows/security/information-protection/tpm/trusted-platform-module-services-group-policy-settings#standard-user-individual-lockout-threshold) policy is not configured in Windows TPM.

Syntax: `tpmvscmgr.exe destroy /instance <instance ID>`

Example: `tpmvscmgr.exe destroy /instance ROOT\SMARTCARDREADER\0004`

Destroying the instance does not delete the enrolled Device Authentication method. However, users are required to re-enroll the Device Authentication method.

Prerequisite

Before you configure the Device Authentication method, ensure that user's system is Windows 10 machine with fully functional TPM as a prerequisite.

Preconfiguration tasks

To set up a Windows workstation for using the TPM virtual smart card, refer to the Microsoft Walkthrough (https://docs.microsoft.com/en-us/windows/security/identity-protection/virtual-smart-cards/virtual-smart-card-get-started#step-2-create-the-tpm-virtual-smart-card) guide and perform the following tasks:

- Create the certificate template
- Create the TPM virtual smart card
- Enroll the certificate on the TPM virtual smart card

NOTE: In the pre-configuration tasks, creation of certificate template and enrollment the certificate are not required when you allow users to enroll and authenticate with the Device Authentication method through the key pair generation.

Adding the trusted root certificates

You must upload the trusted root certificates for the Device Authentication method. Ensure that the Root CA certificate is in the `.pem` format. However, the trusted root certificates are not required when you allow users to enroll and authenticate with the Device Authentication method through the key pair generation.

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

1. Click the Add icon + in the **Device Authentication** page.
2. Click **Choose File** and select the `.pem` certificate file.
3. Click **Upload**.
4. Click **Save**.

Disabling the key-pair option

The **Allow key-pair** option is enabled by default. This indicates that users can enroll the Device Authentication method either with the CA certificates or through the key-pair generation. However, you can set **Allow key-pair** to **OFF** to disable the key-pair based enrollment and enforce enrollment only using a user certificate issued by the CA.

7.10.2 Without using the Trusted Platform Module (Non-TPM)

This mode is supported on Linux, macOS, and Windows operating systems. In this mode, a key pair generates during enrollment and is stored in the file system of workstation rather than the TPM chip. The key pair is secured using the PIN.

To disable the TPM chip in Windows workstation, see Device authentication setting.

7.11 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.

Parameter	Description
OTP format	Length of an OTP token. The default value is 6 digits.
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: ◆ `{user}`: Username. ◆ `{endpoint}`: Device that a user authenticates to. ◆ `{event}`: Name of the event where the user is trying to authenticate to. ◆ `{number}`: Sequence of the OTP, user is required to specify to authenticate. ◆ `{otp}`: One-Time-Password to be sent to the user.
Allow re-sending after (seconds)	The duration from previous OTP to re-send a fresh OTP for authentication.
Allow overriding 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.
Verify email address	This option sends the verification code to a specified email address and allows users to validate the email address during the manual enrollment. The option is set to **OFF** by default. Set this option to **ON** to permit users to check whether the enrolled email address is valid.
Allow user enrollment without e-mail	Option to configure settings for the user to enroll the Email OTP authenticator without an email in the repository. Set this option to **OFF** to ensure that a user does not enroll the Email OTP authenticator without an email. The user gets an error message that you can specify in **Error message**. Set this option to **ON** to allow the user to enroll the Email OTP authenticator without an email.
Allow as first authentication method	Option that allows a user to authenticate using a chain where Email OTP authenticator is the first authentication method. The option is set to **ON** by default. Set this option to **OFF** to prevent user from authenticating using a chain where Email OTP authenticator is the first authentication method. If the option is set to **OFF**, and a user tries to authenticate using a chain where the Email OTP method is the first authentication method, the user is displayed a `The method cannot be first in the login chain` message and the user cannot authenticate.

NOTE: After you configure the Email OTP method, it is required to configure the Mail sender policy to deliver the Email OTP to users.

7.11.1 Customizing Email settings for an event

You can customize the email settings for a specific event in the **Event Customization** tab. An email with OTP is delivered to users based on the settings configured for each event.

To customize the Email Settings to a specific event, perform the following steps:

1. Navigate to **Methods > Email OTP > Event Customization** in the Administration portal.
2. Click **Add Custom Event** icon +.
3. Select a preferred event from the list.
4. Modify the email settings for the event as per the requirement.
5. (Conditional) If you want to customize the method name for the event, expand **Custom names** and specify the method name in required language field.
6. Click **Save**.

For example, let us assume an organization's requirement is to customize the email settings for Windows logon event as follows:

Parameter	Value
OTP period	180 seconds
OTP format	8 digits
Subject of email	OTP for authentication
Body	Hi {username}Your one-time-password to authenticate to the Windows workstation is {otp}.Thanks, Support team

Following are the steps to customize the email settings for Windows logon event according to the preceding requirement:

1. Click **Methods > Email OTP > Event Customization** in the Administration portal.
2. Click **Add Custom Event** icon.
3. Select **Windows logon** event from the list.
4. Modify the settings as per the above table.
5. Click **Save**.

With the above configuration, when an end-user tries to log in to Windows workstation with the Email OTP method, 8 digits OTP is sent to registered email address and the OTP is valid for 3 minutes (180 seconds).

7.12 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 minimum length of the password. The default value is 10.
Password age (minutes)	The validity period of a password. The default value is 4320 minutes. The value must not exceed 7200 minutes.
Maximum logins	The maximum number of login attempts that a user can perform before the password gets expired. The default value is 10. The value must not exceed 100.
Complexity requirements	By default this option is set to **ON** to enforce users creating a complex password. When enabled, the password must meet three of 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
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.

7.13 Facial Recognition

Advanced Authentication provides advanced biometric authentication with the Facial Recognition method. This method allows users to get automatically authenticated by presenting their face. The image of the face is captured by an integrated or external camera and recorded by the configured API server, when the user enrolls the method. When the user tries to authenticate on an application, the API engine identifies and validates the recorded image with the actual image. If the images match, the user is authenticated.

IMPORTANT: It is recommended to configure the blink detection (https://www.netiq.com/documentation/advanced-authentication-63/device-service-installation/data/facialrecognition.html) or combine the Facial recognition method with another method in a chain to enhance security.

WARNING: You must have the Advanced Authentication Device Service installed to use the Facial recognition method for logging in to the following:

- Operating System: Windows, Linux, and Mac workstations.
- Integration: OAuth 2.0 and SAML 2.0.

Advanced Authentication supports the following Face Recognition API services:

- Azure cognitive service
- Contactable KYC service

NOTE: You can configure one of the API service to apply Facial Recognition method for authentication.

7.13.1 Azure cognitive service

The Azure Cognitive service provides algorithms that detect, identify, and validate human faces for identity verification. This service also captures high quality images. Also, extracts a set of face-related attributes, such as head pose, emotion, facial hair, and liveliness.

Before you configure the Facial Recognition method, you must generate the **Access Key** and **Endpoint URL** from the Microsoft Cognitive Services (https://azure.microsoft.com/en-in/products/cognitive-services/face/).

To configure the Facial Recognition method, perform the following steps:

1. Click **Methods > Facial Recognition** on the Advanced Authentication Administration Portal.
2. Check **Azure Cognitive Service** is set as **API Provider**.
3. Specify **Endpoint URL**. This URL is location based.

   NOTE: The Endpoint URL must contain `face/v1.0` at the end.

   For example: `https://westcentralus.api.cognitive.microsoft.com/face/v1.0`.
4. Specify **Access Key** that you have generated in the Microsoft Cognitive Services. This key is used while authenticating the user.

NOTE: Verify the following points before implementing Facial Recognition method with Azure Cognitive service:

- For a better quality of recognition, you must use cameras with a high definition of 720p and above.

- During enrollment, the captured images are placed on Microsoft servers and Microsoft Cognitive Services return only the Face ID to Advanced Authentication. The Advanced Authentication stores this Face ID as enrolled authenticator. Therefore, when you change the Access Key, the related enrollments are lost.
- This method does not support caching on Windows Client, Mac OS X Client, and Linux PAM Client.

7.13.2 Contactable KYC service

Contactable provides a highly effective identity orchestration platform that includes advanced Intelligent Facial Biometric Anti-spoofing algorithms. These algorithms are made available in a frictionless user verification journey using simply 2 dimensional facial images captured directly from a mobile or web-based camera.

The anti-spoofing algorithms are NIST rated and i-Beta level 1 and level 2 ISO 30107-3 compliant qualifying them as quite reliable. The anti-spoofing algorithms can test for presentation attacks related to printed photos, cut-out masks, digital and video replay attacks and 3-dimensional masking. While testing, the service also checks for liveness and prevent fraud being committed through the use of false facial identity by comparing a living person's facial image to a reference image often obtained independently from an independent data source. A successful biometric comparison of a user's facial image to a reference image then binds the two data sets to each other for use in assessing truth.

To configure the Contactable KYC service for Facial Recognition, perform the following steps:

1. Click **Methods > Facial Recognition** on the Advanced Authentication Administration Portal.
2. Select **Contactable KYC Service** as **API Provider**.
3. Specify the following details:

Parameter	Description
Base URL	The Contactable KYC service URL for validating the captured face.
User name	The username to access the Contactable KYC service.
Password	The password to access the Contactable KYC service.
Organization code	An unique code thats helps Contactable KYC service to group the requests.
Encryption Key	The key to secure the communication between the Contactable KYC service and Advanced Authentication.
Encryption initialization vector	A value that is used along with a secret key to encrypt data so that the encrypted values are not identical.
Client timeout (seconds)	The duration till when the Advanced Authentication server waits for a response from the Contactable KYC service.
Allow lower resolution image scan	It enables the Advanced Authentication server to receive the lower resolution facial images that do not comply with standards. The option is set to **OFF** by default. The facial image that does not comply with the standard is not sent to the server for validation. However, if the face recognition device complies with image standards then the authentication is successful without any issue. When set to **ON**, the Advanced Authentication server receives the lower resolution facial images that do not comply with standards. However, authentication might not be successful.

4 Click **Save**.

7.14 FIDO2

The FIDO2 method facilitates users to use the devices that comply with FIDO standards for authenticating to any web-based environment. The devices can be built-into the platform or external devices connected through USB. The FIDO2 method uses the Web Authentication (WebAuthn) API, and Client to Authenticator Protocol (CTAP). The WebAuthn enables strong authentication with public key cryptography and allows password-less authentication.

NOTE: On the Safari browser, while authenticating to a web application with the FIDO2 method, a user must click **Next** to initiate the authentication. This applies irrespective of the order of the FIDO2 method in a chain.

NOTE: Advanced Authentication FIDO2 method supports authentication to the following:

- Portals: Administration, Helpdesk, Self-Service, and Reporting

- Events: OAuth 2.0, SAML 2.0, and Windows logon including the workstation lock or log off cases in compliance with Interactive logon: Smart card removal behavior (https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/jj852235(v=ws.11)) policy.

  The Crescendo C2300 smartcard is supported for Windows logon.

FIDO2 method supports the following browsers with specific device:

- Firefox and Google Chrome browsers with the U2F device
- Microsoft Edge browser with Windows Hello authentication
- Google Chrome browser:
  - With Touch ID authentication on macOS
  - Using Crescendo C2300 smartcard on Windows

While you use Google Chrome browser, it is required to set a valid domain name for your Advanced Authentication server rather than an IP address.

If users have enrolled the FIDO2 method using the Windows Hello in Microsoft Edge 17 or earlier supported browser versions then they must authenticate using the same browser. After upgrading to the latest version of Edge that supports the FIDO 2.0 standards, users must re-enroll the FIDO2 method.

To authenticate with the FIDO2 method using the Crescendo C2300 card as second-factor authenticator to Windows workstation, see

 http://www.youtube.com/watch?v=X7j9xph3_g0

For more information about the WebAuthn and FIDO2 authenticators, see these articles: Web Authentication (https://w3c.github.io/webauthn/), Web API for FIDO 2.0 (https://www.w3.org/Submission/2015/SUBM-fido-web-api-20151120/), and Microsoft Web authentication (https://docs.microsoft.com/en-us/archive/microsoft-edge/legacy/developer/dev-guide/windows-integration/web-authentication).

You can configure following options for the FIDO2 method:

- **Resident Key Requirement**: Resident keys are discoverable credentials like private key stored on the authenticator rather than the website (relying party). When the relying party (RP) sends a request to create or retrieve a credential, the authenticator searches for credential with the provided domain name of the RP. Authenticator discovers the credentials that are associated with the RP. To achieve the Username-less login experience Resident key is required.

  Select the required option that indicates the Resident key requirement on the RP during enrollment and authentication. The available options are:

  - **Preferred (Default)**: Represents the relying party favors to create the resident key if the browser supports it. The enrollment and authentication with FIDO2 succeed irrespective the availability of the resident key.

    NOTE: Google Chrome creates and stores the resident key whereas Firefox does not support creation of resident key.

- **Required**: Represents the relying party must create the resident key and display an error message if creation of the resident key is not possible. The enrollment and authentication with the FIDO2 method happen only on the resident key supported browsers.

  For example, if the **Resident Key Requirement** is set to **Required** then user cannot enroll the FIDO2 method on the Firefox as the browser does not have that capability. However, one can use Chrome to enroll the FIDO2 method.

- **Discouraged**: Represents the resident key is not mandatory to complete enrollment and authentication with the FIDO2 method. The relying party does not require the resident key.

- **User Verification (pin)**: Select the required option to allow the authenticator (FIDO2 devices) to verify the authorized user and send the verification response to RP.

  Select the required option that indicates the User Verification that is necessary to perform enrollment, testing, and authentication with the FIDO2 method. The available options are:

  - **Preferred (Default)**: Indicates a prompt to specify the PIN is displayed on the supported browsers like Chrome. The prompt is not displayed on the unsupported browsers like Firefox. The enrollment and authentication with FIDO2 succeed in both cases.
  - **Required**: Indicates PIN is mandatory to complete enrollment, testing and authentication with the FIDO2 method. Therefore, the enrollment and authentication with FIDO2 succeed only on resident key and PIN supported browsers.
  - **Discouraged**: Indicates the prompt to specify PIN is not displayed to users during enrollment, testing and authentication with the FIDO2 method.

NOTE:

Some platform and/or browser combinations do not support User Verification or Resident Key for FIDO2 devices. Therefore, FIDO2 enrollment and authentication might fail if you set **User Verification (pin)** and **Resident Key Requirement** as **Required**.

- **Username-less login enabled**: This option allows users to authenticate to the Web Authentication event and Windows Client using the FIDO2 compliant cards without specifying the username. The option is set to **OFF** by default and user must specify the username to authenticate with the FIDO2 method.

  Set this option to **ON** to allow users to authenticate with FIDO2 card that contains username. The FIDO2 Login button is displayed on the Web Authentication login page. When users tap the card, username gets pre-filled in the Username.

IMPORTANT:

- Before you set the **Username-less login enabled** to **ON**, ensure the following:
  - Set the **Resident Key Requirement** option to **Required**.
  - Fill the domain ID in **Username-less login RP ID**.
- If the **Username-less login enabled** option is set to **ON**, ensure that the **USB Transport**, **NFC Transport**, and **BLE Transport** options are not applicable.
- For Windows Client:
  - Enroll each FIDO2 device to only one user, even though the device supports multiple user enrollment.
  - Enroll the FIDO2 device with the **Required Key Requirement** set to **Required** to enable the username-less login feature on Windows Client.

- Configure the required parameters in the `config.properties` file of the Windows Client. For more information, see Enabling username-less login feature for FIDO2.
- Inform users to present their FIDO2 device directly without clicking anywhere on the Windows Welcome Screen for username-less login

- **Username-less login RP ID**: Unique ID required for username-less login functionality of FIDO2 method.
- **USB Transport**: This option allows users authentication using the USB transport protocol. By default, this option is set to **OFF**.

  Set this option to **ON** to allow users to authenticate with FIDO2 method using the external devices connected through USB ports.
- **NFC Transport**: This option allows authentication using the Near Field Communication (NFC) transport protocol. By default, this option is set to **OFF**.

  Set this option to **ON** to allow users to authenticate with FIDO2 method only through NFC devices.
- **BLE Transport**: This option allows authentication using Bluetooth transport protocol. By default, this option is set to **OFF**.

  Set this option to **ON** to allow users to authenticate with FIDO2 method using the devices connected through Bluetooth.

An example of authenticating with the FIDO2 method

Thomas, an end user, has enrolled the FIDO2 method in the Advanced Authentication Self-Service portal by using the FIDO compliant U2F token. He wants to authenticate to the `mycompany.com` website. When he opens the browser and follows the prompts to access the website. Then, he is required to touch the token when there is a flash. Thomas is validated with the device and gets authenticated to `mycompany.com`.

7.15 Fingerprint

The **Fingerprint** method is one of the strongest biometric authentication methods of Advanced Authentication. 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 50. Reducing the score may result in different fingerprints getting validated.

2 Select the number of fingers that a user must enroll from **Minimum number of fingers to enroll**.

It is recommended to specify a number that is more than 1 because if a finger is injured, the user can use the other enrolled finger.

NOTE: If you want to allow the use of multi-finger reader for enrollment, ensure to select the number of fingers to be enrolled as 4, 6, 8, or 10.

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 Set **Enable multi-finger reader to enroll** to **ON**, to allow users to enroll the Fingerprint method using the Green Bit DactyScan84c multi-finger reader. Users can set **Use multi-finger reader for enrollment** to **ON** and enroll with the multi-finger reader on the Self-Service portal. The Green Bit DactyScan84c device can scan one of the following fingers combination at a time:

- Four fingers of the right hand
- Four fingers of the left hand
- Two thumbs

To enforce the users to scan fingers using the Green Bit DactyScan84c reader, set **Force to use multi-finger reader** to **ON**.

5 Set **Specify fingers during enrollment** to **ON**, if you want to enforce selected fingers for a user to enroll.

6 Select the preferred fingers to enroll from the **Selected fingers** list.

7 Set **Enable Duress finger configuration** to **ON**, to allow users to assign one of the enrolled fingers as duress. In case of emergency or under a threat, user can authenticate with the duress finger. Authentication with the duress finger triggers an alert notification to the configured email address and phone number.

In the **Alert Configuration** section, specify the following details to configure the alert notification that is to be sent to the preferred email address and phone number:

Table 7-1

Parameter	Description
Email Alert Settings	
Email Recipient	The email address of recipient to whom you want to send the email alert.
Email Alert Subject	Subject of the email alert.
Format	Format of email alert. Plain Text is the default format. Other available option is HTML. If you select HTML format, specify the message in HTML.
Email Alert Body	Body of email alert. You can specify the following variables: ◆ {user}: Username. ◆ {endpoint}: Device that a user authenticates to. ◆ {event}: Name of the event where the user is trying to authenticate to.
SMS Alert Settings	
SMS Recipient	Phone number of recipient to whom you want to send the SMS alert.
SMS Alert Body	Text in the SMS that is sent to the recipient. You can specify the following variables: ◆ {user}: Username. ◆ {endpoint}: Device that a user authenticates to. ◆ {event}: Name of the event where the user is trying to authenticate to.

8 Click Save.

NOTE: Ensure that you configure the Mail sender and SMS sender policies with the sender details that are required to send an alert.

Example 1: enrolling multiple fingers and authenticating with one of the enrolled fingers

Consider Thomas, an administrator has performed the following steps to enforce users to enroll the Fingerprint method using the Greenbit DactyScan84c device. Users can authenticate to Linux workstation with the Fingerprint method.

1. Set Force to use multi-finger reader to ON in the Fingerprint method.
2. Created a chain with the Fingerprint method and added another preferred method such as LDAP password or Password.
3. Mapped the chain to the Linux Logon event.

Paul, an end user, logs in to the Self Service portal and clicks on the Fingerprint icon. He selects the four fingers of Right hand and enrolls using the Green Bit DactyScan device. After enrollment, Paul authenticates to his Linux workstation with the Nitgen device using one of the enrolled fingers. He gets authenticated successfully.

Example 2: authenticating with a duress finger during an emergency situation

Consider Thomas, an administrator has performed the following steps to assign an enrolled finger as duress:

1. Set **Enable Duress finger configuration** to **ON** in the Fingerprint method.
2. Configured **Alert Configuration** with the alert notification text, mail address and phone number of a network security officer to send email and SMS.
3. Created a chain with the Fingerprint method along with preferred methods such as **LDAP password** and **Password**. Assigned the chain to **Networks** group.
4. Mapped the chain to the **Linux logon** event. Mail server is hosted on the Linux workstation.

Paul, a network staff, logs in to the Self Service portal and clicks on the Fingerprint icon. He enrolls the middle, index, ring and little fingers of the left hand. Later, he selects **Left index** from **Assign Duress Finger** drop down.

Assume, on an unfortunate day, a miscreant forcibly enters the organization and threatens Paul to authenticate to the Linux workstation. In this situation, Paul can use the duress finger (Left index finger) for authentication which triggers an alert notification to configured security personnel, who will take the necessary action.

7.16 Flex OTP

In the Flex OTP authentication method, users can authenticate using the one-time password (OTP) that they receive from the following enrolled methods:

- HOTP
- TOTP
- Smartphone(Offline)

To configure this method, add Flex OTP method to the authentication chain.

NOTE: Advance Authentication validates the specified OTP in the order: HOTP - TOTP - Smartphone OTP.

For example, if the HOTP and TOTP did not match, the AA Server will count it as an authentication failure for both the methods. If nothing from the methods (HOTP, TOTP, Smartphone OTP) matched, it will be counted as three unsuccessful authentications. So if you use the **Lockout Options** policy, please ensure you don't need to increase the Attempts failed value to avoid sudden lockout. For more information about the lockout settings policy, see “Lockout options” on page 220.

To configure the Flex OTP method as second factor authenticator to secure Windows workstation, see

 http://www.youtube.com/watch?v=euajS7PJ5nE

7.17 HANIS Face

Advanced Authentication provides the HANIS (Home Affairs National Identification System) Face method that facilitates citizens of South Africa to authenticate through their facial recognition that has been enrolled in the National Identification System. However, when the user enrolls this method using their Passport number or National ID. Advanced Authentication forwards the captured details to the third-party Service Provider that is integrated with National Identification System where the validation takes place. Based on the validation result, the user gets authenticated to the required resource or endpoint.

NOTE: The HANIS Face method is supported only in the Advanced Authentication as a Service (SaaS) model. In the on-premises model of Advanced Authentication, this method will be available in the upcoming 6.3 Service Pack 7 release.

To understand how the authentication flows in HANIS Face method, see Authentication flow in the HANIS method.

To configure the HANIS Face method, specify the following details:

Parameter	Description
Base URL	The third-party Service Provide URL that is integrated with National Identification System.
User name	The username to access the third-party Service Provider.
Password	The password to access the third-party Service Provider.
Organization code	An unique code using which the third-party Service Provider requires to group the requests.
Encryption Key	The key to secure the communication between the third-party Service Provider and Advanced Authentication.
Encryption initialization vector	A value that is used along with a secret key to encrypt data so that the encrypted values are not identical.
HANIS API client timeout (seconds)	The duration till when the Advanced Authentication server waits for a response from the third-party Service Provider.
User ID/Passport attribute	The passport number or national ID of a user against which the validation takes place. You can use custom attribute `workforce ID` of the repository.
User cell phone attribute	The cell phone number of a user that the third-party Service Provider requires for processing the authentication request. You must define the attribute in User cell phone attributes of the Repositories section.
Allow overriding ID/Passport number	Option to prevent users from providing a passport number that is not registered in the LDAP repository. The option is set to **ON** by default. Set to **OFF** to prevent users to specifying the passport number during the enrollment.

Parameter	Description
Allow overriding 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.
Allow lower resolution image scan	It enables the Advanced Authentication server to receive the lower resolution facial images that do not comply with standards. The option is set to **OFF** by default. The facial image that does not comply with the standard is not sent to the server for validation. However, if the face recognition device complies with image standards then the authentication is successful without any issue. When set to **ON**, the Advanced Authentication server receives the lower resolution facial images that do not comply with standards. However, authentication might not be successful.
Max liveness detection attempts	The maximum number of times the server tries to detect the liveness of the face during authentication. Liveness includes some actions such as eye movement, blink, head tilt and so on. The default value is 3.

NOTE: When you modify the settings related to the HANIS Face method, ensure to specify the **Password**, **Encryption Key**, and **Encryption initialization vector** to apply the changes.

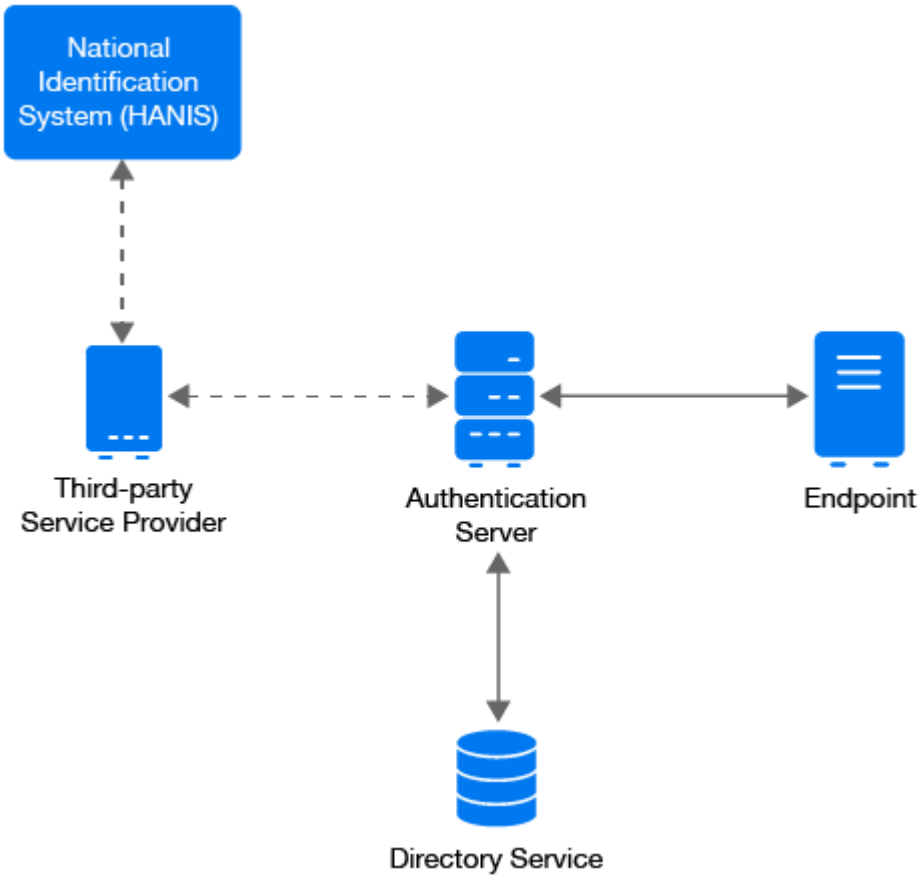
7.18 HANIS Fingerprint

Advanced Authentication provides the HANIS Fingerprint method that facilitates citizens of South Africa to authenticate through their fingerprint that has been enrolled in the National Identification System. However, when the user enrolls this method using their Passport number or National ID. Advanced Authentication forwards these details and captured fingerprint to the third-party Service Provider that is integrated with National Identification System where the validation takes place. Based on the validation result, the user gets authenticated to the required resource or endpoint.

The HANIS Fingerprint method is implemented to authenticate to the Advanced Authentication portals, such as Self-Service (Enrollment) and Helpdesk.

Authentication flow in the HANIS method

The authentication flow for the HANIS 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 HANIS method. The following steps describe the authentication flow:

1. When the authentication request is initiated, the endpoint contacts the Advanced Authentication server.
2. Along with the biometrics data (fingerprint scan or facial image), the Advanced Authentication server retrieves the user's details, such as Passport number or National ID and phone number from the repository if available. However, the endpoint must send these details as part of the authentication request.
3. The Advanced Authentication server forwards the authentication request to the third-party Service Provider.
4. The Service Provider that is integrated with the National Identification System forwards the authentication request to the Identification System.
5. The National Identification System validates the details, such as passport number, phone number, and biometrics data.
6. After the validation, the National Identification System shares validation status with the third-party Service Provider.
7. The third-party Service Provider transmits the validation status to the Advance Authentication server.
8. Finally, the Advanced Authentication server authenticates a user to the endpoint based on the validation status.

To configure the HANIS Fingerprint method, specify the following details:

Parameter	Description
Base URL	The third-party Service Provide URL that is integrated with National Identification System.
User name	The username to access the third-party Service Provider.
Password	The password to access the third-party Service Provider.
Organization code	An unique code using which the third-party Service Provider requires to group the requests.
Encryption Key	The key to secure the communication between the third-party Service Provider and Advanced Authentication.
Encryption initialization vector	A value that is used along with a secret key to encrypt data so that the encrypted values are not identical.
HANIS API client timeout (seconds)	The duration till when the Advanced Authentication server waits for a response from the third-party Service Provider.
User ID/Passport attribute	The passport number or national ID of a user against which the validation takes place. You can use custom attribute `workforce ID` of the repository.
User cell phone attribute	The cell phone number of a user that the third-party Service Provider requires for processing the authentication request. You must define the attribute in User cell phone attributes of the Repositories section.
Allow overriding ID/Passport number	Option to prevent users from providing a passport number that is not registered in the LDAP repository. The option is set to **ON** by default. Set to **OFF** to prevent users to specifying the passport number during the enrollment.
Allow overriding 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.
Allow lower resolution fingerprint image scan	Option that enables the Advanced Authentication server to receive the lower resolution fingerprint images that do not comply with standards. The option is set to **OFF** by default. The fingerprint image that does not comply with the standard is not sent to the server for validation. However, if the fingerprint device complies with image standards then the authentication is successful without any issue. When set to **ON**, the Advanced Authentication server receives the lower resolution fingerprint images that do not comply with standards. However, the authentication might not be successful.

NOTE: When you modify the settings related to the HANIS Fingerprint method, ensure to specify the **Password**, **Encryption Key**, and **Encryption initialization vector** to apply the changes.

Scenario for authenticating with the HANIS method

Paul, an end user, wants to authenticate to the new Enrollment portal of Advanced Authentication. He authenticates to the website with the Digital Persona device using one of the enrolled fingers. The National Identification System receives and validates the user details and fingerprint image then shares the validation status. Paul is authenticated to the new Enrollment portal successfully.

7.19 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.

NOTE: You can bypass the password synchronization dialog after the password change or reset by configuring the Password Filter. For configuring the Password Filter, see“Password filter for Active Directory”.

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 specifies the LDAP Password.
- Specify the **SSPR URL**. This URL points to the Self Service Password Reset portal.
- Set **Enable cached logon** to **ON** if you want to validate the user specified password is validated with password stored (cached) in the Advanced Authentication server when the LDAP server is unavailable.

On-Premise	Advanced Authentication as a Service (SaaS)
If the user password does not match with the stored password or password is not stored on the Advanced Authentication server, then cached value gets reset and Advanced Authentication server contacts the LDAP server to validate the user password. If the validation failed, the password stored on Advanced Authentication Server gets reset, so next login will be without cache.	If the user password does not match with the stored password or password is not stored on the Advanced Authentication server, the authentication fails. However, the cached password resets only after exceeding the set **Cached logon offline period**. If the user specified password matches the cached password, the Advanced Authentication server validates user password with LDAP server in the background. After the set **Cached logon offline period**, if the validation fails the password stored on the server gets reset and the subsequent login will be without cache.

When the **Enable cached logon** option is set to **OFF** (default behavior), the Advanced Authentication server always contacts the LDAP server to validate the user password. It may cause performance issues.

- With **Enable cached logon** set to **ON**, you can set the duration until which users can perform offline login when the repository is unavailable in **Cached logon offline period (minutes)**. By default, offline period is set to 60 minutes.

- Set **Disable password change in Self-Service Portal** to **ON** to prevent users from updating their existing LDAP Password in the Self-Service Portal. The option is set to **OFF** by default. Set to **ON** to prevent users to update their LDAP passwords during enrollment.

NOTE: The **Enable cached logon** option works only if any one of the following setting is set to **ON**:

- **Save LDAP password** in the **LDAP Password** method.
- **Enable local caching** in the **Cache Options** policy.

LDAP password is stored on the Advanced Authentication server at the following 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.

7.20 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
- TOTP

You can use the following device or applications for OATH OTP methods:

- Yubikey to generate Hash-based OTP (HOTP)
- Google Authenticator, Microsoft Authenticator, or Advanced Authentication application to generate Time-base OTP (TOTP)

You can configure the following settings for the OATH methods:

- Importing PSKC or CSV files
- CSV file format to import OATH compliant tokens

7.20.1 HOTP

HOTP is a counter based one time password. To configure the HOTP authenticator, 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 size of OTP window defines number of valid OTP for authentication. When the counters are out of sync, this parameter determines the difference between the counter on the token and the server. Based on the difference, the server can recalculate the next OTP value to validate with the OTP received from the token. The server stores the last counter value (C) for which the user has provided a valid password. While verifying a new OTP from the token, the server validates C+1, C+2... until one of the OTP is identical, or till C+w, where w represents the OTP window.

  You can use the HOTP token such as Yubikey token to access not only Advanced Authentication, but also some websites or third-party services. After each use or when users press the token button accidentally, the HOTP counter on the token is increased by 1. Therefore, the counter will be out of sync between the token and Advanced Authentication server.

  For example, if the OTP window is set to 10 (by default), and the current counter value of the server is 100, then any OTP generated from the token with a counter value from 100 to 110 are valid for authentication.

  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 can be used for a long period before the enrollment in Advanced Authentication and the value is unknown. Also, the value can be equal to some thousands.
- Secure because users must provide three consequent HOTPs.

Configuring yubikey for Advanced Authentication server

1. Download and install the Yubikey Personalization Tool from Yubico.

   To download the Yubikey Personalization Tool, see the Yubico website (https://www.yubico.com/products/services-software/download/yubikey-personalization-tools/).
2. Insert the Yubikey token.

   Ensure that the token is recognized. The recognition is indicated by a message `Yubikey is inserted` at the top-right corner of the Personalization tool.
3. Select **OATH-HOTP mode**.
4. Select **Configuration Slot 1**, generate the **OATH Token Identifier** and **Secret Key**.
5. In **Logging Settings**, select **Log configuration output**.
6. Select **Traditional format** or **Yubico format**.
7. Click **Write Configuration** and save the CSV file.

For information about how to enroll the HOTP method, see "HOTP" in the *Advanced Authentication-user* guide.

7.20.2 TOTP

TOTP is a time based one time password. To configure the TOTP authenticator, 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 2*30 seconds before current time and 2*30 seconds after current time, which is ±2 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.

- Set the **Display Rules** option to configure which enrollment option should be displayed to users. Set the one of the following options based on your requirements:
  - **Display Both**: Select this option to display the OATH Token options for entering the user's token details along with the QR code to be scanned using the supported application for enrollment. By default, this option is set to **Display Both**.
  - **Display TOTP Only**: Select this option to display only the QR code that the user needs to scan using the supported application for the TOTP enrollment of the software token.
  - **Display OATH Token Only**: Select this option to display only the OATH Token option allowing users to enter their OATH Token details for enrollment.
- **Google Authenticator QR code (Key URI)**: Option to display the QR code for the TOTP enrollment of the software token in a format that is compatible with the Google Authenticator, Microsoft Authenticator, or the Advanced Authentication apps. When you disable the option, the displayed QR code can be scanned only with the Advanced Authentication app. Enable the option to allow enrollment with the Google Authenticator or Microsoft Authenticator apps. The QR code of Google Authenticator format can also be scanned with the OpenText Auth app (supported by the last iOS and Android apps).

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

- **Allow manual enrollment**: When you enable the option, the **Specify the TOTP secret manually** section is displayed on the TOTP enrollment page of the Self-Service portal with the following parameters: **Secret**, **Period**, and **Google Authenticator format of secret (Base32)**. By default, the option is disabled and the settings are hidden. Enabling the option may result in security risks.

- **Disable self enrollment**: This option allows to disable the manual enrollment of TOTP method in the Self-Service portal. The option is enabled by default. When enabled, the TOTP method is unavailable in the old Self-Service portal and not displayed in the new Self-Service portal.
- **Hide TOTP on rooted smartphones**: Enable this option to hide the OTP in rooted smartphone. By default, the option is disabled.

You must perform the following tasks to allow the users to enroll TOTP method using the OTP Desktop tool:

- Generating an enrollment link
- Sending an enrollment link through Email

Generating an enrollment link

Users can click the enrollment link to enroll the TOTP authenticator automatically on the OTP Desktop tool and following the further steps as described in Desktop OTP tool. To generate an enrollment link, you can encode the server URL, tenant ID, and category name to the Base64 format using any online tool. The generated link is then sent to the users through the email to access the OTP Desktop tool and enroll the TOTP authenticator. The users can create an account on the tool to enroll the TOTP authenticator in the Self-Service portal.

To generate the enrollment link in the base64 format, perform the following steps:

1. To encode use the details such as server URL, tenant ID and category name in the following JSON format:

   `{"server_url":"<domain-name>","tenant_name":"<tenant-name>","category_name": "HOME"}`

   For example, {"server_url": "aafserver.company.com", "tenant_name":"opentext”, "category_name": "HOME"}

   You can specify the preferred category name for `category_name` parameter if you have added categories in the Event categories policy. You can remove the parameter `category_name`, if you have not added any category.

   In case of further problems with the enrollment link, please validate the syntax using Validating JSON syntax in SLAnalyzer.
2. Encode the value including {} to Base64 (charset: UTF-8) format.

   For example, the encoded link is displayed as:

   eyJzZXJ2ZXJfdXJsIjogImFhZnNlcnZlci5jb21wYW55LmNvbSIsICJ0ZW5hbnRfbmFtZSI6Im5ldGlx4oCdLCAiY2F0ZWdvcnlfbmFtZSI6ICJIT01FIn0=
3. Copy the encoded link for further use.

Sending an enrollment link through Email

1. Compose an email with the subject and body.

   For example, specify TOTP Enrollment Link in the Subject and body as follows:

   Hi Users, Click here to enroll for the TOTP authenticator using the OTP Desktop tool.
2. Right click on the preferred text and select **Hyperlink**.
3. Specify the encoded link and prefix `aaf-otp` in **Address**.

For example, aaf-otp:eyJzZXJ2ZXJfdXJsIjogImFhZnNlcnZlci5jb21wYW55LmNvbSIsICJ0ZW5hbnRfbmFtZSI6Im5ldGlx4oCdLCAiY2F0ZWdvcnlfbmFtZSI6ICJIT01FIn0=

4 Specify the email address of the preferred users in **To** then click **Send**.

User can click the hyperlink to open the OTP Desktop automatically.

7.20.3 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 select 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.You can select **Not encrypted**, if the `PSKC` file is not encrypted with either the password or key.

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 **Edit Method** 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.

7.20.4 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**).

7.21 Out-of-band

The Out-of-band method facilitates users to perform out-of-band authentication through the Out-of-band (OOB) portal. Out-of-band authentication allows you to use different supported methods in unusual scenarios.

For example, use fingerprint or card to login to VPN (RADIUS authentication), face recognition or a U2F token to login to an SSH session, and SMS OTP or Smartphone to log in to z/OS mainframe. The Out-of-band method is enrolled automatically.

Advanced Authentication offers the Out-of-band portal where users can manage the authentication requests and perform authentication. This portal displays all authentication requests when a user tries to authenticate with the Out-of-band method. It works similar to the Smartphone method. On the portal, a user can accept or reject the authentication request.

To allow users to access the Out-of-band portal, ensure to meet the following prerequisites:

- Specify the **Hostname** in the `host.domain.com` format during the Advanced Authentication server installation. Ensure, the hostname is resolvable through DNS properly.

  For more information, see **Step 7** in Installing Advanced Authentication.
- Specify the DNS hostname in **My DNS hostname** when you configure the Advanced Authentication server post-installation.

  NOTE: Ensure the DNS name is resolvable by the specified DNS server.

- Upload a valid public SSL certificate for the DNS name on the AA servers or a load balancer in **Server Options**.

  For more information, see Configuring the server options.
- Set the **Public URL** with the hostname of Advanced Authentication server (for example, `https://host.domain.com/`) in **Policies > Public External URL**.
- Assign a chain to the OOB UI logon event.

  For more information, see OOB UI Logon event.

For ease of accessibility, users can install one of the following authentication agents:

- Authentication agent for Windows
- Authentication agent for web

In the **Push notification max age (minutes)** option, you can configure the maximum time (in minutes) until when the push notification is sent to the Authentication Agent for Web or OOB portal on the subscribed device. The subscribed device can be the Authentication Agent for Web on the desktop or Android smartphone. Apple iOS does not support push notifications for the PWA apps. The default value is `525600` minutes (1 year).

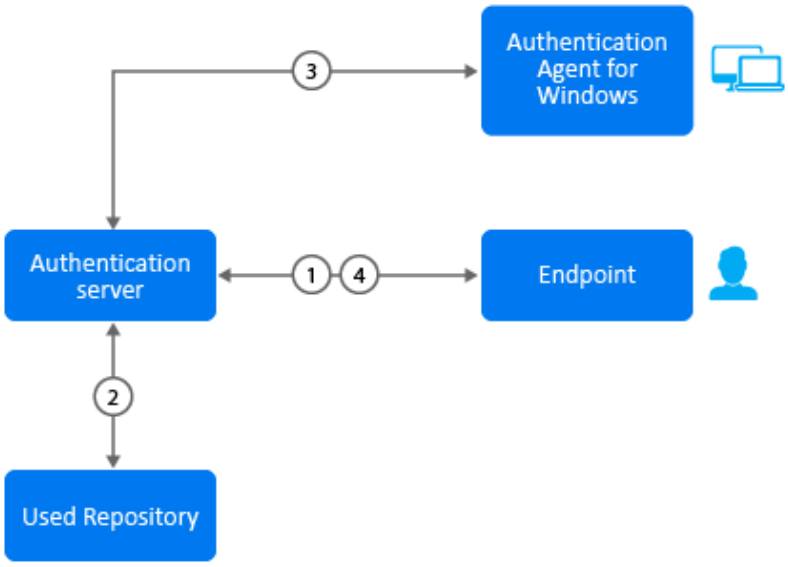
7.21.1 Authentication agent for Windows

Authentication Agent for Windows is supported only on Microsoft Windows. It enables users to perform multi-factor authentication on one device to get authorized access to an event or another device that does not have a user interface or where it is not possible to connect or use a required authentication device.

When a user initiates the out-of-band authentication, an Authentication Agent window appears automatically. User must authenticate using any available chain to access the authentication request with the **Accept** and **Reject** buttons.

For more information, see Advanced Authentication - Windows Authentication Agent.

The following image describes the authentication flow for the Out-of-band method when the Authentication Agent for Window is in use.



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

1. When the authentication request is initiated on the Client side (application, Client, RADIUS, etc), 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 an authentication request to the Windows machine with Authentication Agent for Windows. A restricted browser window prompts to authenticate. User authenticates using any available chain to log in to the OOB portal. The authentication is indicated by the **Accept** and **Reject** options. The user's response is then sent to the server.
4. Finally, the server validates the authentication and the endpoint gets authenticated.

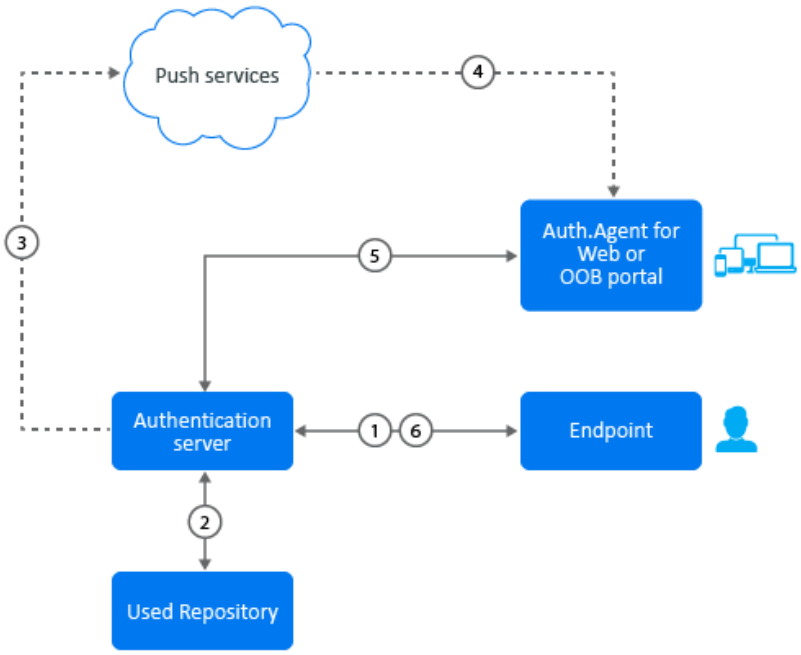
   HTTPS protocol is used for the communication.

7.21.2 Authentication agent for web

A browser-based Progressive Web Application (https://en.wikipedia.org/wiki/Progressive_web_application) (PWA) that can be installed using the Google Chrome browser on any desktop or mobile operating system.

When a user initiates the out-of-band authentication, a push notification is sent on the last subscribed device with the Authentication Agent for Web. The push notification provides information about the pending authentication request. After initiating the out-of-band authentication, the user need not wait for the push notification. However, can access the Authentication Agent for Web or log into the OOB portal to check for the authentication request.

The following image describes the authentication flow for the Out-of-band method when the Authentication Agent for Web is in use.



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

1. When the authentication request is initiated on the Client side (application, Client, RADIUS, etc), 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 the third-party Push services.
4. The third-party Push services forwards the push message to the subscribed device which is an Authentication Agent for Web PWA app or OOB portal.
5. User clicks the push message to open the PWA app or OOB portal, or opens the PWA app or OOB portal manually. Message prompts to authenticate. User authenticates using any available chain to log in to the OOB portal. The authentication is indicated by the **Accept** and **Reject** options. The user's selection is then sent to the server.
6. Finally, the server validates the authentication and the endpoint gets authenticated.

   HTTPS protocol is used for the communication.

IMPORTANT: To receive the push messages, you must enable the notifications in your browser for the OOB portal or Authentication Agent for Web app. By default, the notifications are blocked.

7.22 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 minimum length of the password. The default value is 10.
- **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. The value must not exceed 999 days.
- **Complexity requirements**: Option to enable users to create a complex and not easily detectable password.By default this option is set to **ON**. When enabled, the password must meet three of 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

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.

7.23 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.

Advanced Authentication supports the following two forms of PKI authentication:

- PKI device
- Virtual smartcard

From Advanced Authentication 6.4.2.1, the server facilitates auto-enrollment of PKI smart card based on the value in `altSecurityIdentities` attribute of LDAP repository for a specific user. The **PKI card certificate id attribute** in Group name attributes verifies the value of `altSecurityIdentities` attribute before auto-enrolling the PKI method.

IMPORTANT: Some key points to remember post upgrade to Advanced Authentication 6.4.2.1:

- If a user has enrolled the PKI method earlier, then the existing enrollment takes precedence.
- If a user has not enrolled the PKI method earlier and the `altSecurityIdentities` attribute has an appropriate value in it, then the PKI method gets auto-enrolled. However, a successful authentication using the PKI method happens based on the following factors:
  - Successful certificate mapping
  - Proof of matching certificate and private-key

- If a user has not enrolled the PKI method and the `altSecurityIdentities` attribute does not have an appropriate value in it, then the PKI method does not get auto-enrolled. However, users can manually enroll.
- Auto-enrollment of the PKI method is supported, if the Advanced Authentication Server and Device Service are upgrade to 6.4.2.1 version. Different versions of these components is not supported.
- Auto-enrollment of the PKI method is only supported on the Window Device service.

7.23.1 PKI device

PKI device stores the digital certificates and private keys securely. It uses the PKI infrastructure to store personal details of user such as private key, PIN, and digital certificate.

You can configure the following settings for the PKI method:

- "Adding the trusted root certificates" on page 132
- "Disabling the key-pair option" on page 134

Adding the trusted root certificates

You must upload the trusted root certificates for the PKI method. These certificates must meet the following requirements:

- **Root CA** certificate is in the `.pem` format.
- You can also upload intermediate certificate if the root certificate is not self signed or it is cross signed by another CA.
- All certificates in the certification path (except Root CA) contain **AIA** and **CDP** http link to check revocation status.
- 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.

For more information, see Single Tier PKI Hierarchy Deployment (https://learn.microsoft.com/en-us/archive/technet-wiki/11750.adcs-step-by-step-guide-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** page.
2. Click **Browse**.
3. Choose a `.pem` certificate file and click **Upload**.
4. Click **Save**.

You can configure the PKI method (with certificates) in one of the following ways:

- Standalone Root CA
- Subordinate CA

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 Active Directory certificate services for a standalone root CA

For generating the root CA certificate on Microsoft Windows Active Directory Certificate Services (ADCS), perform the following steps:

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 Active Directory certificate services for a subordinate CA

For generating the subordinate CA certificate on Microsoft Windows Active Directory Certificate Services (ADCS), perform the following steps:

1. Install **Web Server (IIS) Role**.
2. Create the `CertEnroll Folde`r 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.

Disabling the key-pair option

The **Allow key-pair** option is enabled by default. This indicates that the enrollment of the PKI method can be done with either the CA certificates or through the key-pair generation. However, you can disable the key-pair based enrollment of the PKI device and enforce PKI enrollment only using a user certificate issued by the CA. To disable this option, set **Allow key-pair** to **OFF**.

7.23.2 Virtual smartcard

Virtual Smartcard is an extension of PKI method. Advanced Authentication allows users to enroll the PKI method using a virtual smartcard that is imported to the browser on the user's system and used for authentication. Virtual smartcard is a certificate that contains information, such as digital signature, expiration date, name of user, name of CA (Certificate Authority), and can be used in client SSL certificate. Typically, the certificate is available in `.pfx` format. The information available in the virtual smartcard is used to authenticate the user to any web environment.

NOTE: The virtual smartcard supports authentication to the OAuth 2.0 and SAML 2.0 events. The virtual smartcard does not support authentication to Advanced Authentication portals, such as Administration, Helpdesk, Self-Service, and Reporting.

To configure the virtual smartcard, perform the following steps:

NOTE: Before you configure the virtual smartcard support for the SAML 2.0 events, ensure to specify the **Identity Provider's URL** in format `https://webauth.domain_name` in the Web Authentication policy. Later, save the settings before downloading the SAML 2.0 metadata file.

NOTE: Before you configure virtual smartcard support for the PKI method, ensure to perform the following tasks:

- Resolve the IP address of Advanced Authentication server with the following host names on the DNS server:
  - `<aaserver_ip_address> <aaserver_hostname>`
  - `<aaserver_ip_address> <webauth.aaserver_hostname>`

- Define the following attributes in the third-party application that you want to integrate with Advanced Authentication server:
  - `authorization_endpoint = https://webauth.aaserver-hostname/osp/a/TOP/auth/oauth2/auth`
  - `token_endpoint = https://webauth.aaserver-hostname/osp/a/TOP/auth/oauth2/token`

---

1 Configure the following settings in the **HTTPS Options** policy:
   - Set **Enable Client SSL for Webauth Service** to **ON** and upload Root CA certificate in the `.pem` format that is used by the Web server.
   - Set **Enable auto enrollment based on certificate** to **ON**. This enables you to allow users to auto-enroll the PKI method using virtual smartcard for the **OAuth 2.0** and **SAML 2.0** events.

     NOTE: The manual enrollment of the PKI method using the virtual smartcard is not supported. Therefore, it is required to set **Enable auto enrollment based on certificate** to **ON** in the **HTTPS Options** policy. With this configuration, the users can auto-enroll PKI method using virtual smartcard when they access **OAuth 2.0** event for the first time and select a valid certificate. This auto-enrollment happens irrespective of enrollment status of other method(s) that are available with the PKI method in the same authentication chain.

     To allow a user to login to the **OAuth 2.0** and **SAML 2.0** events before auto-enrolling the PKI method, ensure to add at least one more chain to the event (for example, a chain with only the LDAP Password method) below the PKI chain. The user must enroll all method(s) of new chain. During the first login attempt, the PKI method using the virtual smartcard gets enrolled automatically. For the sub-sequent log ins, the top chain in the list (which is PKI) is selected and user is authenticated automatically.

     ---

2 Upload Root CA certificate in the **Trusted root certificates** section of **PKI** method.

3 Import the client SSL certificate to the users browser.

   NOTE: The procedure to import the client SSL certificate varies on each browser.

   For more information about how to import the client SSL certificate to the Chrome browser, see Importing client SSL certificate to a certificate store.

An example of auto-enrolling PKI method with the virtual smartcard

Consider the administrator has performed the following steps to allow auto-enrollment of the PKI method using the virtual smartcard:

- Created a chain with the PKI method and another chain with preferred methods such as **LDAP password** and **Password**.
- Mapped the chain to the **OAuth 2** event.
- Configure the following settings in the **HTTPS options** policy:
  - Set **Enable SSL Client Certificate** to **ON** and uploaded a valid CA certificate.
  - Set **Enable Auto Enrollment based on certificate** to **ON**.
- Imported the client certificate to the user's browser in the `.pfx` format containing details, such as digital signature, expiration date, name of user, name of CA and so on.

Mark, an end user, wants to auto-enroll the PKI method using the virtual smartcard. When he tries to access the `somecompany.com` website, the user name stored in the certificate gets filled in the user name field in the login form automatically. Mark is required to select the preferred certificate to validate his identity in the **User Identification Request** dialog box. Then, Mark must specify LDAP details for additional validation. If the specified details are valid, Mark gets auto-enrolled to the PKI method using the virtual smartcard without physical PKI token.

During subsequent logins, Mark might experience one of the following scenario:

- If there is a chain with only PKI method associated to the web authentication event, then Mark gets authenticated automatically.
- If there are more than one chain associated to the web authentication event, then Mark is prompted with the list of chains that contains PKI in addition to other available chains. In this case, he can select the chain with only PKI method to authenticate automatically or select preferred chain and provide corresponding details to authenticate successfully.

Importing client SSL certificate to a certificate store

To enable and achieve the virtual smartcard authentication to the web environment, it is required to import the Client SSL certificate to the browser.

NOTE: The procedure to import the client SSL certificate varies on each browser.

To import the client SSL certificate to Google Chrome browser, perform the following steps:

1. Navigate to **Settings > Manage Settings**.

   The **Certificates** wizard is displayed.
2. Click **Import** and select the client SSL certificate.

   Ensure that the certificate is in `.pfx` format.
3. Click **Next** and **Finish**.

   A message `Certificate has been imported successfully` is displayed.

7.24 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:

- **Send the 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.

- : Specify the number of seconds till when the RADIUS client waits for the RADIUS server to reply before prompting an error `Connection time out`. The default value is 5 seconds.
- **Retries count**: Specify the number of times, the RADIUS client tries to connect to the RADIUS server. If a connection is not established during the retry attempts, a message `Failed to connect to the server` is displayed. The default value is set to 3. If set to 0, the RADIUS client does not try to connect after the first unsuccessful attempt.
- **Specify servers per site**: Option to configure the third-party RADIUS servers that are specific to a site. When set to **ON**, the sites available in the cluster are populated and you can add more than one servers to the preferred site.

  When this option is set to **OFF**, you can add single third-party RADIUS server details that are applicable for all sites in the cluster by specifying the following details:

  - **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.

7.25 SAML service provider

Advanced Authentication facilitates you to authenticate with SAML 2.0 with the Web Authentication method.

WARNING: You must configure the SAML Service Provider method with the relevant Identity Provider details before adding it to an authentication chain.

NOTE: A chain with the SAML Service Provider method can be assigned to the OAuth 2.0 and SAML events. Ensure to meet the following points:

- The event must contain the **Support Authorization Code** enabled in the Advanced Settings section.
- The SAML Service Provider method can be single or the first method in the chain. Even if it is not the first method in the chain, it will be requested before the other methods.
- The user who authenticates using the SAML SP method must be present in only one repository.
- The SAML Service Provider method is not enrolled automatically when using the new Enrollment Portal. It must be enrolled for users before authentication.

To configure the SAML Service Provider method for Advanced Authentication, perform the following steps:

1. Click **Methods > SAML Service Provider**.
2. Click **Add** in **Identity providers**.
3. Select **SAML** in **Authentication type**.
4. Click the arrow → icon.
5. Specify the identity provider name in **Identity Provider**.

6 Specify the attribute name used in the SAML assertion that identifies the user in **Assertion Attribute**. By default it is set as username.

7 Click **Choose File** to upload the **Identity Provider Metadata** file.

IMPORTANT: Ensure that you choose the Identity Provider Metadata file that is exported from a used Identity Provider. Do not use the metadata file exported from the **Administrative Portal > Policies > Web Authentication**.

8 Click the save icon.

9 Click **Save**.

NOTE: You can obtain Service Provider metadata from Advanced Authentication. Use the URL mentioned below to obtain the Service Provider metadata:

`https://AAF_SERVER/osp/a/TENANT/auth/saml2/metadata`.

In the above URL, the TENANT must be replaced by the actual tenant name. Use TOP as the TENANT name if you are not using the Advanced Authentication as SaaS version or the multi-tenancy feature is not enabled.

Sample configuration

Lets assume an organization requires to secure an OAuth2 event with SAML Service Provider method and want to add OpenText Access Manager as Identity Provider for validating users' identity. To achieve this administrator must configure the following:

1. Configure Advanced Authentication server
2. Configure Access Manager server
3. Verify the SAML service provider method

Configure Advanced Authentication server

Ensure to download the Access Manager metadata file as a prerequisite. Use the below link syntax to download the metadata:

https://<NAM IDP URL>:<Port>/nidp/saml2/metadata

NOTE: Ensure to replace `<NAM IDP URL>` and `<Port>` with the valid details.

1 Navigate to **Methods > SAML Service Provider** on the Advanced Authentication Administration Portal.

2 Click Add.

The **Authentication type** is set to **SAML** by default.

3 Click icon.

4 Specify the following:

- **Identity Provider**: Name of Identity Provider.

  In this example, Access Manager.

- **Assertion Attribute**: Attribute name in SAML assertion.

In this example, username.

- **Identity Provider Metadata file**: Click **Choose File** and upload the Access Manager metadata file.

5 Click icon and **Save**.

6 Create a chain with SAML Service Provider method in **Chains**.

7 Map the chain with SAML Service Provider method to **OAuth2** event in **Events**.

Configure Access Manager server

Ensure to obtain the Advanced Authentication metadata file from **Policies > Web Authentication** as a prerequisite.

1 Navigate to **Devices > Identity Provider > Edit** on the Access Manager Administration Portal.

2 Click **SAML 2.0**.

3 Click **New > Service Provider**.

4 Specify the following:

- **Provider Type**: Select General.
- **Source**: Metadata Text
- **Name**: Name of the Service Provider. In this case, Advanced Authentication.
- **Text**: Paste the SAML2 metadata of Advanced Authentication.

5 Click **Next** and **Finish**.

6 Import the Signing and encryption certificates from Advanced Authentication to Access Manager. Later add the certificate to NIDP Trust store (Optional, if the self-signed certificate is in use).

7 Navigate to **Devices > Identity Server > Shared Settings > Attribute Sets > New** to create an attribute set.

8 Specify the name of attribute set in **Set Name**.

9 Click **Next**.

10 Click **New** to add an attribute to the set.

11 Perform the following:

- **Local attribute**: Select **Ldap Attribute:cn [LDAP Attribute Profile]** from the list.
- **Remote attribute**: Specify the attribute as username.

  This attribute must match the Assertion Attribute configured in **SAML Service Provider** method in Advanced Authentication.
- **Remote namespace**: Select **none**.

NOTE: Retain the default value for other options.

---

12 Click **OK** and **Finish**.

13 Assign the attribute set to SAML 2 service provider that you created and move the attribute from **Available list** to **Send with Authentication**.

14 Save the changes and update Identity Server.

Verify the SAML service provider method

When users access the Oauth2 event, users are redirected to Access Manager for authentication. After the authentication in Access Manager, Advanced Authentication receives the authentication response (Success or Fail). Based on the authentication response Advanced Authentication grants access to Oauth2 application.

NOTE: Advanced Authentication does not prompt for any additional authentication unless there are other methods in the chain in addition to the SAML SP method.

7.26 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 (http://goodsecurityquestions.com/) are as follows. For a full list of examples, see the website 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:

- **Minimum 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.

7.26.1 Adding questions

You can add questions based on your requirement. These questions can be translated in languages that are supported by the Advanced Authentication portals. For example, you set a security questions as **What is your pet name?**. While enrolling and authenticating, this question will be displayed in the language that the user selects in the portal.

To add questions, perform the following:

1. Click **Add** to add a question in the **Question** window.
2. Specify the question in **Question**.
3. You can specify the question to be translated in the required language.

   This translated question is displayed in the portals and Clients based on the selected language.
4. Click the save  icon to save the question related settings.

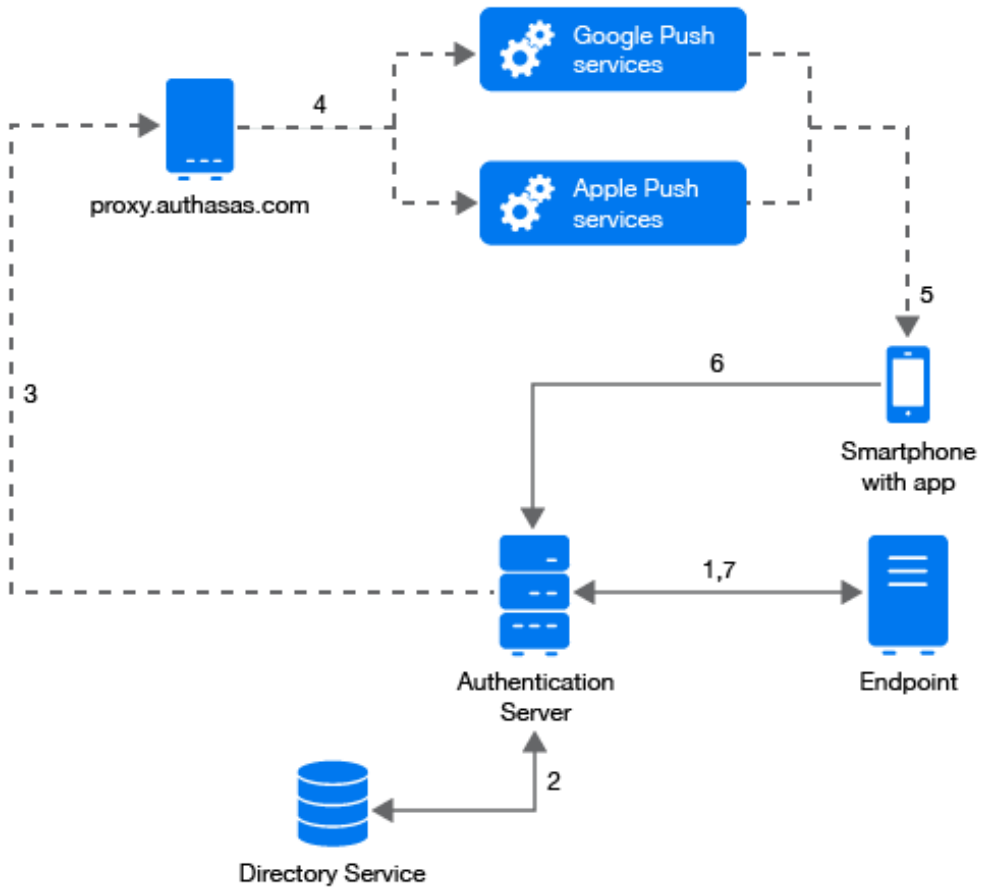
You can add more questions depending on the requirement.

Click **Save** to save the configuration settings for the Security questions method.

7.27 Smartphone

Advanced Authentication provides the **Smartphone** method that facilitates users to authenticate through their Smartphone. The authentication happens through the OpenText Advanced Authentication application 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.authasas.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.

NOTE: To use Smartphone authentication method, at least one Advanced Authentication server from each site (if any) should be accessible from the Public URL.

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 his mobile phone. When he opens the Smartphone app installed on his phone, the **Accept** and **Reject** buttons are displayed. 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 authenticates to **myexample.com**.

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

This section covers the following configurations related to the Smartphone method:

- Section 7.27.1, “Configuring Smartphone method,” on page 143
- Section 7.27.2, “Configuring enrollment link,” on page 148
- Section 7.27.3, “Setting up geo-fence for Smartphone,” on page 148
- Section 7.27.4, “Priority vendor requirements,” on page 149

NOTE: You can customize the authentication request message that is displayed on the OpenText Advanced Authenticationapplication using the Custom messages policy.

For more information about customizing the authentication request message, see Customizing authentication request message for Smartphone method.

7.27.1 Configuring Smartphone method

To configure the Smartphone method, specify the following details:

Parameter	Description
Learn timeout	The time that is valid for the user to scan the QR code for enrollment. he default timeout is 60 seconds.
TOTP Length	The length of OTP token used for backup authentication. The default length is 6 digits.
TOTP step	The time a TOTP is displayed on a screen before the next OTP is generated. The default time is 30 seconds.

Parameter	Description
TOTP time window	The time in seconds in which the specified TOTP 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** for authenticating to the Smartphone application. A user cannot edit the settings on the application. **NOTE:** If the PIN is not set, then the user is prompted to set the PIN on launching the app. On the first launch of the app, the user must set the PIN irrespective to the settings.
Minimum PIN length if the PIN is required	The minimum length of the PIN. The available options are 4,5, and 6.
Require biometrics	Set to **ON** to enforce the fingerprint or facial recognition settings for authenticating to the Smartphone application. A user cannot edit the settings on the application. **NOTE:** Before Advanced Authentication 6.3 Service Pack 5 Patch 1 enabling **Require biometrics** enabled the **Require PIN** option. Also, it was not possible to disable **Require PIN** without disabling **Require biometrics**. Following are different possibilities of using the **Require biometrics** and **Require PIN** options: ◆ Both **Require PIN** and **Require Biometrics** are set to **OFF** A user can disable both settings in the smartphone application if required. ◆ **Require PIN** is set to **ON** and **Require Biometrics** is set to **OFF** In this case, user must set PIN to authenticate to the smartphone application and can disable biometric if required. ◆ Both **Require PIN** and **Require Biometrics** are set to **ON** In this case, if biometric is available user must use it or use PIN code to authenticate to the smartphone application. User cannot change any settings in the application. ◆ **Require PIN** is set to **OFF** and **Require Biometrics** is set to **ON** In this case, user must always use biometrics to authenticate to the smartphone application. If the biometrics is not available, then user cannot use the application.

Parameter	Description
Enroll TOTP method when enrolling Smartphone	Set to **ON** to enable enrolling both the Smartphone and TOTP methods during the Smartphone method enrollment. After enrollment, the OpenText Advanced Authentication application on the user's Smartphone displays only one authenticator. However, it corresponds to both Smartphone and TOTP authenticators enrolled on the Self Service Portal. **IMPORTANT:** Even if you set the option to **OFF**, the user can use the Smartphone method in following ways: 1. Out-of-band: Sending a push notification and accepting it on the user's Smartphone. 2. OTP: Open the list of enrolled authenticators in the OpenText Advanced Authentication application, and use the one-time password if the user is not able to use the out-of-band option. For Example, when there is no internet connection on the Smartphone.
Allow to accept/reject authentication through push notification	Set to **ON** to display the action buttons **Accept** and **Reject** with the notification in the mobile notification bar. This allows users to take action directly from the notification without opening the app. This option is applicable for Android and iOS versions of the OpenText Advanced Authentication application. **NOTE:** After enrolling the Smartphone method, for the first authentication the actions buttons are not displayed with the notification in the notification bar. Therefore, the user is required to launch the OpenText Advanced Authentication application to accept or reject the request.
Prevent login from a rooted device	Set to **ON** to enable a root check for mobile devices. The smartphone app must detect whether the device is rooted and prevent login from that device. Rooted devices can provide administrative privileges to third-party software that is not secured and mostly not allowed by device vendors.

Parameter	Description
Use image on mobile devices	Select the option to use a customized image on your Smartphone app. 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. **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. ◆ 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.
Disable Offline OTP Options **NOTE:** In Advanced Authentication 6.4 SP1 and prior versions, the label is **Disable Offline Authentication**.	Select this option to disable users from authenticating with the Smartphone TOTP. By default this option is disabled and users are allowed to log in using Smartphone even when without the network. Enabling this option prevents users from using the One-Time Password of the Smartphone method to login to the offline mode.
Allow as first authentication method	Option that allows a user to authenticate using a chain where Smartphone authenticator is the first authentication method. The option is set to **ON** by default. Set this option to **OFF** to prevent user from authenticating using a chain where Smartphone authenticator is the first authentication method. If the option is set to **OFF**, and a user tries to authenticate using a chain where the Smartphone method is the first authentication method, the user is displayed a `The method cannot be first in the login chain` message and the user cannot authenticate.
Advanced Settings	These settings are optional.

Parameter	Description
Default Vendor	The Default Vendor is set to `OpenText` and this vendor sends the push notifications to the OpenText Advanced Authentication application for users to complete the Smartphone authentication. **NOTE:** You can add only the approved vendor as a default vendor. A certificate for your custom application must be provided to Micro Focus and be applied to `proxy.authasas.com`.
Priority Vendor	Click **Add** to add the preferred vendor as a priority vendor that sends push notifications to the custom smartphone application. To understand the requirements to add the priority vendor, see Priority vendor requirements. **NOTE:** You can add only the approved vendor as a priority vendor. A certificate for your custom application must be provided to Micro Focus and uploaded to `proxy.authasas.com`. Before adding a priority vendor, the default vendor manages all the smartphone enrollment and authentication requests. After you add a priority vendor, new enrollment requests get associated with the priority vendor. Even after adding the priority vendor, the default vendor continues to process the authentication requests of the enrollments that were associated earlier. If you add more than one priority vendor, then the **Vendor** list appears for the user to select the preferred vendor while enrolling the Smartphone method on the Self Enrollment portal.
Google project ID	You can specify Google Project ID for your Android app if you have an approved vendor and the Private key (in JSON format) has been generated, provided to Micro Focus, and applied on the `proxy.authasas.com`. The push notifications are sent only to the application which matches the configured Google Project ID.
Geo Zones	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. To set up the Geo-fence, see Setting up geo-fence for Smartphone.

NOTE: To use geo-fencing, ensure that access to the location is enabled for the OpenText Advanced Authentication application on the smartphone.

To configure the Smartphone method as second factor authenticator to secure Windows workstation, see

 http://www.youtube.com/watch?v=u6O8xnJN7hg

NOTE: The OpenText Advanced Authentication application icon displayed in the video has been updated. However, the concept and configuration steps remain same.

7.27.2 Configuring enrollment link

Users can enroll the Smartphone method either by a QR code or through a link sent to their email or SMS. You as an administrator must configure the link and send it to all the users whom you want to enroll the authenticator. You can use one of the following links as per the requirement:

`https://<public_external_url>/smartphone/enroll`

`https://<public_external_url>/smartphone/enroll?category=cat1`

`https://<public_external_url>/smartphone/enroll?tenant=t1`

`https://<public_external_url>/smartphone/enroll?category=cat2&tenant=t1`

Default category is `default`. For more information about the category, see Event categories.

Default tenant is `TOP`. For more information about the tenant, see Multitenancy Options (https://www.netiq.com/documentation/advanced-authentication-63/server-administrator-guide/data/multitncy_opts.html).

To allow users to enroll the Smartphone method using the link, ensure to configure the Smartphone enrollment event.

7.27.3 Setting up geo-fence for Smartphone

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:

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.

   You can click the full-screen icon to view the map in the full screen.
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.

   You can also edit the marked polygon by clicking the edit icon.
7. Click **Save**.

7.27.4 Priority vendor requirements

OpenText sends the push notification to the smartphone application by default. You can add a priority vendor that sends the push notification to the customized iOS application during the authentication. Before adding a priority vendor, you must generate the .p8 key file, obtain Key ID, Bundle ID, and Apple Team ID. The .p8 key file enables the configured priority vendor to send the push notification to any iOS application.

To generate the .p8 key file, perform the following steps:

1. Sign in to your Apple Developer account.
2. Navigate to **Certificates, IDs & Profiles > Keys**.
3. Click + icon to add a key.
4. Specify name of the key in **Key Name**.
5. Select **Apple Push Notification service (APNs)**.
6. Click **Continue** then click **Register**.
7. Click Download to download the key file.

   The name of key file includes the Key ID.

   For example, the file name is `AuthKeyABCD1234.p8`, the `ABCD1234` represents the Key ID.

   NOTE: Ensure to store and secure the Key file for further use because you cannot download the key file again.
8. Click Membership and gather the TeamID.
9. Click **Certificates, IDs & Profiles > Identifiers**.
10. Select the name of the application to view the **Bundle ID**.

    NOTE: The Bundle ID is created by the customer who develops the application. Apple allows developers to use reverse domain name notation deriving the bundle identifier for the application.

    For example, the Bundle ID for the domain `abc.com` can be `com.abc.<appname>`.

    To proceed, you must share the above details with the Micro Focus Support team to configure the proxy server.

7.28 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.

When authenticating on the same smartphone that receives SMS, a user can do one of the following actions based on the platform of smartphone:

- **iOS**: The OTP is auto inserted to clipboard, tap on the input field > OTP displayed above the keyboard.

- **Android 11 and later versions**: Tap Copy <OTP> in the SMS notification then tap on the input field > OTP displayed above the keyboard.
- **Android 10 and prior versions**: Open the SMS notification and copy OTP. Tap on the input field > OTP displayed above the keyboard.

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:
  - {otp}: One-Time Password.

    NOTE: In **Body**, the {otp} variable must be placed first to allow Android or iOS to capture the OTP to clipboard.

  - {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.
  - {number}: Sequence of the OTP, user is required to specify to authenticate.

  The text in {} contains variables. {otp} is a required variable, and the other variables are optional. Apart from the default variables, the custom variables are not supported. You can customize the text outside {}.

  For example:

  1. Company Name: a one-time password for multifactor authentication: {otp}.
  2. {user} is trying to login to {event}. Please approve it by using the OTP: {otp}. Security Department.
- **Allow re-sending after (seconds)**: The duration from previous OTP to re-send a fresh OTP for authentication.
- **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 overriding 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.
- **Verify phone number**: Option that sends the verification code to a specified phone number and allows users to validate the phone number during the manual enrollment. The option is set to **OFF** by default. Set this option to **ON** to permit users to check whether the phone number is valid before the enrollment.
- **Allow user enrollment without a phone**: Option to configure settings for the user to enroll the SMS OTP authenticator without a phone number in the repository.

  Set this option to **OFF** to ensure that a user does not enroll the SMS OTP authenticator without a phone. The user is prompted with an error message that you can specify in **Error message**.

  Set this option to **ON** to allow the user to enroll the SMS OTP authenticator without a phone.

  If the user's phone number is available in the repository, the account gets enrolled automatically.
- **Allow as first authentication method**: Option that allows a user to authenticate using a chain where SMS OTP authenticator is the first authentication method.

  The option is set to **ON** by default. Set this option to **OFF** to prevent user from authenticating using a chain where SMS OTP authenticator is the first authentication method.

  If the option is set to **OFF**, and a user tries to authenticate using a chain where the SMS OTP method is the first authentication method, the user is displayed a `The method cannot be first in the login chain` message and the user cannot authenticate.

NOTE: After configuring the SMS OTP method, it is required to configure the SMS sender policy to deliver the SMS OTP to users.

7.29 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 (http://www.mobileid.ch/en/login).

For more information about the Swisscom Mobile ID method, see the Mobile ID Reference guide (http://www.mobileid.ch/en/documents).

7.30 FIDO U2F

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

Advanced Authentication supports the Microsoft policy Interactive logon: Smart card removal behavior (https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-R2-and-2012/jj852235(v=ws.11)?redirectedfrom=MSDN) 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 (https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-R2-and-2012/jj852235(v=ws.11)?redirectedfrom=MSDN).

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.

NOTE: Ensure to set a valid domain name for your Advanced Authentication server rather than an IP address and host the domain name appropriately before users authenticate to any event or device using the U2F method.

You can configure the following settings for this method:

- Section 7.30.1, “Configuring the certificate settings,” on page 153
- Section 7.30.2, “Configuring Facets,” on page 153
- Section 7.30.3, “Configuring yubikey for Advanced Authentication server,” on page 154
- Section 7.30.4, “Configuring a web server to use the FIDO U2F authentication,” on page 154

7.30.1 Configuring the certificate settings

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.

To validate the attestation certificate for the FIDO U2F authentication, perform the following steps:

1 Set **Require attestation certificate** to **ON** to enable validation of attestation certificate.

2 Select the attestation certificate:

2a To use a default certificate, click **Add Default**.

2b To use a custom certificate instead of predefined device manufacturer certificate, perform the following steps:

2b1 Click  next to the default attestation certificate to remove the certificate.

2b2 Click **Add** to add a custom certificate.

2b3 Click **Browse** then select the custom certificate and click **Upload**.

The certificate must be in the `PEM` format.

To restore the deleted attestation certificate, click **Add Default**.

7.30.2 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 the primary domain `https://yourdomain.com` and wants to get authenticated on sub-domain `https://app.yourdomain.com`, you as an administrator can add `https://app.some.domain` as a facet of the primary domain `https://yourdomain.com`.

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 pprompted with an error, `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 Click **Add** and specify the facet in **Facets**.

For example, you can specify `https://app.yourdomain.com`.

You can add more than one facets.

3 Specify the main URL in **App ID**. This ID is used to identify applications.

For example, `https://yourdomain.com`.

A single facet can be configured using **App ID**. In case of multiple facets, those facets will be associated by the AppID. If the AppID is empty then the first listed facet will be used as default AppID.

From the above example, if a user attempts to log in to `https://app.yourdomain.com` with the U2F token enrolled on `https://yourdomain.com`, 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** list.

To support legacy usages in Chrome or AA native clients, the AppID is required to be listed in Facets when there are multiple facets in use.

4 Click **Save**.

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

7.30.3 Configuring yubikey for Advanced Authentication server

1 Download and install the Yubikey Personalization Tool from Yubico.

To download the Yubikey Personalization Tool, see the Yubico website (https://www.yubico.com/products/services-software/download/yubikey-personalization-tools/).

2 Insert the Yubikey token.

Ensure that the token is recognized. The recognition is indicated by a message `Yubikey is inserted` at the top-right corner of the Personalization tool.

3 Select **Yubico OTP mode**.

4 Select **Configuration Slot 1**, generate the **Public Identity, Private Identity, and Secret Key.**

5 Click **Write Configuration** and specify the configurations.

6 Open the Advanced Authentication Self-Service portal and select U2F method.

7 Click **Save** to complete the enrollment.

7.30.4 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 OpenText 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 OpenText 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 OpenText 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 OpenText 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”.

7.31 Voice

In the **Voice** authentication method, a user receives a call with a PIN request, after which the user must specify the PIN on his or her phone.

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 PIN request.
3. User must specify the PIN that has 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.

- **Allow overriding 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.
- **Allow user enrollment without a phone**: Option to configure settings for the user to enroll the Voice authenticator without a phone number in the repository.

  Set this option 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 this option to **ON** to allow the user to enroll the Voice authenticator without a phone.
- **Allow as first authentication method**: Option that allows a user to authenticate using a chain where Voice authenticator is the first authentication method.

  The option is set to **ON** by default. Set this option to **OFF** to prevent user from authenticating using a chain where Voice authenticator is the first authentication method.

  If the option is set to **OFF**, and a user tries to authenticate using a chain where the Voice method is the first authentication method, the user is displayed a `The method cannot be first in the login chain` message and the user cannot authenticate.

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

NOTE: After configuring the Voice method, it is required to configure the Voice sender policy to call the user with a PIN request.

7.32 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.

  NOTE: From Advanced Authentication 6.3 Service Pack 6, the maximum value for the OTP period is 86400 seconds (1 day).

- **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. You can specify the following variables:
  - `{otp}`: One-Time-Password to be sent to the user.

    To repeat the one-time password during the call, you can specify: `Use the OTP for authentication: {otp}. OTP: {otp}.`
  - `{number}`: Sequence of the OTP, user is required to specify to authenticate.

    To include the sequence of OTP and repeat the one-time password, you can specify: `Your One-Time Password number {number} and the OTP is {otp}, one more time: {otp}.`
- **Allow re-sending after (seconds)**: The duration from previous OTP to re-send a fresh OTP for authentication.
- **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" "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 overriding 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.
- **Verify phone number**: Option that sends the verification code to a specified phone number and allows users to validate the phone number during the manual enrollment. The option is set to **OFF** by default. Set this option to **ON** to permit users to check whether the phone number is valid before the enrollment.
- **Allow user enrollment without a phone**: Option to configure settings for the user to enroll the Voice OTP authenticator without a phone number in the repository.

Set this option to **OFF** to ensure that a user does not enroll the Voice OTP authenticator without a phone. The user gets an error message that you can specify in **Error message**.

Set this option to **ON** to allow the user to enroll the Voice OTP authenticator without a phone.

- **Allow as first authentication method**: Option that allows a user to authenticate using a chain where Voice OTP authenticator is the first authentication method.

  The option is set to **ON** by default. Set this option to **OFF** to prevent user from authenticating using a chain where Voice OTP authenticator is the first authentication method.

  If the option is set to **OFF**, and a user tries to authenticate using a chain where the Voice OTP method is the first authentication method, the user is displayed a `The method cannot be first in the login chain` message and the user cannot authenticate.

NOTE: After configuring the Voice OTP method, it is required to configure the Voice sender policy to deliver OTP over a call to users.

To configure the Voice OTP method as the second factor authenticator to secure Windows workstation, see

http://www.youtube.com/watch?v=sjj4R3QLSfc

7.33 Web authentication method

Advanced Authentication facilitates you to authenticate with different Identity Providers, such as OAuth 2.0, OpenID Connect, and SAML 2.0 with the Web Authentication method. The Web Authentication method uses browser and http based authentication protocols and can be used in web environment or hybrid applications.

NOTE: Ensure that you use a valid certificate for the Advanced Authentication server. Users may face enrollment issues on the Internet Explorer and Microsoft Edge browsers, if the certificates are not valid.

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

1. Click **Methods > Web Authentication**.
2. Click **Add** in **Identity providers**.
3. Select the **Authentication type**.
4. Click the arrow → icon.

Web authentication method supports the following authentications:

- **OAuth 2.0**: Advanced Authentication applies RFC6749 for OAuth 2.0 authentication. For more information, see https://github.com/OpenIDC/pyoidc (https://github.com/OpenIDC/pyoidc).

- **OpenID Connect**: Advanced Authentication uses OpenID Connect Core specification 1.0. For more information, see https://openid.net/specs/openid-connect-core-1_0.html (https://openid.net/specs/openid-connect-core-1_0.html).
- **SAML 2.0**: Advanced Authentication implements SAML 2.0 on top of xmlsec and python-saml. For more information see, https://github.com/mehcode/python-saml (https://github.com/mehcode/python-saml). This implementation of SAML protocol does not completely comply to SAML 2.0 standards (https://www.oasis-open.org/standards/#samlv2.0) but is compatible with Microsoft ADFS.

You can configure the Web Authentication method to use the following Identity Providers:

- SAML
- OpenID Connect
- OAuth 2.0

7.33.1 SAML for Advanced Authentication

To add the SAML Identity Provider, perform the following steps:

1. Specify the identity provider name in **Identity Provider**.
2. Select the **Available presets for Name ID Format**.

   The **Name ID Format** is automatically populated.

   or

   Specify manually in **Name ID Format**.
3. Click **Browse** to upload the **Identity Provider Metadata file**.

   WARNING: Ensure that you choose the Identity Provider Metadata file that is exported from a used Identity Provider. Do not use the metadata file exported from the **Administrative Portal > Policies > Web Authentication**.

   NOTE:

   - The Web Authentication method supports only HTTP-POST for the Single Sig-On (SSO) Service Binding parameter in the metadata file. The HTTP-Redirect is not supported.
   - If you upgrade to Advanced Authentication 6.4 or later versions whereas the WebAuth method is already configured, you must update the Advanced Authentication's metadata in your IDP to include a single logout service.

   ---
4. Click the save  icon.
5. In the **Upload SAML Service Provider signature certificate** section, you must upload a certificate file in the `PEM` format with a private key. This certificate is used by the Web Authentication method to sign a SAML `AuthnRequest` token.

   If the private key is protected by a password, specify the password in **Private key password**.
6. Click **Save**.

An example configuration with ADFS

Perform the following steps to add ADFS as an Identity Provider for the Web Authentication method.

1 Specify **myexample-adfs** as the **IdP provider name**.

2 Select **urn:oasis:names:tc:SAML:1.1:nameid-format:WindowsDomainQualifiedName** from **Available presets for Name ID Format**.

The selected **Name ID Format** will be extracted from the SAML `AuthnResponse` token and saved as an authentication data (unique data which will be associated with the user).

3 Click **Browse** to upload the **IdP Metadata file** from the ADFS server.

4 Click the save icon.

5 In the **Upload SAML Service Provider signature certificate** section, upload a certificate file in the `PEM` format with a private key.

If the private key is protected by a password, specify the password in **Private key password**.

6 Click **Save**.

Configuring the ADFS Identity Provider

1 Save the Service Provider metadata from Advanced Authentication to a file. Use the URL mentioned below to obtain the Service Provider metadata:

```
https://AAF_SERVER/webauth/TENANT/metadata
```

NOTE: The default TENANT is TOP. Use TOP as TENANT if you are not using multi-tenancy.

A sample Service Provider metadata is mentioned below:

```
<md:EntityDescriptor xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
ID="_7a8608ad1cfbc149" entityID="https://www.d18r14.tk/webauth">
<md:SPSSODescriptor
protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol
urn:oasis:names:tc:SAML:1.1:protocol
urn:oasis:names:tc:SAML:1.0:protocol">
<md:KeyDescriptor>
<ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:KeyName>https://www.d18r14.tk/webauth</ds:KeyName>
<ds:X509Data>
<ds:X509Certificate>
MIIEOzCCAyOgAwIBAgIJAJcsrIQZzcT0MA0GCSqGSIb3DQEBCwUAMIGyMQswCQYD
VQQGEwJDSDEcMBoGA1UECAwTR3JlYXRlciBadXJpY2ggQXJlYTEPMA0GA1UEBwwG
WnVyaWNoMRcwFQYDVQQKDA5NaWNybyBGb2N1cyBBRzERMA8GA1UECwwIQXV0aGFz
YXMxFzAVBgNVBAMMDm1pY3JvZm9jdXMuY29tMS8wLQYJKoZIhvcNAQkBFiBhbGV4
YW5kZXIuZ2FsaWxvdkBtaWNyb2ZvY3VzLmNvbTAgFw0xNjA1MjAwOTMyMzlaGA8y
MTE2MDQyNjA5MzIzOVowgbIxCzAJBgNVBAYTAkNIMRwwGgYDVQQIDBNHcmVhdGVy
IFp1cmljaCBBcmVhMQ8wDQYDVQQHDAZadXJpY2gxFzAVBgNVBAoMDk1pY3JvIEZv
Y3VzIEFHMREwDwYDVQQLDAhBdXRoYXNhczEXMBUGA1UEAwwObWljcm9mb2N1cy5j
b20xLzAtBgkqhkiG9w0BCQEWIGFsZXhhbmRlci5nYWxpbG92QG1pY3JvZm9jdXMu
Y29tMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA5ZjKCY2x2ruYkW8e
/IgOa5y9xqSx4bUogYuZnAwLgZH2EIEx54T1YzKKc6a58t9tFU0Xb1Z47ay57g/B
A1oOOV4HOsl6SRG4lJojiOKSpLb1zZMqj3s1dd9hLE9KuScchApcJ5F8GxPf6YHO
VpY4d6e6Z+fS071lK3UHpjbLQ71yoDV+s+wJ+pmgsLxiyV/7A+CurxixibyXKx2x
jHvynZBPWf1P/goi54gbCZ1PjQnRPKfxUzRvWipH8T2xvfT0UAZL3HO8C6JJGZxQ
```

```
t82lw/za9tADH0CxPolL/JJyHeEGJAj07uw1wks6mEv8wZY5KkhuDpVv6BUl146+
tL5LSQIDAQABo1AwTjAdBgNVHQ4EFgQUoeHvvSDZn/GIul8Q6T0yleN9q48wHwYD
VR0jBBgwFoAUoeHvvSDZn/GIul8Q6T0yleN9q48wDAYDVR0TBAUwAwEB/zANBgkq
hkiG9w0BAQsFAAOCAQEAQ+T4XForCi/FFSpNLVxb7x/yO1eBi7JujH7CfNTKXUC3
STlTZiJaTLVXzNd9dvxSjzAoDy4NVV/T4KiA4ss7JCTPwGrD3S8k/a+GpogRzRcE
R1i/Z/bx2I4PmQk1g1z4lpuqnic0aIg/OVAE0+kwDBK3E0/pgpoSixAAvxEqM5tw
X9vdt3W/QCoAO3rFABRDboaLkslGbk80Q37tEASKFYm4/0fyB3PEv2uL0S6rP/+E
Fp1Xhlk/5MVRHNb0hLqpZmJxne96dnXpo+ZDeCCn87B3257eRFI1eUeAnxuw79vv
uterPobGSjjPm+y7sY2U3hLKsoVymRvqAohrd9kXSQ==
</ds:X509Certificate>
</ds:X509Data>
</ds:KeyInfo>
</md:KeyDescriptor>
<md:SingleLogoutService
Binding="urn:oasis:names:tc:SAML2.0:bindings:HTTP-POST"
Location="https://www.d18r14.tk/webauth/logout"
ResponseLocation="https://www.d18r14.tk/webauth/logout"/>
<md:AssertionConsumerService
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://www.d18r14.tk/webauth/callback" index="0"/>
</md:SPSSODescriptor>
</md:EntityDescriptor>
```

2. In the ADFS Management console, click **Relying Party Trusts > Add relying party trust**.
3. In the **Add Relying Party Trust wizard**, click **Start**.
4. Select **Import data about the relying party from a file**.
5. Click **Browse** to upload the Advanced Authentication's metadata file that you created in **Step 1**.
6. Click **Next**.
7. Specify the **Display name**.
8. Click **Next**.
9. Ensure that **Open the Edit Claim Rules dialog for this relying party trust when the wizard closes** is selected.
10. Click **Close**.

    The **Edit Claim Rules wizard** is displayed.
11. Click **Add Rule**.
12. Select **Transform an Incoming Claim** from **Claim rule template**.
13. Click **Next**.
14. Specify the **Claim rule name.**
15. Set **Incoming claim type** to **Windows account name**.
16. Set **Outgoing claim type** to **Name ID** and **Outgoing name ID format** to **Windows Qualified Domain Name**.
17. Ensure that **Pass through all claim values** is selected.
18. Click **Finish**.
19. Click **OK**.
20. In the ADFS Management console, click **Relying Party Trusts** and select the relying party trust you added.

21 Right-click on the relying party trust and select **Properties** from the menu.

22 In **Properties**, click the **Encryption** tab and remove the certificate by clicking **Remove**.

23 Click **OK**.

NOTE: Web authentication method does not support the encrypted tokens.

7.33.2 OpenID connect for Advanced Authentication

To add the Open ID Connect Identity Provider, perform the following steps:

1 Specify the name of the provider in **Provider name**.

2 Select the **Available presets**.

The **Issuer, Scope**, and **Key field** are automatically populated.

3 Specify the **Client ID** and **Client secret**.

The **Client ID** and **Client secret** can be obtained by registering with the respective Identity Provider that you select, for more information see Integrating third party Applications with Advanced Authentication using openid connect.

NOTE: Set the Callback URL at the respective Identity Provider. For example, `https://<aahostname>/webauth/callback`.

4 Turn **Send Client secret as an URL parameter** to **ON** to send the Client secret as a URL. By default, the option is set to **OFF**.

5 Click the save icon.

6 Click **Save** to save the method configuration.

Integrating third party Applications with Advanced Authentication using openid connect

The following sample configurations explains how to configure third party applications with Advanced Authentication using OpenID Connect.

- "Integrating Advanced Authentication with Facebook" on page 163
- "Integrating Advanced Authentication with Google" on page 164
- "Integrating Advanced Authentication with yahoo" on page 165
- "Integrating Advanced Authentication with Microsoft Azure" on page 165

Integrating Advanced Authentication with Facebook

Perform the following steps to integrate Advanced Authentication with Facebook using OpenID Connect:

1 Login to facebook for developers (https://developer.facebook.com).

2 Click **My Apps**.

3 In the left pane, click **Settings > Basic**.

4 Make a note of **App ID** and **App Secret**. These are the Client ID and Client Secret for Advanced Authentication.

5 In **Display Name**, specify `Advanced Authentication`. This is the name for this OpenID Connect configuration.

6 In **App Domains**, specify the domain name of the Advanced Authentication Server. For example `aafapp.demo.live`.

7 In **Privacy Policy URL**, specify the URL of the Advanced Authentication Server. For example `aafapp.demo.live`.

8 Scroll through the page until you find the **Website** section. If you cannot find the **Website** section, click **Add Platform > Website**.

9 In the **Website** section, specify the web address of the Advanced Authentication Server. For example `aafapp.demo.live`.

10 Click **Save Changes**.

11 In the left pane, click **Settings > Advanced**.

12 Scroll through the page until you find the **Domain Manager** tab.

13 Click **Add a Domain**.

14 In the Add a Domain window, specify the URL of the Advanced Authentication Server in **Site URL**. For example `aafapp.demo.live`.

15 Click **Apply**.

16 Click **Save Changes**.

17 In the left pane, click **App Review.**

18 Make your application public by clicking the toggle switch in the **Make Advanced Authentication public?** section.

19 In the left pane, below the **Products** tab, click Settings.

20 In **Valid OAuth Redirect URIs**, specify `https://<Advanced Authentication Server>/webauth/callback`.

21 Click **Save Changes**.

22 Specify the Client ID and Client Secret generated in Step 4 on page 164 in the **Client ID** and **Client Secret** fields of Advanced Authentication Administrative Portal.

Integrating Advanced Authentication with Google

Perform the following steps to integrate Advanced Authentication with Google using OpenID connect:

1 Login to Google APIs (https://console.developers.google.com/apis/credentials).

2 Click **Credentials > Create**.

3 Specify a **Project Name** and a **Location**.

4 Click **Create**.

5 Click **Create credentials > OAuth client ID**.

6 Click **Configure a consent screen**.

7 Specify a name in the **Application name** field. For example Advanced Authentication.

8 In **Authorised domains**, specify the domain name of the Advanced Authentication Server. For example `aafapp.demo.live`.

9 In **Application Homepage link**, specify the web address of the Advanced Authentication Server. For example `https://aafapp.demo.live`.

10 In **Application Privacy Policy link**, specify the web address of the Advanced Authentication Server. For example `https://aafapp.demo.live`.

11 In **Application type**, select **Web application**.

12 In **Application Terms of Service link**, specify the web address of the Advanced Authentication Server. For example `https://aafapp.demo.live`.

13 In **Name**, specify a name for the OpenID Connect configuration.

14 In **Authorized JavaScript origins**, specify the Advanced Authentication server address. Ensure that you specify the complete server address including `https`. For example `https://aafapp.demo.live`.

15 In **Authorized redirect URIs**, specify `https://<Advanced Authentication Server>/webauth/callback`. Ensure that you specify the valid Advanced Authentication server name inside <>.

16 Click **Save**.

17 Make a note of the client ID and client secret specified in the **OAuth client** window. Click **OK**.

18 Specify the Client ID and Client Secret generated in Step 17 on page 165 in the **Client ID** and **Client Secret** fields of Advanced Authentication Administrative Portal.

Integrating Advanced Authentication with yahoo

Perform the following steps to integrate Advanced Authentication with Yahoo using OpenID connect:

1 Login to Yahoo Developer Network (https://developer.yahoo.com/apps/).

2 Click **Create an app**.

3 In **Application Name**, specify a name for the OpenID Connect configuration.

4 In **Application Type**, select **Web Application**.

5 In **Callback Domain**, specify the domain name of the Advanced Authentication Server. For example `aafapp.demo.live`.

6 Click **Create**.

7 Make a note of the client ID and client secret. Click **Update**.

8 Specify the Client ID and Client Secret generated in Step 7 on page 165 in the **Client ID** and **Client Secret** fields of Advanced Authentication Administrative Portal.

Integrating Advanced Authentication with Microsoft Azure

Perform the following steps to integrate Advanced Authentication with Microsoft Azure using OpenID connect:

1 Login to Microsoft Azure (https://portal.azure.com/).

2 In the left pane, click **Azure Active Directory**.

3 In the **Manage** section, click **App registrations**.

4 Click **New application registration**.

5 In **Name**, specify a name for the OpenID Connect configuration.

6 In **Application Type**, select **Web app / API**.

7 In **Sign-on URL**, specify `https://<Advanced Authentication Server>/webauth/callback`. Ensure that you specify the correct Advanced Authentication server address inside <>.

8 Click **Create**.

9 Make a note of **Application ID**. It is the Client ID for Advanced Authentication.

10 Click **Settings > Keys**.

11 In the **Passwords** section, specify key description and key duration.

12 Click **Save**.

13 Make a note of the text generated in the **VALUE** field. It is the Client Secret for Advanced Authentication.

14 In the left pane, click **Azure Active Directory**.

15 Click **Properties**.

16 Make a note of the text specified in the **Directory ID** field.

17 Specify the text generated in Step 16 on page 166 in the **Issuer field** of Advanced Authentication Administrative Portal.

18 Specify the Client ID generated in Step 9 on page 166 and Client Secret generated in Step 13 on page 166 in the **Client ID** and **Client Secret** fields of Advanced Authentication Administrative Portal.

7.33.3 OAuth 2.0 for Advanced Authentication

To add the OAuth 2.0 Identity Provider, perform the following steps:

1 Specify the name of the provider in **Provider name**.

2 Select the **Available presets**.

The **Authorization endpoint**, **Token endpoint**, **Attributes endpoint**, **Scope**, and **Key field** are automatically populated.

3 Specify the **Client ID** and **Client secret**.

The **Client ID** and **Client secret** can be obtained by registering with the respective Identity Provider that you select.

NOTE: Set the Callback URL at the respective Identity Provider. For example, `https://<aahostname>/webauth/callback`.

---

4 Turn **Send Client secret as an URL parameter** to **ON** to send the Client secret as a URL. By default, the option is set to **OFF**.

5 Select the format of the access token from **Access token is returned in body encoded as**.

6 Set **Send access token in "Authorization: Bearer" header** to **ON** to send the access token as a header. By default, the option is set to **OFF**.

7 Click the save  icon.

8 Click **Save** to save the method configuration.

7.34 Windows Hello

Windows Hello authentication allows the users to use the Windows Hello Fingerprint and Facial Recognition authentication to log in to Windows 10. Advanced Authentication supports the Windows Hello fingerprint and facial recognition authentication.

To configure Windows Hello method in Advanced Authentication, perform the following steps:

1. Click **Methods > Windows Hello**.
2. (Optional) Set **Allow to specify Username (for AD Users only)** to **ON** if you want the Active Directory users to specify their account name while enrolling. By default, the option is disabled.

   This is applicable for Active Directory users only. This option does not affect local and other repository users and they must specify their account name while enrolling.
3. Click **Save**.

8 Creating a chain

A chain is a combination of authentication methods. A user must pass all methods in the chain to successfully authenticate. 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 provides the following chains by default:

- **LDAP Password Only**: Any user from a repository can use this chain to get authenticated with the LDAP Password (single-factor) method.
- **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 enhanced security, 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 smart card, 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 assign the chain to a 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. Click **Chains > New Chain**.
2. Specify a name of the chain in **Name**.

   NOTE: It is recommended not to use special characters (for example, +, & and so on) in the chain name. This is to avoid issues in the OAuth 2.0 and SAML 2.0 events.

3. Set **Is enabled** to **ON** to enable the chain.
4. Select the methods that you want to add to the chain from the **Available** list of **Methods**.

   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.

   You can reorder or remove methods in the chain using the up and down arrows, as well as the remove button next to the **Used** list.

NOTE: The Events using this chain displays all events that are associated with that specific chain. Events list cannot be modified manually.

5 Specify the groups that will use the authentication chain in **Roles and Groups**.

You can specify the following roles and groups based on your requirement:

- **ALL USERS**: Applicable for all users and groups of all added repositories.
- **<REPO\Group>**: Applicable for a specific group from the repository. For example, to specify users of an **IT staff** group, specify **FOCUS\IT staff**.
- **<REPO Users>**: Applicable for all users of a specific repository. For example, to use all users in the repository **FOCUS**, specify **FOCUS Users**.

IMPORTANT: It is recommended to not use those groups from which you cannot exclude users because you will not be able to free up a user's license. For example, you use a **Repo Users** group or **ALL USERS** group. If an employee from these groups leaves the company and you do not delete the user's domain account but disable it, the license will not be freed.

6 Expand **Advanced Settings** by clicking **+** and configure the following settings as required:

6a Set **Apply if used by endpoint owner** to **ON** if an Endpoint owner must use the chain.

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

6b (Conditional) 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 configured the Logon Filter.

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

NOTE: If the user credentials are saved with **Remember my credentials,** the MFA tag does not work while connecting to the Remote Desktop.

6c **Required Identity Assurance level**: This option enables you to employ Identity Proofing. Specify the identity assurance level required to authenticate using the chain. By default, the value is set to 0 indicating users without any identity assurance level can use this chain for authentication.

For example, if you want to restrict the use of the Password + SMS OTP chain for users with identity assurance level 1 then set the **Required Identity Assurance** level to 1. So, users who are granted with specify assurance level can use the chain for authentication.

6d **Granted Identity Assurance level**: This option enables you to employ Identity Proofing. Specify the Identity assurance level that is issued to a user after succeeding the authentication chain. By default, the value is set to 0 indicating users who pass this chain will be grated 0 or no identity assurance level.

This assurance level of a user is considered during the subsequent authentication attempt to display the authentication chain with equivalent **Required Identity Assurance level**.

For example, if the **Granted Identity assurance level** is set to 2 for a chain with the LDAP Password method then the identity assurance level 2 is granted for users who authenticate with that chain.

6e (Conditional) Set **Required chain** to **Nothing** if this is a regular required (high-security) chain with no other chain linked to it. To configure a linked chain, create a simple chain that includes a single method, then under Advanced Settings select a Required chain (with multifactor) and specify **Grace period (mins)**. Within the grace period, the simple linked chain can be used instead of the required chain. The maximum value for grace period is 44640 minutes (31 days).

For example, create a chain `Card` as the linked chain and specify `LDAP Password+Card` as the Required Chain. Set the grace period to 480. Users must use `LDAP Password+Card` chain for initial login. However, for the next eight hours post login will be able to authenticate with just Card without the LDAP Password.

IMPORTANT: The **Required chain** option is available when Linked chains is set to ON in the Linked chains policy. You must assign both a required and a linked chain to an Event.

---

7 (Conditional) Expand **Risk Settings** by clicking **+** and select a risk level in **Minimum Risk Level**.

A user can use this chain for completing authentication if the risk associated with the login attempt matches or above the selected value.

For example, you have selected `Low`. This chain will be shown to the user if the risk level of that login attempt is low, medium, or high.If you have selected `Medium`, the chain will be shown to the user when the risk level of the login attempt is medium or high.

IMPORTANT: This option is available when you enable Risk Settings. For more information, see Part III, “Configuring risk settings,” on page 265.

The following scenarios describe which chains are displayed if a rule is set as the decisive rule with a specific action:

- When a rule is set as the decisive rule with action, **Allow Access** and if the rule succeeds, the risk level is calculated as Low. User is shown with all chains (Low, Medium, and High) for authentication.
- When a rule is set as the decisive rule with action, **Deny Access** and if the rule fails, the risk level is calculated as High. User is denied access and a message `Access has been denied` is displayed without the chain selection.

8 (Conditional) In **Custom names**, you can specify the chain name in a specific language. To do this click **+** to expand the settings and specify the chain name.

9 Click **Save**.

IMPORTANT: If you have configured more than one chain using one method (for example, **Smartphone + LDAP Password**, **LDAP Password**) and assigned to the same group of users and the same event, then the top chain (first chain in the list) is always used if the user has enrolled all methods in the chain.

---

9 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 an application or a 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 scenarios:

- Third-party integrations.
- To use Windows Client, Linux PAM Client, or Mac OS X Client on both domain joined and non-domain workstations. It requires a separate event to use the non-domain mode.
- Integrations using SAML 2.0 and OAUTH 2.0.
- To create more than one RADIUS Server event.

This section discusses the following topics:

- Section 9.1, “Configuring an existing event,” on page 173
- Section 9.2, “Creating a customized event,” on page 183

9.1 Configuring an existing event

Following are the generic steps to configure an existing event:

1. Click **Events > New Event**.
2. Specify a name of the event in **Name**.
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**.

   This option is only applicable for **Windows logon**, **Linux logon,** and **Mac OS logon** events. You can change the **Event type** from **OS Logon (domain)** to **OS Logon (local)** if the workstations are not joined to the domain.

   - Select OS Logon (domain) to allow only the domain joined users to login to the event.
   - Select OS Logon (local) to allow any Advanced Authentication user from any repository to access the event. However, users must map themselves to a local user account during their first login by providing the credentials.
5. Enable the **reCAPTCHA** option to **ON** if you want the Google reCAPTCHA option to be displayed in the login page for the particular event.

   The reCAPTCHA option is displayed only when you enable the Google reCAPTCHA Options policy.

NOTE: The reCAPTCHA option is supported only for the **Admin UI** event, **Authenticators Management** event, **Helpdesk** event, **Helpdesk user** event, **Report logon** event, **Tokens Management** event, and the **Search Card** event.

6 By default, **All Categories** is set to **ON**. When the multiple event categories are created, users can enroll an authentication method multiple times (one enrolled method per category).

When **All Categories** is set to ON, users can authenticate to the event using any of the supported methods (Card, FIDO U2F, HOTP, Password, and TOTP) and Advanced Authentication automatically chooses an appropriate authentication method.

To use other methods, Advanced Authentication prompts for the category selection.

The **All Categories** option is displayed only if you have added categories in the "Event categories" policy.

For example, an administrator has configured two categories CAT1 and CAT2. The **Default** category is predefined in the Administration portal. Users can enroll three devices. The **All Categories** is set to **ON** for the Windows logon event. A user has three cards and enrolls each to a category as follows:

- Card 1 to Default
- Card 2 to CAT1
- Card 3 to CAT2

After enrolling cards, the user can authenticate to the Windows event by using one of the enrolled cards.

You can set **All Categories** to **OFF** if you want to disable support for multi-enrollment of supported methods.

The **Authenticator category** is displayed when **All Categories** is set to **OFF**. Select the preferred category from **Authenticator category**.

7 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.

The top chain (first chain) in the list of selected chains will be considered as a high-priority or high-security chain for that specific event.

You can reorder or remove methods in the chain using the up and down arrows, as well as the remove button next to the **Used** list.

8 (Conditional) In **Risk Policy**, select the policy that you want to assign to this event for assessing the risk associated with a login attempt.

9 (Conditional) Click **Create New Policy** to create a new risk policy for this event.

Clicking this option opens the Risk Settings page.

IMPORTANT: **Risk Policy** and **Create New Policy** options are available when you enable Risk Settings. For more information, see Part III, "Configuring risk settings," on page 265.

10 (Conditional) Set the **Enable chain selection** with one of the following options on your requirements:

- **ON**: Select this option to allow users to select their preferred authentication chain from all the chains that are available to them. By default, this option is set to **ON**.

- **OFF**: Select this option to force users to use the chain that has the highest priority for authentication.
- **OPTIONAL**: Select this option to display the high-priority chain with the ability to select the other chains from the list. If the user doesn't wish to continue with the highest priority chain, they can click the **Select Chain** button to select their preferred chain from all the chains that are available to them.

NOTE: This option is available only for the **Authenticators Management** (New Enrollment UI), **OOB UI Logon**, and **Smartphone Enrollment**. It is also available for **OAuth2 / OpenID Connect** and **SAML2** events.

---

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

IMPORTANT: Endpoints whitelist supports only the Windows Logon, Linux Logon, and Mac OS Logon events.

---

12 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.

---

13 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.

14 Set **Logon with Expired Password** with one of the following options based on your requirement:

- **Allow**: Select this option to allow users to log in to the event with the expired LDAP password.
- **Ask to change**: If the password has expired this option prompts users to change the password during logon. Change in the LDAP Password is supported only for the Active Directory repositories. However, the LDAP Password change in Advanced Authentication is not allowed when the LDAP Servers in the Repository settings are configured with port 389. The LDAP server rejects the new password.
- **Deny**: Select this option to deny access to the event with the expired LDAP password. When the access is denied, the following message is displayed to users:

  ```
  You must change your password to logon.
  ```

15 Set **Bypass user lockout in repository** to **ON**, users, who are locked in the repository, must select and authenticate with the chain that does not include the LDAP Password method. By default, **Bypass user lockout in repository** is set to **OFF** and locked users cannot authenticate by using any chain.

To use this functionality, it is required to have more than one chain without the LDAP Password method assigned to the event. This is to provide more options to users.

NOTE: All authentication chains irrespective of whether it includes the LDAP Password method or not are displayed to users who are locked in the repository.

---

16 Set **Allow token re-use** to **ON**, if you want to allow users to apply the OTP multiple times within the **Allow re-sending after (seconds)** duration for authentication. This option is applicable for Email OTP, SMS OTP, and Voice OTP methods.

By default, **Allow token re-use** is set to **OFF** and users are not allowed to apply the OTP more than once within the **Allow re-sending after (seconds)** duration that has been set for Email OTP, SMS OTP, and Voice OTP methods.

17 Set **Return groups on logon** to **ON** if you want to retrieve the group details of users who authenticated to the event in the authentication response.

With **Return groups on logon** set to **ON**, if **Groups** is empty, all the groups that the users are associated with are returned in the response. However, to return the required groups, specify the preferred groups in **Groups**.

Sometimes, the authentication response of RADIUS event is lengthy if a user is associated with several groups. Therefore, it is recommended to use **Groups** to limit the groups' in the response.

By default, **Return groups on logon** is set to **ON** for all events except for Authenticators Management, Smartphone Enrollment, and SAML 2.0 events.

When this option is set to **OFF**, the groups of users authenticated to the event are not returned in the response.

18 Select the **Allow to logon to this event by shared authenticator** option to allow users to login using shared authenticators. By default this option is disabled for the `Authenticators Management`, `Helpdesk`, `Helpdesk User`, `AdminUI`, `Search Card`, `Token Management`, and `Report Logon` events and enabled for all the other events.

NOTE: The **Allow to logon to this event by shared authenticator** option is displayed if you enable the **Enable sharing of authenticators** option in Authenticator management options policy.

---

19 Click **Save**.

20 Click **Initialize default chains** to revert the changes that are applied to the default configuration.

NOTE: If you have configured more than one chain using one method (for example, **Smartphone + LDAP Password**, **LDAP Password**) and assigned to the same group of users and the same event, then the top chain (first chain in the list) is always used if the user has enrolled all methods in the 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.

By default, Advanced Authentication contains the following events:

- Section 9.1.1, “ADFS event,” on page 177
- Section 9.1.2, “AdminUI event,” on page 177
- Section 9.1.3, “Authentication agent event,” on page 178
- Section 9.1.4, “Authenticators management event,” on page 178
- Section 9.1.5, “Desktop OTP tool event,” on page 179
- Section 9.1.6, “Helpdesk event,” on page 179

- Section 9.1.7, “Helpdesk user event,” on page 179
- Section 9.1.8, “Linux Logon event,” on page 180
- Section 9.1.9, “Mac OS Logon event,” on page 180
- Section 9.1.10, “Mainframe Logon event,” on page 180
- Section 9.1.11, “NAM event,” on page 180
- Section 9.1.12, “NCA event,” on page 181
- Section 9.1.13, “OOB UI Logon event,” on page 181
- Section 9.1.14, “RADIUS server event,” on page 181
- Section 9.1.15, “Report Logon event,” on page 181
- Section 9.1.16, “SCIM API event,” on page 181
- Section 9.1.17, “Search card event,” on page 182
- Section 9.1.18, “Smartphone enrollment event,” on page 182
- Section 9.1.19, “Tokens management event,” on page 182
- Section 9.1.20, “Windows Logon event,” on page 182

9.1.1 ADFS event

This event is used to integrate Advanced Authentication with ADFS using the previous ADFS plug-in for Advanced Authentication 5.x.

For 6.0, you can use the new ADFS MFA plug-in. For more information see the *Configuring the Advanced Authentication server for ADFS Plug-in* guide.

For more information about **ADFS** event configuration options, refer the Generic Options detailed in Configuring an existing event.

9.1.2 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.

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

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: If you have enabled the Google reCAPTCHA policy for the Admin UI event, you must consider the following guidelines. Otherwise, a deadlock scenario might happen and you cannot access the Administration portal without the cluster re-installation:

- If the site key or secret key gets deleted at the Google server, you will not be able to get the same site key or secret key. The site key and secret key used on the Administration portal are no more valid and there is no way to bypass the reCaptcha on the Administration portal.
- If you have registered the reCAPTCHA for one domain name and you change the domain name or migrate the Advanced Authentication server to another domain name, the site key or secret key used on the Administration portal are no more valid.

For more information about **AdminUI** event configuration options, refer the Generic Options detailed in Configuring an existing event.

9.1.3 Authentication agent event

Configure the settings of this event to enable a login to the Authentication Agent for Windows in Advanced Authentication 6.3 SP4 and prior versions.

From Advanced Authentication 6.3 SP5, the OOB UI Logon event is used instead of this event.

For more information about **Authentication Agent** event configuration options, refer the Generic Options detailed in Configuring an existing event.

9.1.4 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.

For more information about **Authenticators Management** event configuration options, refer the Generic Options detailed in Configuring an existing event.

In addition to the generic steps, perform the below event specific configuration:

1 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.

2 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, 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`.

3 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**. This option is applicable only for the Old Enrollment UI.

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 "Managing authenticators" in the *Advanced Authentication- user* guide.

4 Click **Save**.

9.1.5 Desktop OTP tool event

Use this event to enroll the TOTP method using the OTP Desktop tool. This event supports a chain with either LDAP Password or Password method as a single factor authenticator.

For more information about **OTP Desktop Tool** event configuration options, refer the Generic Options detailed in Configuring an existing event.

9.1.6 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.

For more information about **Helpdesk** event configuration options, refer the Generic Options detailed in Configuring an existing event.

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 "Managing Authenticators" in the *Advanced Authentication- Helpdesk Administrator* guide.

9.1.7 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.

NOTE: You must enable the **Ask credentials of management user** option in the Helpdesk Options policy before using this event.

For more information about **Helpdesk User** event configuration options, refer the Generic Options detailed in Configuring an existing event.

9.1.8 Linux Logon event

Configure the settings of this event to enable login to the Linux Client.

NOTE: If you want to use Linux Client on non-domain joined workstations, change the **Event type** from **OS Logon (domain)** to **OS Logon (local)**.

For more information about **Linux Logon** event configuration options, refer the Generic Options detailed in Configuring an existing event.

9.1.9 Mac OS Logon event

Configure the settings of this event to enable login to the Mac OS Client.

NOTE: 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)**.

For more information about **Mac OS Logon** event configuration options, refer the Generic Options detailed in Configuring an existing event.

9.1.10 Mainframe Logon event

Configure the settings of this event to enable login to the Mainframe system.

Example of Mainframe logon event is Advanced Authentication Connector (https://www.microfocus.com/documentation/advanced-authentication-connector/).

For more information about **Mainframe Logon** event configuration options, refer the Generic Options detailed in Configuring an existing event.

9.1.11 NAM event

Configure the settings of this event to facilitate the integration of Advanced Authentication with OpenText Access Manager (https://www.netiq.com/products/access-manager/).

For more information about **NAM** event configuration options, refer the Generic Options detailed in Configuring an existing event.

9.1.12 NCA event

Configure the settings of this event to facilitate the integration of Advanced Authentication with OpenText CloudAccess (https://www.netiq.com/products/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 CloudAccess (https://www.netiq.com/documentation/cloudaccess-3-0/) documentation.

For more information about **NCA** event configuration options, refer the Generic Options detailed in Configuring an existing event.

9.1.13 OOB UI Logon event

Configure this event to log in to the Advanced Authentication OOB portal, Authentication Agent for Windows, and Authentication Agent for Web. These components enable users to manage the authentication requests of the Out-of-band method to authenticate to a specific event for which a chain with the **Out-of-band** method is assigned.

NOTE: You must not assign a chain containing the **Out-of-band** method to the **OOB UI logon** event.

For more information about **OOB UI Logon** event configuration options, refer the Generic Options detailed in Configuring an existing event.

9.1.14 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. For more information about configuring the RADIUS Server event, see “RADIUS server”.

For more information about **RADIUS Server** event configuration options, refer the Generic Options detailed in Configuring an existing event.

9.1.15 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 “Reporting”.

For more information about **Report Logon** event configuration options, refer the Generic Options detailed in Configuring an existing event.

9.1.16 SCIM API event

IMPORTANT: The SCIM (System for Cross-domain Identity Management) API event is applicable only for Advanced Authentication as a Service (SaaS).

Configure the settings of this event to access the SCIM API. This event enables administrators to handle the SCIM operations for the cloud deployments through SCIM API. SCIM operations include tasks related to user identity and access management, such as creating, updating, or deleting user accounts, as well as managing user groups.

For more information about the SCIM API calls, see SCIM (https://www.netiq.com/documentation/advanced-authentication-64/apidoc/data/apidoc.html#scim).

For additional information about **SCIM API** event configuration options, refer the Generic Options detailed in Configuring an existing event.

9.1.17 Search card event

Configure the settings of this event to log in to the Advanced Authentication Search Card portal. The Search Card functionality helps you to get the card holder's contact information by inserting the card in the card reader. For more information about searching a card holder's information, see "Searching a card holder's information".

For more information about **Search Card** event configuration options, refer the Generic Options detailed in Configuring an existing event.

9.1.18 Smartphone enrollment event

The Smartphone method can be enrolled in two ways:

- By scanning a QR code that is shown in the Self-Service Portal.
- By using an enrollment link that can be manually sent through SMS or Email.

This event allows managing enrollment using the enrollment link. For more information about preparing the enrollment link, see Configuring enrollment link.

This event supports a chain with either LDAP Password or the Password method as a single factor authenticator.

To enroll the Smartphone method using an enrollment link, users are required to click the link on their smartphone with the OpenText Advanced Authentication application installed, then specify their user name and password. The users of LDAP repositories can use the LDAP password, the local users and users of other repo (for example, SQL repo) who do not have an LDAP password can use their enrolled password to enroll in the Smartphone method by link. If the app is not installed on the user's smartphone, the user will be prompted to install the app. After entering the credentials the authenticator is enrolled automatically and is ready to use.

For more information about **Smartphone Enrollment** event configuration options, refer the Generic Options detailed in Configuring an existing event.

9.1.19 Tokens management event

Configure the settings of this event to log in to the Advanced Authentication Tokens Management portal. The Tokens Management functionality allows you to assign each token to specific user. For more information about assigning a token to user, see Managing Tokens.

For more information about **Token Management** event configuration options, refer the Generic Options detailed in Configuring an existing event.

9.1.20 Windows Logon event

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

NOTE: If you want to use Windows Client on non-domain joined workstations, change the **Event type** from **OS Logon (domain)** to **OS Logon (local)**.

For more information about **Windows Logon** event configuration options, refer the Generic Options detailed in Configuring an existing event.

9.2 Creating a customized event

You can create customized events in the following scenarios:

- 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.
- To create more than one RADIUS Server event.

You can create the following types of customized events:

- Generic
- OS Logon (domain)
- OAuth2 / OpenID Connect
- SAML2
- RADIUS

9.2.1 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 > New Event**.
2. Specify a name for the event.
3. Set **Is enabled** to **ON**.

   By default **Generic** is set in **Event Type**.
4. Select the **Authenticator category.** The **Authenticator category** option is displayed only if you have added categories in the "Event categories" policy.
5. Select the chains that you want to assign to the current event.
6. (Conditional) In **Risk Policy**, select the policy that you want to assign to this event for assessing the risk associated with a login attempt.
7. (Conditional) Click **Create New Policy** to create a new risk policy for this event.

   Clicking this option opens the Risk Settings page.

   IMPORTANT: **Risk Policy** and **Create New Policy** options are available when you enable Risk Settings. For more information, see Part III, "Configuring risk settings," on page 265.

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

IMPORTANT: Endpoints whitelist supports only Windows Logon, Linux Logon, and Mac OS Logon events.

---

9 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.

---

10 Set **Logon with Expired Password** with one of the following options based on your requirement:

- **Allow**: Select this option to allow users to log in to the event with the expired LDAP password.
- **Ask to change**: If the password has expired this option prompts users to change the password during logon. Change in the LDAP Password is supported only for the Active Directory repositories. However, the LDAP Password change in Advanced Authentication is not allowed when the LDAP Servers in the Repository settings are configured with port 389. The LDAP server rejects the new password.
- **Deny**: Select this option to deny access to the event with the expired LDAP password. When the access is denied, the following message is displayed to users:

  `You must change your password to logon.`

11 Set **Bypass user lockout in repository** to **ON**, users, who are locked in the repository, must select and authenticate with the chain that does not include the LDAP Password method. By default, **Bypass user lockout in repository** is set to **OFF** and locked users cannot authenticate by using any chain.

To use this functionality, it is required to have more than one chain without the LDAP Password method assigned to the event. This is to provide more options to users.

NOTE: All authentication chains irrespective of whether it includes the LDAP Password method or not are displayed to users who are locked in the repository.

---

12 Set **Return groups on logon** to **ON** if you want to retrieve the group details of users who authenticated to the event in the authentication response.

With **Return groups on logon** set to **ON**, if **Groups** is empty, all the groups that the users are associated with are returned in the response. However, to return the required groups, specify the preferred groups in **Groups**.

By default, **Return groups on logon** is set to **OFF**, the groups of users authenticated to the event are not returned in the response.

13 Select the **Allow to logon to this event by shared authenticator** option to allow users to login using shared authenticators. By default this option is disabled for the `Authenticators Management`, `Helpdesk`, `Helpdesk User`, `AdminUI`, `Search Card`, `Token Management`, and `Report Logon` events and enabled for all the other events.

NOTE: The **Allow to logon to this event by shared authenticator** option is displayed if you enable the **Enable sharing of authenticators** option in Authenticator management options policy.

14 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.

9.2.2 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.

9.2.3 Creating an OAuth 2.0 / openid connect event

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

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

1 Click **Events > New Event**.

2 Specify a name for the event.

3 Set **Is enabled** to **ON**.

4 Select **OAuth2 / OpenID Connect** 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.

The top chain (first chain) in the list of selected chains will be considered as a high-priority or high-security chain for that specific event.

7 Set the **Enable chain selection** with one of the following options on your requirements:

- **ON**: Select this option to allow users to select their preferred authentication chain from all the chains that are available to them. By default, this option is set to **ON**.
- **OFF**: Select this option to force users to use the chain that has the highest priority for authentication.
- **OPTIONAL**: Select this option to display the high-priority chain with the ability to select the other chains from the list. If the user doesn't wish to continue with the highest priority chain, they can click the **Select Chain** button to select their preferred chain from all the chains that are available to them.

8 (Conditional) In **Risk Policy**, select the policy that you want to assign to this event for assessing the risk associated with a login attempt.

9 (Conditional) Click **Create New Policy** to create a new risk policy for this event.

Clicking this option opens the Risk Settings page.

IMPORTANT: **Risk Policy** and **Create New Policy** options are available when you enable Risk Settings. For more information, see Part III, “Configuring risk settings,” on page 265.

10 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.

NOTE: You cannot view the **Client secret** after saving the event. Later, you can reset the **Client secret** if you need.

NOTE: Resetting the **Client secret** will disrupt the service that relies on the event. To resume the service, you need to share the new client secret in the consumer web application and authenticate.

11 In **Advanced Settings**, perform the following actions:

NOTE: The options for **Advanced Settings** are hidden. To view all options click **+** icon.

- Set the **Enable Public Client option** to **ON** to enable the public clients. By default, **Enable Public Client option** is set to **OFF**.
- Set the **Support Authorization Code** to **ON** to enable the event to support the authorization code. By default, **Support Authorization Code** is set to **OFF**
- Enabling the **Use for Resource Owner Password Credentials** setting will enable the event with the ability to use the Resource Owner Password Credentials grant in order to get access tokens as outlined by the OAuth 2.0 specifications (https://datatracker.ietf.org/doc/html/rfc6749#section-1.3.3). By default, **Use for Resource Owner Password Credentials** is set to **OFF**
- Set the **Support Client Credentials** to **ON** to enable the event to support the client credentials. By default, **Support Client Credentials** is set to **OFF**.
- Set the **Support Implicit** to **ON** to enable the event to support Implicit. By default, **Support Implicit** is set to **OFF**.
- Set the **Enable Token Revocation** to **ON** to enable the vent to revoke the token. By default, **Enable Token Revocation** is set to **OFF**.
- Set the **Enable Session Token Revocation** to **ON** to enable the event to revoke the session token. By default, **Enable Session Token Revocation** is set to **OFF**.
- Set the **Enable Token Sharing** to **ON** to enable the event to share the token. By default, **Enable Token Sharing** is set to **OFF**.
- Set the **Enable OpenID Connect** to **ON** to enable the Open ID connect. by default, **Enable OpenID Connect** is set to **OFF**.
- Set the **Enable all Claims in ID token** to On to enable all the claims in ID token. By default, **Enable all Claims in ID token** is set to **OFF**.
- Specify the Attribute Maps in **Attribute Maps. One Map per line** field.

  The Attribute maps should be specified in the following format:

  ```
  localName="<local name>" clientName="<client name>"
  ```

For example, `localName="mail" clientName="user_email"`

where,

- `localName`: This value indicates the name of the attribute in the Web Authentication (local) namespace. This is how it is referred in Advanced Authentication. This value can be defined by users.
- `clientName`: This value is the name by which the attribute value appears in JWTs.

- Specify the timeout value in seconds till when the authorization code is valid in **Authorization Code Timeout**. By default, this value is set to 120 seconds. The request for an Access Token or an ID Token fails if the Authorization Code has expired and is no longer valid. The Authorization code becomes invalid if the client does not request for Token ID from the server within the specified time.

  For security reasons, some OAuth2 / OpenID Connect code flow schemes require that first an Authorization Code be requested. The Authorization Code is then used to request an Access Token and ID Token.

  IMPORTANT: Before configuring the timeout value in **Authorization Code Timeout**, **Access Token Timeout**, **Refresh Token Timeout**, **Public Refresh Token Timeout**, or **Session Token Revocation Timeout** in OAuth event, it is required to enable the relevant OAuth grant types under **Advanced Settings**. If the OAuth grant types are not enabled, default timeout values from Web Authentication policy are applicable.

- Specify the time in seconds till when the access token is valid in **Access Token Timeout**. By default, this value is set to 120 seconds. Once the token expires, a new token is required before accessing the protected resources. The application might create a new token by using a Refresh Token and the client secret, or else the user is required to authenticate again.
- Specify the time in seconds till when the token is valid in **Refresh Token Timeout**. Once the token expires it can no longer be used to create a new Access Token. By default, this value is set to 2592000 seconds.
- Specify the timeout value for refreshing token for public clients in **Public Refresh Token Timeout**. This timeout is applicable when there are two client types, private and public. By default, this value is set to 3600 seconds.
- Specify the timeout value till when the session-based refresh token revocation entries are retained in **Session Token Revocation Timeout**. Retained entries are removed when the session is properly logged out or after the refresh token expires. By default, this value is set to 172800 seconds.

  NOTE: If you do not modify the values in **Authorization Code Timeout**, **Access Token Timeout**, **Refresh Token Timeout**, **Public Refresh Token Timeout**, and **Session Token Revocation Timeout**, these settings will contain default values in the Web Authentication Policy.

  ---

12 Set **Logon with Expired Password** with one of the following options based on your requirement:

- **Allow**: Select this option to allow users to log in to the event with the expired LDAP password.

- **Ask to change**: If the password has expired this option prompts users to change the password during logon. Change in the LDAP Password is supported only for the Active Directory repositories. However, the LDAP Password change in Advanced Authentication is not allowed when the LDAP Servers in the Repository settings are configured with port 389. The LDAP server rejects the new password.
- **Deny**: Select this option to deny access to the event with the expired LDAP password. When the access is denied, the following message is displayed to users:

  `You must change your password to logon.`

13 Set **Bypass user lockout in repository** to **ON**, users, who are locked in the repository, must select and authenticate with the chain that does not include the LDAP Password method. By default, **Bypass user lockout in repository** is set to **OFF** and locked users cannot authenticate by using any chain.

   To use this functionality, it is required to have more than one chain without the LDAP Password method assigned to the event. This is to provide more options to users.

   NOTE: All authentication chains irrespective of whether it includes the LDAP Password method or not are displayed to users who are locked in the repository.

14 Set **Allow token re-use** to **ON**, if you want to allow users to apply the OTP multiple times within the **Allow re-sending after (seconds)** duration for authentication. This option is applicable for Email OTP, SMS OTP, and Voice OTP methods.

   By default, **Allow token re-use** is set to **OFF** and users are not allowed to apply the OTP more than once within the **Allow re-sending after (seconds)** duration that has been set for Email OTP, SMS OTP, and Voice OTP methods.

15 Select the **Allow to logon to this event by shared authenticator** option to allow users to login using shared authenticators. By default this option is disabled for the `Authenticators Management`, `Helpdesk`, `Helpdesk User`, `AdminUI`, `Search Card`, `Token Management`, and `Report Logon` events and enabled for all the other events.

   NOTE: The **Allow to logon to this event by shared authenticator** option is displayed if you enable the **Enable sharing of authenticators** option in Authenticator management options policy.

16 Click **Save.**

NOTE: The logout URL must follow the below format:

`https://<AAServer>/osp/a/TOP/auth/app/logout`

where TOP is the name of the tenant.

However, it is possible to perform the logout from both Identity Provider and Service Provider using the following URL:

`https://<AAServer>/osp/a/TOP/auth/app/logout?target=https://<Service Provider>/app/logout`

For example: `https://<AAServer>/osp/a/TOP/auth/app/logout?target=https://<NAMServer>/nidp/app/logout`

After you have created an **OAuth 2 / OpenID Connect** 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 endpoint (authorization endpoint) in the web application.

   For example, `https://<Appliance IP>/osp/a/TOP/auth/oauth2/grant` in the URL, TOP can be replaced by the tenant name.
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.

OAuth events support the step-up authentication. It does not prompt users to authenticate with the same method that the user has succeeded for an event during the session. Let us understand the step-up authentication with an example, assume there are three OAuth events, EVT1, EVT2, and EVT3. Chains associated with each event are as follows:

- EVT1 - LDAP Password + Security Questions
- EVT2 - LDAP Password + SMS OTP
- EVT3 - Security Questions + SMS OTP

Possible scenarios:

- First the user logs in to EVT1 by furnishing LDAP password and valid secret questions. During the same session if the user tries to authenticate to EVT3 then a prompt to provide the SMS OTP is displayed. This happens because user has succeeded Security Questions method for EVT1.
- User logs in to EVT3 first with the Security Questions and SMS OTP. Later, the user can authenticate to EVT2 with the LDAP Password method because SMS OTP is succeeded for the previous event.

NOTE: To bypass the username prompt during the authorization process, you can include the `Ecom_User_ID` parameter to retrieve and pass the username along with the redirect URL as follows:

```
https://<AA_IP>/osp/a/TOP/auth/oauth2/
grant?response_type=code&client_id=<CLIENT_ID>&Ecom_User_ID=<USERNAME_ENTE
RED_IN_CLIENT_SITE>&redirect_uri=<REDIRECT_URL>
```

9.2.4 Creating a SAML 2.0 event

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

To create an **SAML 2.0** event, perform the following steps:

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.

The top chain (first chain) in the list of selected chains will be considered as a high-priority or high-security chain for that specific event.

7 Set the **Enable chain selection** with one of the following options on your requirements:

- **ON**: Select this option to allow users to select their preferred authentication chain from all the chains that are available to them. By default, this option is set to **ON**.
- **OFF**: Select this option to force users to use the chain that has the highest priority for authentication.
- **OPTIONAL**: Select this option to display the high-priority chain with the ability to select the other chains from the list. If the user doesn't wish to continue with the highest priority chain, they can click the **Select Chain** button to select their preferred chain from all the chains that are available to them.

8 (Conditional) In **Risk Policy**, select the policy that you want to assign to this event for assessing the risk associated with a login attempt.

9 (Conditional) Click **Create New Policy** to create a new risk policy for this event.

Clicking this option opens the Risk Settings page.

IMPORTANT: **Risk Policy** and **Create New Policy** options are available when you enable Risk Settings. For more information, see Part III, "Configuring risk settings," on page 265.

---

10 In **SAML 2.0 settings,** perform the following actions:

NOTE: You must configure the Web Authentication policy for the SAML 2.0 event to work appropriately.

10a 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.

10b Select the required option from **NameID formatting options** based on the SAML response requirement of service provider. The available options are:

- **Use default**: To send NameID in SAML response without any customization.
- **Send E-Mail as NameID (suitable for G-Suite)**: To send **email address** in the **NameID** attribute and is required for integrating with the G-suite.
- **Send SAMAccount as NameID**: To send **SAMAccountName** in the **NameID** attribute of SAML response from the Advanced Authentication server.
- **Send CN as NameID**: To send UID of user in the **NameID** attribute of SAML response from the Advanced Authentication server. This is required, when eDirectory is used as the repository and service providers want nameid format as `unspecified` however need Common Name (UID by default) in the SAML response. This is required for integrating with Cyberark.
- **Send ImmutableId (User objectId) as NameID (required for Microsoft Office 365)**: To send **User objectId** in the **NameID** attribute as a SAML response from the Advanced Authentication server. This is required for integrating with Microsoft Office 365.
- **Create Custom NameID**: To send custom details about user, such as Windows domain qualified name, unspecified and so on in the **NameID** attribute as a SAML response.

For sending custom details in NameID attribute, perform the following:

- Select the preferred **NameID Format** to send in the SAML response. The available options are:
  - **urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified**
  - **urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress**
  - **urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName**
  - **urn:oasis:names:tc:SAML:1.1:nameid-format:WindowsDomainQualifiedName**
  - **urn:oasis:names:tc:SAML:2.0:nameid-format:kerberos**

- Specify the attribute that you want as identifier in the SAML response.

  Some attributes that do not need additional configurations are DN, CN, mail, mobile, sid, upn, netbiosname, sAMAccountName, and userImmutableID. Other attributes that are not in the list must be configured in the LDAP repository's Custom attributes to fetch and Custom attributes to return.

.

10c Specify the Attribute Maps in **Attribute Maps. One Map per line**.

The Attribute maps should be specified in the following format:

```
localName="<local name>" samlName="<Service Provider name>"
```

For example,

```
localName="mail" samlName="e-mail address"
```

Here, the service provider expects the `"e-mail address"` instead of `"mail"` from the Identity Provider (in this case, Advanced Authentication).

```
localName="userLastName" samlName="Surname"
```

```
localName="userFirstName" samlName="Given Name"
```

```
localName="mobile" samlName="Telephonenumber"
```

where,

- `localName`: This value indicates the name of the attribute in the Web Authentication (local) namespace. This is how it is referred in Advanced Authentication. This value can be defined by users.
- `samlName`: This value indicates the name of the attribute in SAML assertion.

10d Set **Logon with Expired Password** with one of the following options based on your requirement:

- **Allow**: Select this option to allow users to log in to the event with the expired LDAP password.
- **Ask to change**: If the password has expired this option prompts users to change the password during logon. Change in the LDAP Password is supported only for the Active Directory repositories. However, the LDAP Password change in Advanced Authentication is not allowed when the LDAP Servers in the Repository settings are configured with port 389. The LDAP server rejects the new password.

- **Deny**: Select this option to deny access to the event with the expired LDAP password. When the access is denied, the following message is displayed to users:

  `You must change your password to logon.`

10e Set **Bypass user lockout in repository** to **ON**, users, who are locked in the repository, must select and authenticate with the chain that does not include the LDAP Password method. By default, **Bypass user lockout in repository** is set to **OFF** and locked users cannot authenticate by using any chain.

To use this functionality, it is required to have more than one chain without the LDAP Password method assigned to the event. This is to provide more options to users.

NOTE: All authentication chains irrespective of whether it includes the LDAP Password method or not are displayed to users who are locked in the repository.

10f Set **Allow token re-use** to **ON**, if you want to allow users to apply the OTP multiple times within the **Allow re-sending after (seconds)** duration for authentication. This option is applicable for Email OTP, SMS OTP, and Voice OTP methods.

By default, **Allow token re-use** is set to **OFF** and users are not allowed to apply the OTP more than once within the **Allow re-sending after (seconds)** duration that has been set for Email OTP, SMS OTP, and Voice OTP methods.

10g Set **Return groups on logon** to **ON** to retrieve the group details of users who authenticated to the SAML 2.0 event in the authentication response.

With **Return groups on logon** set to **ON**, if **Groups** is empty, all the groups that the users are associated with are returned in the response. However, to return the required groups, specify the preferred groups in **Groups**.

By default, this option is set to **OFF**, the groups of users authenticated to the event are not returned in the response.

10h Select the **Allow to logon to this event by shared authenticator** option to allow users to login using shared authenticators. By default this option is disabled for the `Authenticators Management`, `Helpdesk`, `Helpdesk User`, `AdminUI`, `Search Card`, `Token Management`, and `Report Logon` events and enabled for all the other events.

NOTE: The **Allow to logon to this event by shared authenticator** option is displayed if you enable the **Enable sharing of authenticators** option in Authenticator management options policy.

---

11 Click **Save.**

SAML events support the step-up authentication. It does not prompt users to authenticate with the same method that is succeeded for an event during the session. Let us understand the step-up authentication with an example, assume there are three SAML events, SMEVT1, SMEVT2, and SMEVT3. Chains associated with each event are as follows:

- SMEVT1 - LDAP Password + Security Questions
- EVT2 - LDAP Password + SMS OTP
- EVT3 - Security Questions + SMS OTP

Possible scenarios:

- First the user logs in to SMEVT1 by furnishing LDAP password and valid secret questions. During the same session if the user tries to authenticate to SMEVT3 then a prompt to provide the SMS OTP is displayed. This happens because user has succeeded Security Questions method for SMEVT1.
- User logs in to SMEVT3 first with the Security Questions and SMS OTP. Later, the user can authenticate to SMEVT2 with the LDAP Password method because SMS OTP is succeeded in the previous event.

9.2.5 Creating a RADIUS event

When you want to add multiple RADIUS clients, you can add them to the predefined RADIUS Server event. But all the RADIUS clients will use the same authentication chain(s). If you want to configure specific authentication chain(s) for different RADIUS clients, then you must create a custom RADIUS event. To add a custom RADIUS event, perform the following steps:

1. Click **Events > New Event**.
2. Specify a name for the event.
3. Ensure that **Is enabled** is set to **ON**.
4. Select **RADIUS** from **Event Type**.
5. Select the chains that you want to assign to the event.
6. (Conditional) In **Risk Policy**, select the policy that you want to assign to this event for assessing the risk associated with a login attempt.
7. (Conditional) Click **Create New Policy** to create a new risk policy for this event.

   Clicking this option opens the Risk Settings page.

   IMPORTANT: **Risk Policy** and **Create New Policy** options are available when you enable Risk Settings. For more information, see Part III, “Configuring risk settings,” on page 265.

   ---
8. Set **Logon with Expired Password** with one of the following options based on your requirement:
   - **Allow**: Select this option to allow users to log in to the event with the expired LDAP password.
   - **Ask to change**: If the password has expired this option prompts users to change the password during logon. Change in the LDAP Password is supported only for the Active Directory repositories. However, the LDAP Password change in Advanced Authentication is not allowed when the LDAP Servers in the Repository settings are configured with port 389. The LDAP server rejects the new password.
   - **Deny**: Select this option to deny access to the event with the expired LDAP password. When the access is denied, the following message is displayed to users:

     ```
     You must change your password to logon.
     ```
9. Set **Bypass user lockout in repository** to **ON**, users, who are locked in the repository, must select and authenticate with the chain that does not include the LDAP Password method. By default, **Bypass user lockout in repository** is set to **OFF** and locked users cannot authenticate by using any chain.

To use this functionality, it is required to have more than one chain without the LDAP Password method assigned to the event. This is to provide more options to users.

NOTE: All authentication chains irrespective of whether it includes the LDAP Password method or not are displayed to users who are locked in the repository.

---

10 Set **Return groups on logon** to **ON** if you want to retrieve the group details of users who authenticated to the event in the authentication response.

With **Return groups on logon** set to **ON**, if **Groups** is empty, all the groups that the users are associated with are returned in the response. However, to return the required groups, specify the preferred groups in **Groups**.

The RADIUS protocol according to RFC (https://tools.ietf.org/html/rfc2865) has a 4KB limit of response size. The authentication response might exceed the set limit, if a user is a member of several groups. Therefore, it is recommended to use **Groups** to limit the groups' in the response.

By default, **Return groups on logon** is set to **OFF**, the groups of users authenticated to the event are not returned in the response.

11 Select the **Allow to logon to this event by shared authenticator** option to allow users to login using shared authenticators. By default this option is disabled for the `Authenticators Management`, `Helpdesk`, `Helpdesk User`, `AdminUI`, `Search Card`, `Token Management`, and `Report Logon` events and enabled for all the other events.

12 Configure Input rule

13 Configure Chain selection rule

14 Configure Result specification rule

You can configure the above RADIUS rules in RADIUS Options policy also. For more information about configuring the RADIUS rules in RADIUS Options Policy, see RADIUS Options.

The rules configured in RADIUS Options policy are called Global level rules and rules configured in RADIUS event are called Event level rules. All the RADIUS rules are executed in the following order.

14a Input rule configured in Global level rules.

14b Event Selection rule configured in Global level rules.

14c Input rule configured in Event level rules.

14d Chain selection rule configured in Event level rules.

14e Chain selection rule configured in Global level rules (if no chain in Event level rules).

14f Authenticate the user.

14g Result specification configured in Global level rules.

14h Result specification configured in Event level rules.

15 Click **Save**.

10 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.

   Advanced Authentication Windows Client 5.6 or newer, Advanced Authentication Linux PAM Client 6.0 or newer, Advanced Authentication Mac OS X Client 6.0 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 **New Endpoint**.
2. On the **New Endpoint** page, specify a **Name** of the endpoint and its **Description**.
3. Set **Is enabled** to **ON**.

4 Set **Is trusted** to **ON** if the endpoint is trusted.

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.

11 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:

- Section 11.1, “Authenticator management options,” on page 198
- Section 11.2, “Cache options,” on page 199
- Section 11.3, “Custom branding,” on page 200
- Section 11.4, “Custom messages,” on page 208
- Section 11.5, “Database options,” on page 214
- Section 11.6, “Delete me options,” on page 214
- Section 11.7, “Endpoint management options,” on page 215
- Section 11.8, “Event categories,” on page 216
- Section 11.9, “Geo fencing options,” on page 216
- Section 11.10, “Google reCAPTCHA options,” on page 217
- Section 11.11, “Help options,” on page 218
- Section 11.12, “Helpdesk options,” on page 219
- Section 11.13, “Linked chains,” on page 219
- Section 11.14, “Lockout options,” on page 220
- Section 11.15, “Login options,” on page 221
- Section 11.16, “Logon filter for Active Directory,” on page 223
- Section 11.17, “Mail sender,” on page 223
- Section 11.18, “OAuth2 Application,” on page 224
- Section 11.19, “Password filter for Active Directory,” on page 225
- Section 11.20, “RADIUS Options,” on page 226
- Section 11.21, “Reporting options,” on page 237
- Section 11.22, “SMS sender,” on page 237
- Section 11.23, “Voice sender,” on page 244
- Section 11.24, “Web Authentication,” on page 246
- Section 11.25, “Windows Credential Providers,” on page 249

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.

You can also double-click on the policy to edit the configuration.

3 Make the required changes for a specific policy.

4 Click **Save**.

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

11.1 Authenticator management options

This policy allows you to configure the following two settings:

- Section 11.1.1, “Enabling sharing of authenticators for the helpdesk administrators,” on page 198
- Section 11.1.2, “Disabling re-enrollment of the authenticators,” on page 198

11.1.1 Enabling sharing of authenticators for the helpdesk administrators

This setting allows a user to authenticate with his or her authenticator to another user’s account. The helpdesk administrator can share an authenticator of one user with another user.

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

The account of an helpdesk administrator must be added to the **SHAREAUTH ADMINS** group to grant privilege to share the authenticators. For more information about how to allow the helpdesk administrators to share authenticators, see “Local Repository”.

NOTE: Shared authenticators work only in the online mode. Cached login does not work for the shared authenticators. The supported methods for sharing authenticators are TOTP, HOTP, Password, Fingerprint, Flex OTP, Card, FIDO U2F, Smartphone, and RADIUS Client.

For more information, see “Sharing Authenticators”in the *Advanced Authentication- Helpdesk Administrator* guide.

11.1.2 Disabling re-enrollment of the authenticators

This setting allows you to restrict users from re-enrolling, editing, and deleting the enrolled authenticators in the Self-Service and Helpdesk portals and API integrations.

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

WARNING: If you access the Administration portal with a 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. 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).

11.2 Cache options

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.

This policy allows you to configure the following settings:

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

  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 **Cache expire time** is set to 0, to indicate that the cache never expires. Use the **Cache expire time** option to set the duration (in hours) to store user authenticators in Client cache. The maximum expiry time that you can set is 24 * 366 (8784 hours). This setting is applicable for the Advanced Authentication Clients.

  When a user logs in with cached authenticators, Advanced Authentication compares the last online login time with the current offline authentication time. If the time duration is less than or equal to the specified duration in **Cache expire time**, the user is authenticated to Clients.

  For example, consider the **Cache expire time** is set to 2 hours. The last online log in time of the user to Client is 1:00 PM. When the user tries to log in to Windows Client using cached authenticator credentials at 2:30 PM, the authentication is successful and the user is logged in to Windows Client. But, if the user tries to log in with cached authenticator credentials at 4:00 PM, the offline authentication fails and displays the following message as the cache has expired.

  ```
  Authenticators of <user name> were not cached. Press OK and try again to
  log in as local user or cached user
  ```

- By default, the **Allow Local caching for logons by shared templates** is set to **OFF**, to indicate that shared authenticators are not cached. To enable caching shared authenticators in Clients, set **Allow Local caching for logons by shared templates** to **ON**. Clients can use cached details for validation during the offline authentication.

  Before you enable this option, ensure to enable the following settings to cache shared authenticators:

  - **Enable Sharing of Authenticators** in **Policies > Authenticator management options**

    For more information, see Authenticator management options.

- **Enable Allow logon to this event by shared authenticator** for the required events in **Events**

  For more information, see Configuring an existing event.

NOTE: You can use the enforced cached logon instead of the default online logon, to improve the logon and unlock speed on Clients. For more information, refer to the following topics:

- For Linux, see "Configuring the enforced cached login" in the "*Advanced Authentication- Linux PAM client*" guide.
- For mac OS, see "Configuring the enforced cached Logon" in the "*Advanced Authentication - Mac OS X client*" guide.
- For Windows, see "Configuring the enforced cached login" in the "*Advanced Authentication - Windows client*" guide.

11.3 Custom branding

This policy allows you to customize the look and feel of the web portals, such as Helpdesk, Report, Tokens, Search card, new Enrollment and Administration Portals. By configuring this policy, you can change the title, logos, and application bar colors of the portals.

NOTE: When you make modifications, you can preview the changes you have applied in the sample Application Bar placed on the top of the page.

To configure the Custom Branding policy, perform the following steps:

1. Specify the title that you want to display in the application bar in **App Bar Title**.
2. Set **Show Title Text** to **OFF** to hide the title in the application bar. The **Show Title Text** is set to **ON** by default.
3. Specify the logo or title URL in **Title/Logo Link URL**. When a user clicks the logo or title in the application bar, the user is directed to the specified link.
4. Set the **Use Logo Image in App Bar** to **OFF** to hide logo in the application bar. The **Use Logo in App Bar** is set to **ON** by default.
5. Click **Choose File** in **App Bar Logo** and select the image from the local drive.

   NOTE: The resolution of image must not exceed 200x200 pixels. The image must be in `.png` format.

   ---

6. Click the **Custom Color** icon  in **Title Text Color** and select the preferred text color.
7. Click the **Custom Color** icon  in **Background Color** and select the preferred background color.

8 Set **Blend Two Background Colors** to **ON** to add two background colors and blend the colors in the application bar.

If **Blend Two Background Color** is set to **ON**, click the **Custom Color** icon  in **Background Color Left** and **Background Color Right,** and select the preferred background colors.

11.3.1 Customizing the login page of web authentication events

You can customize the messages on new Enrollment Portal or login page of the OAuth 2.0 or Open ID Connect and SAML 2.0 events. To do this, perform the following steps:

1 Set **Use Custom Branding File for Web Authentication** to **ON**.

By default, this option is set to **OFF**.

2 Click **Choose File** to upload the `osp-custom-resources.jar` file in the **Web Authentication Branding File**.

3 Click **Template** to download the branding template in the **Download Custom Branding Template**.

3a Save the `osp-custom-resources.jar` file.

3b Unzip the `osp-custom-resources.jar` file and in the **resources** folder open the file that you want to customize as follows:

- `naaf_enduser_custom_resources_<language>.properties` - Use this file to customize the text related to the all methods on the login and Chain Selection pages.

  For example, to edit the text on the **Authentication Chain Selection** page, customize the values of the following parameters in the `naaf_enduser_custom_resources_<language>.properties` file:

  - `NAAFENDUSER.ChainPageHeader`: To edit the title
  - `NAAFENDUSER.ChainPageSubHeader`: To edit the paragraph after the title
  - `NAAFENDUSER.ChainSelectChain`: To edit the name of chain list drop down

  NOTE: In case, you have customized the messages in the following files:

  - The `naaf_enduser_custom_resources_<language>.properties` file of `osp-custom-resources.jar`
  - **Policies > Custom Messages**

  Depending on the status of **Custom Branding**, the messages are displayed to end-users as follows:

  - With **Use Custom Branding File for Web Authentication** set to **ON**, Web authentication events display the customized messages from the `osp-custom-resources.jar` file
  - With **Use Custom Branding File for Web Authentication** is set to **OFF**, Web authentication events display the messages from **Custom Messages** policy

  ---

- `oidp_enduser_custom_resources_<language>.properties` - Use this file to customize css, logo, copyright text and links of the login page.

  For examples, refer to Example of customizing a login page.

NOTE: Ensure that you edit the attributes in the **Login page properties** section of the `oidp_enduser_custom_resources_<language>.properties` file for the custom branding of the login page.

You must also add your customized `.css` file in the **css** folder and required logo to the **images** folder of the `osp-custom-resources.jar` file.

3c After you edit the specific file in the **resources** folder, zip the file `osp-custom-resources.jar`.

NOTE: To avoid the manifest file overwriting, make sure the jar file has been unzipped and zipped instead of opening and closing it with the Java .jar tool. Otherwise use the following command line switch to set the manifest file and make sure the name of the manifest file matches the filename in the template:

```
jar -cm <jar file name> <manifest file name>.
```

4 (Conditional) To restore the default look and feel of the new Enrollment Portal, Administration Portal, and login page of web authentication events, click **Restore** adjacent to **Restore Default Branding**.

5 Click **Save**.

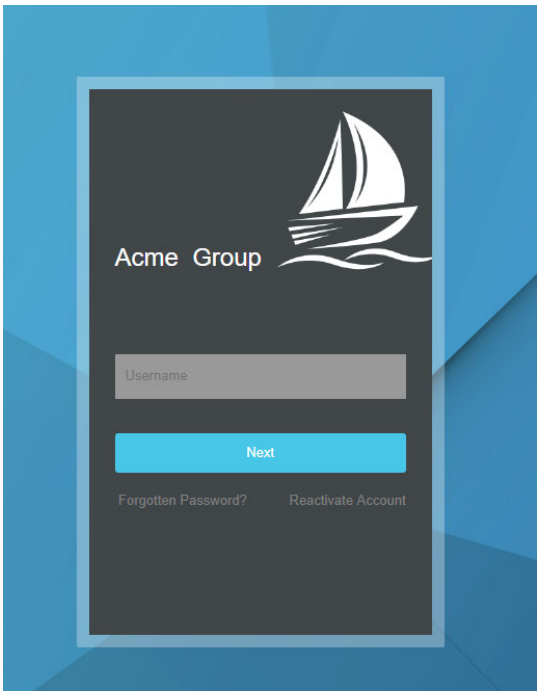
The following section describes an example of the customization that you can achieve for the Web authentication.

Example of customizing a login page

To achieve the customized login page in the Figure 11-1 for Acme Group of company, you can perform the following:

- “Adding a customized CSS for the login page” on page 203
- “Customizing the logo of an enterprise” on page 206
- “Customizing the copyrights” on page 207
- “Customizing the branding text” on page 207
- “Adding links on the login page” on page 208

Figure 11-1 *Customized page for acme group*



Adding a customized CSS for the login page

You can add a customized css file to reflect changes for the login pages.

The following sample.css file has been customized for achieving the customized login page in Figure 11-1 for the Acme Group of company.

```
/* general styles
------------------------------ */
body {
  margin:0;
  padding:0;
  background:#fff url("/osp/TOP/images/login_bg.jpg") no-repeat center
center fixed;
  -webkit-background-size: cover;
  -moz-background-size: cover;
  -o-background-size: cover;
  background-size: cover;
  font-family:Arial, Helvetica, sans-serif;
}
img {
  border:none;
  max-width: 100%;
}
/* login box
------------------------------ */
div.page-container {
  position:absolute;
  top: 50%;
  left: 0px;
  width:100%;
```

```
  margin:-265px auto 0 auto;
}
div.dialog {
  border: 12px solid rgba(255, 255, 255, 0.3);
  border-radius: 2px;
  width: 318px;
  max-width:100%;
  margin:0 auto;
  background-color: transparent;
}
div.dialog-content {
  height:525px;
  padding:0 15px;
  background:url(/osp/TOP/images/acme.png);
  background-color:#414749 ;
  background-position:180px 20px;
  background-repeat:no-repeat;
  font-family: Arial, Helvetica, sans-serif;
  text-align: left;
  }
.dialog-header {
  margin:0;
  padding: 150px 0 40px 0;
  color:#48c6e7;
  font-size:22px;
  font-weight:100;
  background: none;
}
div.dialog-header-content {
  display:block;
  color:#fff;
  font-weight: 200;
}
p { margin:0; padding:0; }
div.dialog-body {
  padding: 0;
}
.product-name {
  margin: 0;
}
#password, #Ecom_User_ID {
  color: #000 !important;
  background-color: #999;
  font-size: 13px;
  line-height: 20px;
  margin: 0 0 3px 0;
  padding: 11px 10px 12px;
  width: 100%;
  box-sizing: border-box;
  border: none;
  border-radius: 0;
}
.dialog-footer-content {
  display: none;
}
```

```
.button-container button, .btn {
  display: block;
  text-align: center;
  color: #fff;
  font-size: 13px;
  background-color: #48c6e7;
  border: none;
  margin: 30px 0 0 0;
  padding: 11px 10px 12px;
  box-sizing: border-box;
  width: 100%;
  cursor: pointer;
  -webkit-appearance: none;
  text-decoration: none;
}
.button-container button:hover {
  background-color:#00B4DF;
  border: none;
}
.input-box input {
  box-sizing: border-box;
  background-color: #999;
}
p.error {
    color: #cccccc;
    font-size: 13px;
    margin: 0;
    padding: 0 0 18px;
}
#logoutmsg, #logoutmsgsub { color:#fff; }
.error h1 { padding-bottom:20px; }
.help p { margin:0; padding:20px 0 0 0; font-size:11px; }
.help a { color:#cccccc; text-decoration:none; }
.help a:hover { color:#fff; }
.title {
  display: none;
}
.image-custom-link, .login-custom-link {
  display: inline;
}
.image-custom-link a {
  padding: 0;
}
.image-custom-link a:hover {
  color: #fff;
  background-color: transparent;
  display: inline;
  padding: 0;
}
.image-custom-link img {
  height: 0;
  width: 0;
}
#loginCustomLink1 {
  float: right;
```

```
}
/*-------------------------------------*\
    RESPONSIVE
\*-------------------------------------*/
@media only screen and (max-width:480px) {
  div.page-container {
    position: static;
    top: 0;
    margin: 0;
  }
  div.dialog {
    width: auto;
    margin: 0;
  }
}
```

Perform the following steps to add the sample.css file to the `osp-custom-resources.jar` file.

1. Open the `osp-custom-resources.jar` file.
2. Upload your .css file to the `css` folder.
3. Open the `resources` folder.
4. Open the `oidp_enduser_custom_resources_en_US.properties` file to edit the custom branding of the login pages in the English language.
5. Uncomment the line `OIDPENDUSER.LoginCss=reset.css,uistyles.css,uistyles_loginselect.css` by removing the # sign.

   You can add your .css file here. For example, `OIDPENDUSER.LoginCss=sample.css`.
6. Save the `oidp_enduser_custom_resources_en_US.properties` file.

Customizing the logo of an enterprise

You can edit the logo displayed on the login page of web authentication event using the parameter `OIDPENDUSER.LoginProductImage` available in the Login page properties.

For example, to edit the logo of the login page of an OAuth 2.0 event in the English language, perform the following:

1. Open the `oidp_enduser_custom_resources_en_US.properties` file and edit the following attribute:

   `OIDPENDUSER.LoginProductImage=company_img.png`.

   You can also edit the .css file. The following code has been added to the sample.css file to display the logo in the Figure 11-1:

```
div.dialog-content {
  height:525px;
  padding:0 15px;
  background:url(/osp/TOP/images/company_img.png);
  background-color:#414749 ;
  background-position:180px 20px;
  background-repeat:no-repeat;
  font-family: Arial, Helvetica, sans-serif;
  text-align: left;
  }
```

2 Ensure that you add the image that you want as a logo to the `images` folder with the name that matches with the attribute value in `OIDPENDUSER.LoginProductImage`.

   By default the images folder contains the image `company_img`.

Customizing the copyrights

You can edit the copyright text displayed on the login page of web authentication event using the parameter `OIDPENDUSER.50004` available under the **JSP Strings**.

For example, to remove the copyright note that is displayed on the login page of an OAuth 2.0 event in the English language:

1 Open the `oidp_enduser_custom_resources_en_US.properties` file and search the following parameter:

   ```
   #OIDPENDUSER.50004=Copyright [copy] [year] NetIQ[nbsp]Corporation, a Micro[nbsp]Focus company. All rights reserved
   ```

2 Uncomment the following parameter as follows:

   ```
   OIDPENDUSER.50004=
   ```

   This removes the copyright note from the web authentication event - login page.

Customizing the branding text

You can edit the branding text displayed on the login page of web authentication event using the parameter `OIDPENDUSER.LoginProductName` available in the **Login page properties** section of the `oidp_enduser_custom_resources_en_US.properties` file.

For example, to edit the branding of the company to **Acme Group**, perform the following:

1 Open the `oidp_enduser_custom_resources_en_US.properties` file and search the following parameter:

   ```
   #OIDPENDUSER.LoginProductName=Company[nbsp]Name[reg]
   ```

2 Edit the following parameter as follows:

   ```
   OIDPENDUSER.LoginProductName=Acme[nbsp]Group[reg]
   ```

If you want to remove the branding text **Acme Group, perform the following:**

1 Open the `oidp_enduser_custom_resources_en_US.properties` file and search the following parameter:

   ```
   #OIDPENDUSER.LoginProductName=Company[nbsp]Name[reg]
   ```

2 Uncomment the following parameter as follows:

```
OIDPENDUSER.LoginProductName=
```

This removes the branding text, Acme Group, from the web authentication event - login page.

Adding links on the login page

You can add links for the login page of the web authentication event.

For example, if you want to add the link **Forgotten Password** that is displayed on the login page in the English language, add the following:

1 Open the `oidp_enduser_custom_resources_en_US.properties` file.

2 Add the following:

```
OIDPENDUSER.50078=Click here if you've forgotten your username or
password, or if you need to register.
```

NOTE: The hyperlink for the text is taken from **Methods > LDAP Password > SSPR URL** (https://www.netiq.com/documentation/advanced-authentication-63/server-administrator-guide/data/ldap_pwd.html) in the Administration Portal.

11.4 Custom messages

In this policy, you can customize the error messages, method message and prompt message of a specific language.

For example, you can customize the default logon error message in English to `Your login failed.` In the Self-Service portal, when the user specifies wrong user name, the customized error message is displayed.

To customize the messages, perform the following tasks:

- Customizing messages in the custom localization file
- Customizing a specific message on the Portal

NOTE: The customized messages are cached in the Advanced Authentication server. The refresh interval for custom messages is one hour. Therefore, when you customize a message or upload a custom localization file, the respective message is displayed on the corresponding Advanced Authentication portals and clients after an hour.

You can also perform the following tasks in the **Custom Messages** policy:

- Customize the authentication request message displayed on the app. For more information, see Customizing authentication request message for Smartphone method.
- Customize the prompt messages of authentication methods for RADIUS event. For more information, see Customizing prompt messages of the authentication methods for RADIUS event.

- Customize message on the clients. For more information, see Customizing the messages for clients.
- Localize the Web UI and messages to an unsupported language. For more information, see Localizing the web UI and messages.

11.4.1 Customizing messages in the custom localization file

NOTE: The messages customized using the Custom localization file does not reflect on the new Enrollment Portal. However, to customize messages for the new Enrollment Portal, see Customizing the login page of web authentication events.

To customize preferred messages using the **Custom localization** file, perform the following steps:

1. Click **Custom Messages**.
2. Perform one of the following action to download the custom localization file on your local drive:
   - Click **Download original** to save the `custom_messages.tar.gz` file that contains the default messages.
   - If you have customized the messages, click **Download current messages** to save the `current_custom_messages.tar.gz` file that contains the latest messages.
3. Extract the files from the `custom_messages.tar.gz` file.
4. Navigate to the preferred language folder.

   To customize English messages, use the `custom_messages.pot` file and for other languages use the `custom_messages.po` file.
5. Open the `custom_messages.pot` file in the text format.
6. Specify the message in the `msgstr ""`.

```
msgctxt "errors.user_not_found"
msgid "User not found"
msgstr ""

msgctxt "method.swisscom.user_should_accept_request"
msgid "The user should accept your request with his/her mobile phone"
msgstr ""

msgctxt "method.messaging.cannot_determine_recipient_address"
msgid "Cannot determine OTP recipient address"
msgstr ""

msgctxt "method.virtual_password.password_will_expire"
msgid "Password will expire after ${days} days"
msgstr "Password will expire in ${days} days"

msgctxt "method.emergency_password.password_is_not_effective"
msgid "Emergency password is not effective yet. Wait ${wait_days} day(s)"
msgstr ""

```

7. Save the changes.
8. Compress the `custom_messages` folder to `.tar.gz` or `.zip` format.
9. Click **Browse** and select the compressed `custom_messages` file from the local drive.
10. Click **Upload**.

IMPORTANT: English is the default language and administrator cannot upload or delete the English language file.

11.4.2 Customizing a specific message on the Portal

To customize a specific message on the portal, perform the following steps:

1. Click **Custom Messages**.
2. Use the **Message filter** to search for a specific message or you can find the preferred message manually.
3. Use the **Message Group** to search a specific message by group. Options available are **All**, **Method messages**, **Error messages**, and **Other messages**.
4. Click the **Edit** icon next to the preferred message. You can also double-click on the message to edit the content.
5. Specify the message in the preferred language.
6. Click **Save**.

11.4.3 Customizing authentication request message for Smartphone method

You can customize the authentication request message that is displayed on the OpenText Advanced Authenticationapplication when user initiates Smartphone authentication. The authentication can be either to the endpoint or to the Advanced Authentication portals.

To customize the message for smartphone method, perform the following steps:

1. Click **Custom Messages**.
2. Search for one of the following keys:
   - `method.smartphone.authentication_hint` to edit the request message specific to endpoint authentication.
   - `method.smartphone.authentication_hint_no_endpoint` to edit the request message for any authentication that does not use endpoint such as Advanced Authentication portals login.
3. Click for the preferred key.
4. Specify any of the following parameters in the preferred language message as per your requirement:
   - `{user}` to fetch the user name.
   - `{client_ip}` to fetch the client IP address.
   - `{event}` to fetch the event name.
   - `{tenant}` to fetch the tenant name.
   - `{endpoint}` to fetch the endpoint name.
5. Click **Save**.

NOTE: The customized authentication request message will reflect on the OpenText Advanced Authentication application after an approximate delay of one hour.

For example, to customize the endpoint specific authentication message for the smartphone method you must search the key `method.smartphone.authentication_hint` and specify the message `{user} requested for authentication request from the client {client_ip} for the {event} to access the {endpoint}` in the field corresponding to English language. When the user tries to authenticate to Windows Client using the smartphone method then the customized message is displayed on the OpenText Advanced Authentication application as:

```
Bob requested for authentication request from the client 10.3.10.5 for the
Windows logon to access the Windows-machine-589.
```

11.4.4 Customizing prompt messages of the authentication methods for RADIUS event

You can customize prompt messages of the authentication methods that are configured for the RADIUS event. The customized prompt messages are displayed when a user initiates authentication to the RADIUS event using the configured methods.

To customize prompt message, perform the following steps:

1. Click **Custom Messages**.
2. Use the **Message filter** to search for a specific prompt message or you can find the preferred message manually.

   For example, specify `radius.totp.prompt` to search the prompt message displayed on RADIUS client for the TOTP method.
3. Click the Edit icon  or double-click on the preferred message to edit the content.
4. Specify the message in the preferred language on the **Edit Customer Message** page.
5. Click **Save**.

For example, consider Thomas, an administrator, wants to customize the default prompt message of the Voice OTP method that is configured for the RADIUS event. Thomas must first search the key `radius.voice_otp.prompt` and modify the message to `Specify the OTP that you heard from the voice call` in the text box corresponding to English.

When Mark, an end user tries to authenticate to RADIUS event using the Voice OTP method, the customized prompt message is displayed.

11.4.5 Customizing the messages for clients

You can customize the error messages, method message and prompt message specific to any authentication method that is displayed on endpoints such as Windows, Linux PAM, and Mac OS Clients.

To customize the message for clients, perform the following steps:

1. Copy the `aucore.custom.zip` custom localization file from one of the following path based on the Client:
   - **Windows**: `C:\Program Files\NetIQ\Windows Client\locale\`
   - **Linux PAM**: `/opt/pam_aucore/locale/`
   - **Mac OS X**: `Library/Security/SecurityAgentPlugins/aucore_login.bundle/Contents/Resources/aucore/locale`
2. Navigate to **Policies > Custom Messages** in the Administration portal.
3. Click **Choose file** and select the custom localization file.
4. Click **Upload**.

   NOTE: You can find the messages specific to the Clients with the prefix `client.` in the **Key**.

5. Search a specific message using the **Message filter** or find the preferred message manually.

   For example, specify `client.method.smartcard.waiting_for_card` to search the prompt message displayed for the Card method on all clients.
6. Click **Edit** next to the preferred message. You can also double-click on the message to edit the content.
7. Specify the message in the preferred language.
8. (Conditional) If you want to change the font size, color, and font family of custom message, insert the message within the HTML tag:

   ```
   <font size="3" color="red" face="Arial"><b>Message to Display</b></font>
   ```

   For example, to customize the font size, font color and bold the Caps lock message in English language on all clients, search the key `client.method.password.caps_lock` and specify the following HTML tag in **English**:

   ```
   <font size="5" color="blue" face="Arial"><b>Caps Lock in ON!</b></font>
   ```

   NOTE: The supported HTML tags to customize messages are as follows:
   - `<font size="x" color="text_color" face="font-family"> </font>`: To set the font size, color and font-family.
   - `<b> </b>`: To make the text bold.
   - `<i> </i>`: To make the text italic.

   NOTE: When you customize the font size of the message that gets displayed on Advanced Authentication Clients, for some of the font size, the message might be invisible or not readable.
9. Click **Save**.

NOTE: The customized messages reflect on the respective Clients after an approximate delay of one hour. However, after the first online log in to the Client, users can view the customized messages.

For example, consider Thomas, an administrator wants to customize the default method message (`Enter one-time password`) of the TOTP method that for all clients. Thomas must first search the key `client.method.totp.password` and modify the default message to `Specify the OTP that is displayed on Token or App` in the text box corresponding to English language.

When Mark, an end user tries to authenticate to Linux PAM Client using the TOTP method, the customized method message is displayed.

11.4.6 Localizing the web UI and messages

To localize the messages and web UI to an unsupported language, perform the following steps.

1. Click **Custom Messages**.
2. In **Custom locales**, click **Download Template** to save the `bundle-en.tar.gz` file that contains the default messages.
3. Extract the files from the `bundle-en.tar` file.
4. Navigate to the extracted folder.

   To localize core messages, use the `AuCore` file and to localize the web UI elements, use `webui` file.
5. Open the `AuCore` or `webui` file in the text format.
6. Specify the preferred language message in the `msgstr ""`.

   For example, if you need to localize `password will expire in $(days) days` message to Latin, specify `in password erit exspirare $ (dies) dierum` in `msgstr ""` as in the following image.

```
1  msgctxt "errors.user_not_found"
2  msgid "User not found"
3  msgstr ""
4
5  msgctxt "method.swisscom.user_should_accept_request"
6  msgid "The user should accept your request with his/her mobile phone"
7  msgstr ""
8
9  msgctxt "method.messaging.cannot_determine_recipient_address"
10 msgid "Cannot determine OTP recipient address"
11 msgstr ""
12
13 msgctxt "method.virtual_password.password_will_expire"
14 msgid "Password will expire after ${days} days"
15 msgstr "Password will expire in ${days} days"
16
17 msgctxt "method.emergency_password.password_is_not_effective"
18 msgid "Emergency password is not effective yet. Wait ${wait_days} day(s)"
19 msgstr ""
20
```

7. Save the changes.
8. Compress the `bundle-<language name>` folder to `.tar.gz` or `.zip` format.
9. In **Custom locales**, click  to add the template file.

10 Select the preferred language name in **Locale**.

11 Click **Choose File** and select the compressed `bundle-<language>.tar.gz` file from the local drive.

12 Click **Upload**.

11.5 Database options

This policy allows you to customize the configuration of all the database servers (PostgreSQL) in the cluster. You can modify the parameters, such as maximum connections, cache limit, shared buffers size, WAL (Write Ahead Logging) disk use, number of workers for parallel queries and so on by configuring this policy.

NOTE: You cannot apply this policy for Advanced Authentication as a Service (SaaS).

The parameters as well as the values that you specify in Optional Database parameters are validated. If the defined parameter is invalid, an error message with the reason appears. The parameters are replicated to all the database servers in the cluster when you save this policy. Also, applying the changes reboots the database servers in the cluster and changes reflects after 2 minutes.

NOTE: Each Advanced Authentication server spawns from 2.5 up to 10 DB connections per CPU core based on the user activity. The number of DB connections increase when the workload (authentication requests) is high.

For more information on the different parameters, see Setting Parameters (https://www.postgresql.org/docs/13/config-setting.html).

To get right estimation and generate the configurable parameters, see PGTune (https://pgtune.leopard.in.ua/#/).

NOTE: Ensure to set the DB version to 10 to generate only the supported parameters.

11.6 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 the Delete me policy** to **ON,** the users can view the **Delete me** option in a drop-down by clicking on the user name on the top-right corner of the Self-Service portal.

NOTE: To comply with General Data Protection Regulation (GDPR), you must set the **Enable the Delete me policy** option to **ON**.

NOTE: When a user from Local repository uses the **Delete me** option, it completely removes the user account along with the enrolled authenticators associated with the account.

11.7 Endpoint management options

In this policy, you can configure the following settings for managing an endpoint:

- **Require the administrator password to register an endpoint or workstation**: Set this option to **ON** for registering an untrusted endpoint from any IP address. Typically, this option is configured along with **Whitelist IP address**.

  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.

- **Allow unprivileged user to re-register an endpoint or workstation**: Set this option to **ON** to allow all users to re-register an endpoint though the endpoint with same name exists in the Advanced Authentication server. The user is required to specify user name and LDAP password to re-register the endpoint. This option is set to **OFF** by default.

  With this option set to **OFF**, users with ENROLL ADMIN or FULL ADMIN privileges are allowed to re-register an endpoint.

- **Whitelist IP Address**: Add the preferred IP addresses to the **Whitelist IP Address** to register either a trusted or an untrusted endpoint from these IP addresses. You can add a single IP address, multiple IP addresses, or a range of IP addresses to the whitelist. The IP address must be in IPv4 or IPv6 format.

  The following conditions summarizes the use of endpoint management options:

  - **Whitelist IP Address** is empty and **Require the administrator password to register an endpoint or workstation** is **OFF**: Untrusted endpoints can be registered from any IP address without the administrator's credentials.

    Regardless of the status of **Require the administrator password to register an endpoint or workstation** and **Whitelist IP Address** options, the administrator's credentials are required to perform the following actions:

    - To delete and update any endpoint.
    - To register a trusted endpoint.

    Endpoint registration is restricted only from those IPs that are specified in **Whitelist IP Address**.

  - **Whitelist IP Address** is empty and **Require the administrator password to register an endpoint or workstation** is **ON**: The administrator's credentials are required to register an untrusted endpoint from any IP address.

  - IP addresses are specified in **Whitelist IP Address** and **Require the administrator password to register an endpoint or workstation** is **ON**: The administrator's credentials are required to register untrusted endpoints only from the IP addresses specified in the whitelist.

    The endpoint registration request from any other IP address that is not specified in the whitelist is blocked automatically.

11.8 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.

11.9 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.

11.10 Google reCAPTCHA options

The **Google reCAPTCHA Options** policy helps to prevent the Advanced Authentication web portals login page from bots and to confirm that the user is a human and not a robot. This policy adds an additional layer of security before users go through multi-factor authentication. A series of images are displayed and the users must select the images for the specified condition to login.

To configure the Google reCAPTCHA for Advanced Authentication, you must perform the following configuration tasks:

- Section 11.10.1, “Registering the Google reCAPTCHA account,” on page 217
- Section 11.10.2, “Configuring Google reCAPTCHA for Advanced Authentication,” on page 218
- Section 11.10.3, “Enabling the Google reCAPTCHA options policy for events,” on page 218

11.10.1 Registering the Google reCAPTCHA account

Before you configure Google reCAPTCHA in Advanced Authentication, you must have a Google reCAPTCHA account.

To register for the Google reCAPTCHA account, perform the following steps:

1. Log in to the Google reCAPTCHA website with your Google account.
2. Click **Get reCAPTCHA**.
3. Specify a **Label**, select **reCAPTCHA V2** from **Choose the type of reCAPTCHA.**
4. Specify the `IP address` or the domain name of the Advanced Authentication server in **Domain**.
5. Accept the terms of Google reCAPTCHA.
6. Click **Register**.
7. Copy the **Site key** and **Secret key** to configure reCAPTCHA in Advanced Authentication. For more information, see Configuring Google reCAPTCHA for Advanced Authentication.

NOTE: If you forget the generated secret key, you can retrieve it from your Google account.

WARNING: If you have enabled the Google reCAPTCHA policy for the Admin UI event, you must consider the following guidelines. Otherwise, a deadlock scenario can happen and you will not be able to access the Administration portal without the cluster re-installation:

- If the site key or secret key gets deleted at the Google server, you will not be able to get the same site key or secret key. The site key and secret key used on the Administration portal are no more valid and there is no way to bypass the reCaptcha on the Administration portal.
- If you have registered the reCAPTCHA for one domain name and you change the domain name or migrate the Advanced Authentication server to another domain name, the site key or secret key used on the Administration portal are no more valid.

11.10.2 Configuring Google reCAPTCHA for Advanced Authentication

To configure Google reCAPTCHA for Advanced Authentication, perform the following steps:

1. Log in to the Administration portal.
2. Click **Policies > Google reCAPTCHA Options**.
3. Specify the **Site Key** and **Secret Key** that you received when you registered for a Google reCAPTCHA account.

   For more information about how to register the Google reCAPTCHA account, see "Registering the Google reCAPTCHA account".
4. Click **Test** to test the policy after the configuration.
5. Click **Save**.

11.10.3 Enabling the Google reCAPTCHA options policy for events

After you configure the Google reCAPTCHA policy, you must enable the policy for the respective events.

To enable the policy for events, perform the following steps:

1. Click **Events**.

   NOTE: You can enable the Google reCAPTCHA policy only for the Admin UI event, Authenticators Management event, Helpdesk event, Helpdesk User event, Report logon event, , Tokens Management event, and Web authentication events such as OAuth and SAML 2.0 events.

   ---
2. Set **Enable Google reCAPTCHA** to **ON**.
3. Click **Save**.

11.11 Help options

This policy allows you to hide the Help icon displayed on the portals, such as Administration, Self Enrollment, and Helpdesk. Also, you can modify the documentation URL where the user lands with a click on the Help icon.

You can configure this policy with the following options:

- **Documentation URL**: The URL format is *https://www.netiq.com/documentation/advanced-authentication-{version}/{path}* by default. You can modify the URL format to link the documentation available on another domain.
- **Show help**: Set this to **OFF** to hide the Help icon on the Administration, Helpdesk, and Self Enrollment portals. The Help icon helps the user to view the context and settings related each page. This option is set to **ON** and the Help icon is displayed by default.

11.12 Helpdesk options

In this policy, you can configure the following settings for the Helpdesk portal:

- **Ask for the credentials of the managed user**: Set this to **ON** to prompt the helpdesk administrator to provide the credentials of the managed user in the Helpdesk portal. This enhances security, however reduces convenience of the operations.

  When this setting is enabled, the helpdesk administrator must know the users' credentials to manage their authenticators. 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**, the user management becomes faster, but less secure.
- **Allow to unlock user accounts**: Set to **ON** to allow a helpdesk administrator to unlock users who are locked in the Advanced Authentication server local repository. Users are locked when the **Lockout options** policy is enabled. The helpdesk administrator can view and unlock the users in the Helpdesk portal under the **Locked Users** tab.
- **Allow to manage endpoints**: Set **Allow to manage endpoints** to **ON** to allow a helpdesk administrator to manage the endpoints of the Advanced Authentication server. When the helpdesk administrator logs in to the Helpdesk portal, an **Endpoints** tab is displayed where all the endpoints are listed. The helpdesk administrator can remove the endpoints. This option is disabled by default. For more information, see "Managing endpoints".
- **Allow to view user report**: Set this to **OFF** to hide the **User report** tab for a helpdesk administrator. This option is enabled by default to allow the helpdesk administrator to view the user's login report in the Helpdesk portal.

11.13 Linked chains

This policy allows users to authenticate with a simple chain for a specified duration after authenticating with a high-security chain. Enabling this policy allows users to use a single method chain (the linked chain) for a defined grace period after authenticating with a more secure multi-factor chain (the Required chain).

NOTE: This policy has replaced the **Last Logon Tracking Options** policy.

For example, if a user authenticates with the `LDAP Password+Card` chain once in a day, the user can further use a linked chain with only the `Card` method without the `LDAP Password` method, or if a user authenticates with the `Fingerprint+Smartphone` chain once in every four hours, the user can authenticate once with this chain and next authentication he can use only the linked `Smartphone` chain. The duration for which he can use the linked chain depends on the grace period that you specify in the Required chain option.

Perform the following steps to configure this policy:

1. **Enable linked chains**: Turn this option to **ON** to enable the linked chain policy.
2. **Hide required chain**: After using the required chain within the grace period, a user will see both the required and linked chains.

   Use this option to hide the required (high-security) chain after you authenticate once. Therefore after authenticating with the required chain, instead of displaying both the chains, only the linked chain is displayed. By default, this option is disabled.

3 **Limit by same endpoint**: Use this option to restrict a user to authenticate with the alternate linked chain only on the endpoint on which the user has successfully authenticated with a required chain, during the grace period. This option increases security by preventing a user to get authenticated on another endpoint after authenticating with the required (main) chain on an endpoint. By default, the option is **ON**.

For example, Bob authenticates on a Windows Client endpoint named `System1` with a required chain **Card+LDAP password**. Now, Bob wants to get authenticated to another Windows Client endpoint named `System2`, with a linked chain **Card**. When the **Limit by same endpoint** option is enabled, Bob will not be able to authenticate on System2 with the linked chain Card. He must first authenticate with the required chain **Card+LDAP password** on System2.

IMPORTANT: If you use the linked chains to access the Advanced Authentication portals or web integrations, set **Limit by same endpoint** to **OFF**.

---

4 Click **Save**

11.14 Lockout options

In this policy, you can configure settings to lock a user's account when the user fails 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.
- **Attempts failed**: 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 900 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 OpenText eDirectory, you must configure the Intruder Detection appropriately.

  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.

  The Helpdesk administrator can also unlock the locked users, if the **Allow to unlock user accounts** is enabled in the Helpdesk Options policy.

- **Lock if authenticator test was failed**: This option allows to lock the users who fail an authenticator's test in the Self-Enrollment portal for the number of times specified in **Attempts failed**.

  By default, this option is set to **OFF**. This indicates that the users will not be locked if they fail in the test process in the Self-Enrollment portal. You can enable the option to lock the user who tests an enrolled method and the test fails for the number of times specified in **Attempts failed**.

IMPORTANT: To enable the **Lock if authenticator test was failed** option, ensure to enable the **Lockout Options** policy.

11.15 Login options

In this policy, you can configure the settings to add default repository and ensure not to disclose valid username for malicious attack.

This policy allows you to configure the following settings:

- **Default repository**: You can 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**.

- **Username disclosure**: This option is set to **OFF** by default. It is recommended to keep default setting to prevent security vulnerabilities and to make it difficult for hackers to predict the valid username.

  If you set **Username disclosure** to **ON** and a user specifies an invalid username on the Advanced Authentication login page, an error message `User not found` is displayed. When the user specifies a valid username, the associated chain details are prompted to confirm the specified username and disclosing valid username. This can cause security vulnerability making it easy for attackers to guess the valid username.

  When this option is set to **OFF**, chain details are displayed instead of error message even though a user specifies an invalid username on the login page. A user can select a preferred authentication method. If the input data specific to the selected method is incorrect, a generic message `Invalid credentials` is displayed. This does not disclose whether username or first-factor authentication is incorrect.

  For example, a user specifies an invalid username, selects the SMS OTP method from the authentication chain. In this case, the SMS with OTP is not sent to the user. If the user specifies some random 6 digit as OTP, the server prompts an error message `Incorrect OTP password`. This helps the user to determine that specified username is valid though it is invalid.

- **LDAP caching**: This option allows you to enable or disable the caching of a user's information on the Advanced Authentication server. This information can be the lockout status of users, whether users have been disabled, or about the expiry of a user's password.

  By default, the option is set to **OFF**. This indicates that the Advanced Authentication server communicates with the LDAP server each time to check a user's information. You can enable the option to allow the caching of a user's information. Enabling the option increases the performance and cache the user's information for 5 minutes. However, it may also lead to security vulnerabilities. Therefore, it is recommended to set the option to **OFF**.

- **Email as login name**: This option enables the user to use Email address as the login name.

  By default, the option is set to **OFF**. Once you set this option to **ON**, the user can authenticate by specifying user's Email address in login name without specifying the tenant or repository name. When the user specifies the Email address, Email attributes in the repository is matched against the domains configured for the tenant to identify the tenant.

- You can specify the domain names in the **Login domains** field so that the Advanced Authentication allows the specified domain users to log in with their email address if **Email as login name** option is enabled.
- **Logon timeout (seconds)**: You can set the maximum duration of the logon session in this field. If a user fails to specify the login credentials within the specified duration, the session gets terminated. This value applies to all web-based authentication sessions. By default, the value is set to `900` (15 minutes).

  NOTE: The **Logon timeout (seconds)** and **Logon inactivity timeout (seconds)** options are supported only in the Advanced Authentication as a Service (SaaS) model. In the on-premises model of Advanced Authentication, these options will be available in the upcoming 6.3 Service Pack 7 release.

  For example, A user must specify LDAP Password and SMS OTP to authenticate to a web application. The Logon timeout is set to 180 seconds (3 minutes).

  The user action and equivalent outcome are as follows:

  - A user specifies the LDAP Password and waits for SMS OTP. Later, the user enters the OTP within 2 minutes. The authentication is successful.
  - A user specifies the LDAP Password and waits for SMS OTP. Later, the user enters the OTP after 3 minutes. The authentication fails.
- **Logon inactivity timeout (seconds)**: You can set the maximum inactivity timeout of the logon session in this field. If there is no action from the user within the specified duration, the session gets terminated. This value applies to all web-based authentication sessions. By default, value is set to `300` (5 minutes).

  NOTE: While authenticating with the Password and LDAP password methods, the user can enter the password within the **Logon timeout** duration. The **Logon inactivity timeout** does not apply to these methods.

  For example, A user must specify LDAP Password and Smartphone to authenticate to a web application. The Logon inactivity timeout is set to 30 seconds.

  The user action and equivalent outcome are as follows:

  - A user specifies the LDAP Password and waits for the push notification on the smartphone. There is an action at 30 seconds intervals then the user accepts the push notification within the Logon timeout duration. The authentication is successful.
  - A user specifies the LDAP Password and waits for the push notification on the smartphone. There is no action at 30 seconds interval and accepts the push notification at the 31st second. The authentication fails.

11.16 Logon filter for Active Directory

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.

11.17 Mail sender

In the **Mail sender** policy, you can configure settings for the Email OTP method to facilitate sending email messages with one-time passwords to 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. **Authentication Required**: By default, this option is set to **OFF**, keep the option in this state if your SMTP server does not require authorization.

   Set this option to **ON**, to specify the password required for the SMTP server authorization.
4. **Username**: The username of an account that is used to send the authentication email messages. For example, noreply or `noreply@company.com`.
5. **Password**: The password for the specified account. Is required when **Authentication Required** is set to **ON**.
6. **Sender email**: The email address of the sender.
7. **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**.

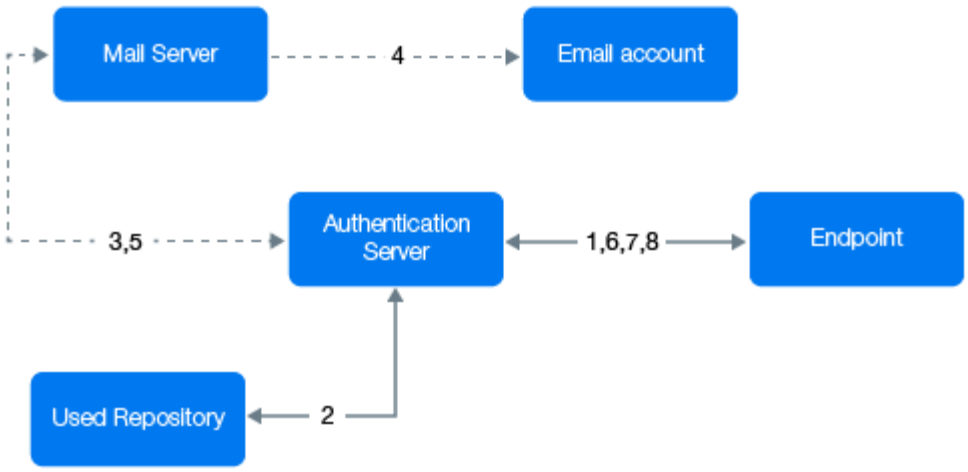
2c Click **Send test message!**.

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).

11.18 OAuth2 Application

This policy allows you to add a service account to the OAuth2 protocol-based applications that can access the Advanced Authentication API. The service account is specific to an application rather than an end-user. A unique Client ID and Secret are generated when you add an application. These details

must be configured in the application to establish trust between the application and the Advanced Authentication server. With the established connection, the application accesses the Advanced Authentication API without approval from a user.

With the **Client ID** and **Secret**, the application sends the token request to the Advanced Authentication Server, which returns an access token. Then, the application uses the token to access the required API. The scope is also defined for each application to restrict the access to a specific API.

For example, `api/v1/chains` is accessible without any scope. However, to view the tenants list, the scope must be set to full admin.

To add an application, perform the following steps:

1. Click **Add** in the **OAuth2 applications** policy.
2. Specify the application name in **Name**.
3. Copy the **Client ID** and **Secret** for further use.

   NOTE: When you save the application, **Secret** will not be displayed.

4. (Opitonal) Set **Full admin** to **ON** to grant the full administrator scope to the application so all APIs are accessible without any restriction. By default, **Full admin** is set to **OFF** and full administrator scope is not granted.

   For example, The API `/api/v1/tenants` is accessible only with the full administrator privilege.
5. (Optional) Set **Tenant admin** to **ON** to grant the tenant administrator scope to the application. With this application can access APIs that are applicable. By default, **Tenant admin** is set to **OFF** and tenant scope is not granted.

11.19 Password filter for Active Directory

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.

NOTE: If you do not include the LDAP Password method in a chain, a prompt to perform a synchronization is displayed. Set **Save LDAP password** to **ON** in LDAP Password method, 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.

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 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: If **Enable local caching** is set to **ON** in the Cache Options policy and when the password is changed or reset in the Active Directory. Then, a user is prompted to synchronize the password while logging in to Windows irrespective of the status of the following **Password Filter for AD** settings:

  - **Update password on change**
  - **Update password on reset**

  If **Enable local caching** is set to **OFF**, the Password Filter works according to the settings configured in this policy.

NOTE: You must install the Logon Filter on Domain Controllers to function the Password Filter for Active Directory.

For more information, see Advanced Authentication - Logon Filter.

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.

11.20 RADIUS Options

In this policy, you can define rules using regular expressions to accomplish the following actions:

- Select an appropriate chain for authenticating users to the RADIUS client
- Authenticate users to a specific event when multiple RADIUS events are available
- Display associated user groups in the authentication response after a successful authentication to the RADIUS client
- Select a particular chain based on the information that the user specifies on the RADIUS client

  For example, if a user specifies username&chain-short-name (bob&OTP), then select the chain with the LDAP and SMS OTP methods. In case, the user specifies only the username (bob) then select the chain with LDAP and Smartphone methods.

  NOTE: The chain short name is defined using the regular expressions in either Chain Selection or Event Selection rule.

- Define a specific authentication chain for a RADIUS client when there are multiple RADIUS clients mapped to the same RADIUS event

You can define the following rules in this policy:

- Section 11.20.1, “Input rule,” on page 227
- Section 11.20.2, “Event selection rule,” on page 228
- Section 11.20.3, “Chain selection rule,” on page 229
- Section 11.20.4, “Result specification rule,” on page 230
- Section 11.20.5, “Adding clients,” on page 236

To understand how to configure RADIUS options policy with rules, use the following sample scenarios:

- Scenario 1: selecting an authentication chain based on NAS ID and display groups of the authenticated user
- Scenario 2: mapping RADIUS requests to a specific RADIUS server event based on NAS ID and display Email address of the authenticated user

11.20.1 Input rule

Configure this rule to obtain the user name or the chain short name from user-specified details in the RADIUS client. The details obtained from the RADIUS client are sent to the RADIUS server for validating users. To enable the RADIUS client to select a specific chain for authenticating a user based on the obtained chain short name, use this rule along with the **Chain selection** rule.

To configure the input rule, perform the following steps:

1. Navigate to **Policies > RADIUS Options**.
2. Click **Add** in the **Input rules** section.
3. Specify the following details based on your requirement:
   - **Target-Input-Attribute**: Specify the attribute or variable that carries the user specified data to the RADIUS server in the Access-Request packet.
   - **Source-Input-Attribute**: Specify the attribute that stores the user-specified details.
   - **Regular expression**: Specify the condition to obtain user-specified details.
   - **Result specification**
   - **Comment**: If any.
4. Click **OK**.

Examples

Example 1: You can define the input rule as follows to obtain chain short name from user specified <username>&<short-chain-name> in the **Username** while logging in to the RADIUS client:

Target-Input-Attribute: chain_name

Source-Input-Attribute: User-Name

Regular expression: (.+)&(.+)

Result specification: Extract chain from User-Name and put into "chain_name" variable

After you configure, the rule looks as follows:

```
chain_name / User-Name / (.+)&(.+) / {2}
```

Example 2: You can define the following input rules to achieve the following:

- **Rule 1**: To extract the password and set to the variable, `User-Password`:

  Target-Input-Attribute: User-Password

  Source-Input-Attribute: User-Password

  Regular expression: (.*)(.{6})

  Result specification: {1}

- **Rule 2**: To extract the six digits OTP from password and set to the variable, `User-OTP`:

  Target-Input-Attribute: User-OTP

  Source-Input-Attribute: User-Password

  Result specification: {2}

After you configure, the rules are displayed as follows:

```
User-Password / User-Password / (.*)(.{6}) / {1}
```

```
User-OTP / User-Password / (.*)(.{6}) / {2}
```

11.20.2 Event selection rule

Configure this rule to map the requests from the RADIUS client to a specific RADIUS event based on the input attribute and condition (regular expression).

To configure the Event selection rule, perform the following steps:

1. Navigate to **Policies > RADIUS Options**.
2. Click **Add** in the **Event selection** section.
3. Specify the following details based on your requirement:
   - **Input-Attribute**
   - **Regular expression**
   - **Result specification**
   - **Comment**
4. Click **OK**.

Examples

Example 1: An administrator configures an event **RADIUS Server2** with OpenVPN as RADIUS client, and the value of NAS ID is 12345.

To map all requests containing 12345 as NAS ID to RADIUS Server2, define the following event selection rule:

Input-Attribute: NAS-Identifier

Regular expression: ^12345$

Result specification: RADIUS Server2

After you configure, the rule looks as follows:

```
NAS-Identifier / ^12345$ / RADIUS Server2
```

Example 2: There are two RADIUS events and two RADIUS clients as follows:

RADIUS Events	RADIUS Clients
RADIUS Server	172.16.0.1
RADIUS Server2	192.168.0.1

To map all requests from 172.16.0.1 to RADIUS Server event and 192.168.0.1 to RADIUS Server2 respectively, define the following event selection rules:

Rule 1	Rule 2
Input-Attribute: Packet-Src-IP-Address	Input-Attribute: Packet-Src-IP-Address
Regular expression: 172.16.0.1	Regular expression: 192.168.0.1
Result specification: RADIUS Server	Result specification: RADIUS Server2

After you configure, rules are displayed as follows:

```
Packet-Src-IP-Address / 172.16.0.1 / RADIUS Server
```

```
Packet-Src-IP-Address / 192.168.0.1 / RADIUS Server2
```

11.20.3 Chain selection rule

Configure this rule to select a specific chain for authenticating users to the RADIUS client. A chain is selected based on the input attribute and condition (regular expression).

To configure the Chain selection rule, perform the following steps:

1. Navigate to **Policies > RADIUS Options**.
2. Click **Add** in the **Chain selection** section.
3. Specify the following details based on your requirement:
   - **Input-Attribute**
   - **Regular expression**
   - **Result specification**
   - **Comment**
4. Click **OK**.

For example, a RADIUS event has two RADIUS clients and two chains defined.

To select a specific chain from multiple chains based on NAS ID of RADIUS client, defined the the following chain selection rules:

Rule 1	Rule 2
Input-Attribute: NAS-Identifier	Input-Attribute: NAS-Identifier
Regular expression: ^12345$	Regular expression: ^openvpn$
Result specification: LDAP + SMS	Result specification: LDAP + Smartphone

After you configure, the rules look as follows:

```
NAS-Identifier / ^12345$ / LDAP + SMS
```

```
NAS-Identifier/ ^openvpn$ / LDAP + Smartphone
```

11.20.4 Result specification rule

Configure this rule to display relevant details of a user in the RADIUS client after authentication. Details can be group name of the user, tenant name, phone number, e-mail address and so on.

To view the list of supported attributes, see Used attributes.

To configure the Result specification rule, perform the following steps:

1. Navigate to **Policies > RADIUS Options**.
2. Click **Add** in the **Result specification** section.
3. Specify the following details:
   - **Return-Attribute**
   - **User attribute**
   - **Regular expression**
   - **Result specification**
   - **Comment**
4. Click **OK**.

For example:

To display only group names of authenticated user on the RADIUS client define the result specification rule as follows:

Return-Attribute: Filter-Id

User attribute: groups

Regular expression: .*?CN=(.*?)(,|$)

Result specification: {1}

After you configure, the rules look as follows:

```
Filter-Id / groups / .*?CN=(.*?)(,|$) / {1}
```

To display the group name of authenticated user on the RADIUS client in the format CN= group name, define the result specification rule as follows:

Return-Attribute: Filter-Id

User attribute: groups

Regular expression: .*?(CN=.*?)(,|$)

Result specification: {1}

After you configure, the rules look as follows:

```
Filter-Id / groups / .*?(CN=.*?)(,|$) / {1}
```

To display the tenant name of authenticated user on the RADIUS client define the result specification rule as follows:

Return-Attribute: User-Name

User attribute: tenant_user_name

After you configure, the rules look as follows:

```
User-Name / tenant_user_name
```

Following table describes the supported user attributes.

Attributes	Description
name	Use this attribute to display name of the user
sid_hex	Use this attribute to display user SID (AD only) in hexadecimal format
repo_name	Use this attribute to display repository name
tenant_name	Use this attribute to display a tenant name
groups	Use this attribute to display group of the user
dn	Use this attribute to display distinguished name of the user
cn	Use this attribute to display common name of the user
email	Use this attribute to display email address of the user
mobile_phone	Use this attribute to display mobile phone of the user

Scenario 1: selecting an authentication chain based on NAS ID and display groups of the authenticated user

An organization has configured the default RADIUS Server event with the following authentication chains and RADIUS clients:

- Authentication chains:
  - LDAP + SMS

- LDAP + Smartphone
- LDAP + HOTP

- RADIUS clients:
  - Client 1: 10.0.0.1 with NAS ID 12345id
  - Client 2: 10.0.0.2 with NAS ID 0789id

Now, the administrator wants to achieve the following tasks as per the RADIUS authentication requirement:

- Select a chain based on NAS ID
  - If the NAS ID is 12345id, select LDAP + Smartphone
  - If the NAD ID is 0789id, select LDAP + SMS
- Display user associated group names after authentication

For this requirement, you can configure the RADIUS policy with Input, Chain selection, and Result specification rules.

Configuration steps:

1. Click **Policies > RADIUS Options** on the Administration portal.
2. Add Input, Chain selection, and Result specification rules as follows:

Rule	Procedure
Input rules	1. Click **Add** in **Input rules**. 2. Specify the following details: ◆ **Target-Input-Attribute**: User-Name ◆ **Source-Input-Attribute**: User-Name ◆ **Regular expression**: (.+)&(.+) ◆ **Result specification**: {1} ◆ **Comment**: To retrieve the user name 3. Click **OK**.

Rule	Procedure
Chain selection	**Rule 1**: 1. Click **Add** in **Chain selection**. 2. Specify the following details: ◆ **Input-Attribute**: NAS-Identifier ◆ **Regular expression**: ^12345id$ ◆ **Result specification**: LDAP + Smartphone ◆ **Comment**: To select a chain 3. Click **OK**. **Rule 2**: 1. Click **Add** in **Chain selection**. 2. Specify the following details: ◆ **Input-Attribute**: NAS-Identifier ◆ **Regular expression**: ^0789id$ ◆ **Result specification**: LDAP + SMS ◆ **Comment**: To select a chain 3. Click **OK**.
Result specification	1. Click **Add** in **Result specification**. 2. Specify the following details: ◆ **Return-Attribute**: Filter-Id ◆ **User attribute**: groups ◆ **Regular expression**: .*?CN=(.*?)(,|$) ◆ **Result specification**: {1} ◆ **Comment**: To display only group name of an authenticated user 3. Click **OK**.

After you implement this RADIUS rules, the following are possible scenarios:

Scenario	Chain Selected for Authentication	Result
A user initiates authentication from RADIUS Client 1 (NAS ID: 12345id)	LDAP + Smartphone	Group names of the user is displayed on the RADIUS Client 1 after successful authentication.
A user initiates authentication from RADIUS Client 2 (NAS ID: 0789id)	LDAP + SMS	Group names of the user is displayed on the RADIUS Client 2 after successful authentication.

Scenario 2: mapping RADIUS requests to a specific RADIUS server event based on NAS ID and display Email address of the authenticated user

An organization has configured two RADIUS Server events with the following details:

Event Name	Chains Assigned to Event	IP Address of RADIUS Client	RADIUS Client Name	NAS ID
RADIUS Server	◆ LDAP + SMS ◆ LDAP + HOTP	10.0.1.1	openvpn1	abc123
RADIUS Server 1	◆ LDAP + Smartphone ◆ LDAP + TOTP	10.0.1.2	openvpn2	xyz456

Now, the administrator wants to achieve the following tasks as per the RADIUS authentication requirement:

- Send request from a RADIUS client to a specific RADIUS Server event based on the chain short name:
  - If the NAS ID is abc123, map requests to RADIUS Server event
  - If the NAS ID is xyz456, map requests to RADIUS Server 1 event
- Display email address of users after authentication

For this requirement, you can configure the RADIUS policy with the Input rule, Event selection rule, and Result specification rule.

Configuration steps:

1. Click **Policies > RADIUS Options** on the Administration portal.
2. Add Input, Event selection and Result specification rules as follows:

Rule	Procedure
Input rule	1. Click **Add** in **Input rules**. 2. Specify following details: ◆ **Target-Input-Attribute**: chain_short_name ◆ **Source-Input-Attribute**: User-Name ◆ **Regular expression**: (.+)&(.+) ◆ **Result specification**: {2} ◆ **Comment**: To retrieve text after the & symbol 3. Click **OK**.

Rule	Procedure
Event selection	**Rule 1**: 1. Click **Add** in **Event selection**. 2. Specify following details: ◆ **Input-Attribute**: NAS-Identifier ◆ **Regular expression**: ^abc123$ ◆ **Result specification**: RADIUS Server ◆ **Comment**: To select an event 3. Click **OK**. **Rule 2**: 1. Click **Add** in **Event selection**. 2. Specify following details: ◆ **Input-Attribute**: NAS-Identifier ◆ **Regular expression**: ^xyz456$ ◆ **Result specification**: RADIUS Server 1 ◆ **Comment**: To select an event 3. Click **OK**.
Chain selection	**Rule 1**: 1. Click **Add** in **Chain selection**. 2. Specify following details: ◆ **Input-Attribute**: chain_short_name ◆ **Regular expression**: ^HOTP$ ◆ **Result specification**: LDAP + HOTP ◆ **Comment**: To select chain 3. Click **OK**. **Rule 2**: 1. Click **Add** in **Chain selection**. 2. Specify following details in the respective fields: ◆ **Input-Attribute**: NAS-Identifier ◆ **Regular expression**: ^TOTP$ ◆ **Result specification**: LDAP + TOTP ◆ **Comment**: To select a chain 3. Click **OK**.

Rule	Procedure
Result specification	1. Click **Add** in **Result specification**. 2. Specify following details: ◆ **Return-Attribute**: Filter-Id ◆ **User attribute**: email ◆ **Regular expression**: . ◆ **Result specification**: email address is {email} ◆ **Comment**: To display email address of authenticated user 3. Click **OK**.

After you implement this RADIUS rules, the following are possible scenarios:

Scenario	Request Sent to the Event	Result
A user initiates authentication from openvpn1 (NAS ID: abc123)	RADIUS Server	Email address of the user is displayed on the openvpn1 RADIUS client after successful authentication.
A user initiates authentication from openvpn2 (NAS ID: xyz456)	RADIUS Server 1	Email address of the user is displayed on the openvpn2 RADIUS client after successful authentication.

11.20.5 Adding clients

You can add one or more RADIUS clients details in the **Clients** section. The defined input, event selection, chain selection, and result specification rules gets applied to the RADIUS clients.

To add a RADIUS Client perform the following steps:

1. Click Add.
2. Specify the IP address of the RADIUS Client in **IP Address**.
3. Specify the RADIUS Client name in **Name**.
4. Specify the RADIUS Client secret and confirm the secret.
5. Ensure that the RADIUS Client is set to **ON**.
6. Enable the **Require message Auth** option to require the Message-Authenticator (MA) attribute for each configured RADIUS client message.

   If this option is set to **ON** and if the RADIUS client does not send the MA attribute, then Advanced Authentication ignores any message from the RADIUS client.

   If this option is set to **OFF**, then Advanced Authentication responds to any RADIUS messages regardless of the presence of the MA attribute in the message.

   By default, this option is set to **ON** for new configurations. For existing (upgraded) configurations, this option is set to **OFF** by default.
7. Click  next to the RADIUS Client to save the details.

The **Clients** section lists all the clients of different RADIUS Events. You can map all requests from a specific client to the required RADIUS event by defining the **Event selection** rule. For more information on how to create an event selection rule, see Event selection rule.

11.21 Reporting options

In this policy, you can configure settings to delete the history about the login information of users that is recorded in the reports.

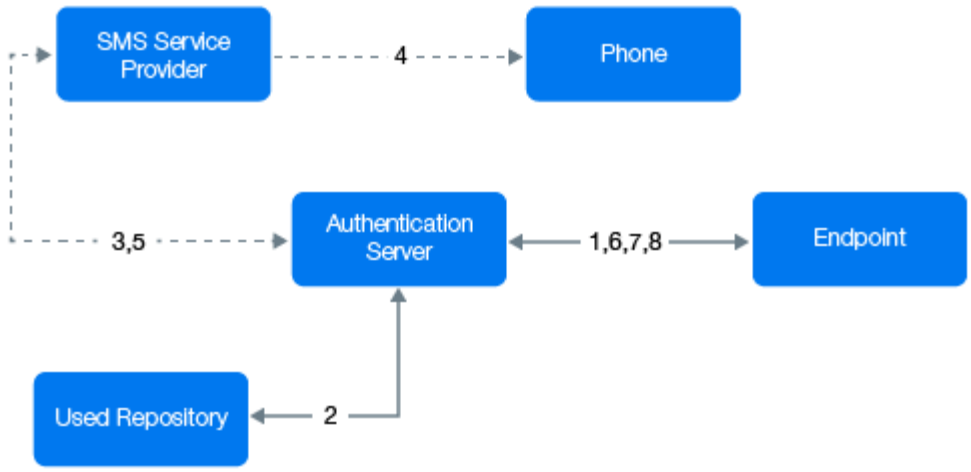
Specify a value in **History max age(days)**. The default value is 30 (days). This indicates that the history about the login information of users will be recorded from the current date to the previous 30 days. Any data before that will be deleted.

11.22 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.

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).

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

- Generic
- Twilio
- MessageBird

11.22.1 Generic

You can configure one of the following generic SMS sender manually:

- Clickatell
- SignalWire
- LOX

Clickatell

To configure Clickatell as the SMS sender perform the following steps:

1 Select **Generic** in **Sender service**.

2 Specify a **Service URL** value.

For example, `https://platform.clickatell.com/messages/http/send?apiKey=szkSkap_SqumXVb4vUfU0Q==&to=359884194544&content=Test message text\`

3 Specify **HTTP Basic Authentication Username** and **HTTP Basic Authentication Password** obtained from Clickatell.

4 Select **POST** from **HTTP request method**.

5 Select the required content type in **HTTP request content type** to send the HTTP request to the service provider. The supported options are:

- URL encoded
- JSON

6 If you want to send the HTTP request in the URL encoded type, perform the following steps:

6a Select **URL Encoded** in the **HTTP request content type**.

6b Click **Add** and create the following parameters in **HTTP request body**.

- Parameter name: **user**

  Parameter value: name of your account
- Parameter name: **to**

  Parameter value: {phone}
- Parameter name: **text**

  Parameter value: {message}
- Parameter name: **apiKey**, 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.
- Parameter name: **from**

  Parameter value: sender's phone number

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

Name: Specify a term to identify the parameter. For example, password

Value: current password that is set on the Clickatell account

For more information about the additional parameters for Clickatell, see the Clickatell documentation (https://www.clickatell.com/developers/api-documentation/http-api-request-parameters/).

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

---

7 If you want to send the HTTP request in the JSON type, perform the following steps:

7a Select **URL Encoded** in the **HTTP request content type**.

7b Enter the HTTP request in the **JSON template**.

For example, `{"to":"{{phone}}"'"message":"{{message}}"}`

where,

- `{{phone}}`: Recipients phone number
- `{{message}}`: Message body

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

For more information about the additional parameters for Clickatell, see the Clickatell documentation (https://www.clickatell.com/developers/api-documentation/http-api-request-parameters/).

SignalWire

Before you configure SignalWire as the SMS sender, ensure that you meet the following prerequisites:

- In SignalWire, create a project, choose a sub-domain (part of the sign-up process), and obtain the Direct Inward Dialing (DID) number.
- Create an API token, obtain the Project Key and Token to configure in the SMS sender policy of the Advanced Authentication Administration portal.

To configure SignalWire as the SMS sender perform the following steps:

1. Select **Generic** from **Sender service**.
2. Specify a **Service URL** value.

   For example, `https://{yourdomain}.signalwire.com/api/laml/2010-04-01/Accounts/{project key}/Messages.json`
3. Specify the Project Key (obtained from SignalWire) in **HTTP Basic Authentication Username**.
4. Specify the Token (obtained from SignalWire) in **HTTP Basic Authentication Password**.
5. Select **POST** from **HTTP request method**.
6. Select the required content type in **HTTP request content type** to send the HTTP request to the service provider. The supported options are:
   - URL encoded
   - JSON
7. If you want to send the HTTP request in the URL encoded type, perform the following steps:
   - **7a** Select **URL Encoded** in the **HTTP request content type**.
   - **7b** Click **Add** and create the following parameters in **HTTP request body**.
     - Parameter Name: **to**

       Parameter Value: {phone}
     - Parameter Name: **from**

       Parameter Value: DID number of your SignalWire project.
     - Parameter Name: **body**

       Parameter Value: {message}
8. If you want to send the HTTP request in the JSON type, perform the following steps:
   - **8a** Select **URL Encoded** in the **HTTP request content type**.
   - **8b** Enter the HTTP request in the **JSON template**.

     For example, `{"to":"{{phone}}"'"message":"{{message}}"}`

     where,
     - `{{phone}}`: Recipients phone number
     - `{{message}}`: Message body

NOTE:

- The parameters may differ for different SMS service providers. But the `{phone}` and `{message}` variables are mandatory.
- Ensure that the from phone number is in E.164 format. Number in this format starts with a plus (+) symbol and the country code.

  For example, if India based phone number is (91) 123-4567 then the E.164 formatted number is +911234567.

For more information, see SignalWire API reference (https://docs.signalwire.com/topics/laml-api/#api-reference).

LOX

To configure LOX as the SMS sender perform the following steps:

1. Select **Generic** from **Sender service**.
2. Specify a Service **URL value**.

   For example, `https://www.lox24.eu/API/httpsms.php?konto=1&password=APIV1Key&service=5\`
3. Specify the Project Key (obtained from LOX) in **HTTP Basic Authentication**.
4. Specify the Token (obtained from LOX) in **HTTP Basic Authentication**.
5. **GET** from **HTTP request method**.
6. Select the required content type in **HTTP request content type** to send the HTTP request to the service provider. The supported options are:
   - URL encoded
   - JSON
7. If you want to send the HTTP request in the URL encoded type, perform the following steps:
   - **7a** Select **URL Encoded** in the **HTTP request content type**.
   - **7b** Click **Add** and create the following parameters in **HTTP request body**.
     - Parameter name: **user**

       Parameter value: name of your account
     - Parameter name: to

       Parameter value: {phone}
     - Parameter name: **text**

       Parameter value: {message}
     - Parameter name: **from**

       Parameter value: sender's phone number
   - **7c** Click **Save** icon  after entering **Parameter name** and **Parameter value** each time.

7d Click **Add secure** and create the following parameters in **HTTP request body**.

- Name: **password**

  Value: current password that is set on the account.

8 If you want to send the HTTP request in the JSON type, perform the following steps:

8a Select **URL Encoded** in the **HTTP request content type**.

8b Enter the HTTP request in the **JSON template**.

For example, `{"to":"{{phone}}"'"message":"{{message}}"}`

where,

- `{{phone}}`: Recipients phone number
- `{{message}}`: Message body

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

For more information about the additional parameters for LOX, see the LOX documentation (http://www.lox24.eu/download/api/LOX24-SMS-API-en.pdf).

11.22.2 Twilio

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 **Authentication token**: In Twilio, the Account SID acts as a username and the Authentication Token acts as a password.

  NOTE: After you save the configuration, Authentication token is not displayed even in the masked form.

  NOTE: If the Authentication token is not visible then the configuration has been saved. Specify the Authentication token again before sending a test message as the **Test** button reads the message from the UI. The real messaging service reads the message from the Advanced Authentication database.

- **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.

  NOTE: When the **Use Twilio Verify (Early Access)** option is set to **ON**, the **Use Copilot** option is hidden.

- **Use Twilio Verify (Early Access)**: This option enables you to utilize the Twilio Verify service for user verification. Twilio Verify is a turnkey API service that sends an OTP to users through multiple channels such as SMS, email, and WhatsApp to enhance user account security and prevent fraud. It also helps in compliance by avoiding carrier and government registration requirements and optimizes messages routing using premium telephony routes to prioritize deliverability and speed.

  For more information on Twilio Verify and its features, see Twilio Verify (https://www.twilio.com/docs/verify)

  - **Messaging Service SID**: Specify the unique identifier assigned by Twilio to a Messaging Service that you created within your Twilio account. Instead of using a specific phone number, the **Messaging Service SID** points to a collection of settings, rules, and phone numbers that are part of that service. The SID enables Twilio to route messages according to your service configurations, such as sender IDs and message delivery preferences.
  - **Default Country Code**: This is the default country code Twilio uses when processing a phone number that does not include a country code. If you provide a local phone number (i.e., without the country code), Twilio will automatically prepend the default country code.

    For example, if you set the default country code to +1 (the U.S. country code) and the user's number in the repository is 555-4444, then Twilio will interpret the number as +1555-4444.

IMPORTANT: This option is only available as a technical preview for testing and providing feedback. The technical preview features are not fully supported and may change significantly based on your feedback and ongoing development. We recommend that you try these features and provide your feedback to aafeedback@opentext.com.

It is recommended to deploy or configure the technical preview features only in the staging environment.

- **Sender phone**: This is the from phone number received from Twilio. Specify the Twilio phone number that you own and prefix the country code and backslash (\).

  For example, 91\9191919191

NOTE: When the **Use Twilio Verify (Early Access)** option is set to **ON**, the **Sender phone** option is hidden.

---

3 (Optional) To configure the Subaccounts, perform the following:

NOTE: Twilio account supports multiple subaccounts that helps to segregate the usage based on geographic location, phone numbers, customers, or any other category. Subaccounts are associated with main Twilio account and share the balance. However, each subaccount has unique Account SID and Auth Token to determine the usage.

For more information, see Twilio Subaccounts (https://support.twilio.com/hc/en-us/articles/360011132374-Getting-Started-with-Twilio-Accounts-and-Subaccounts).

3a Click **Add**.

3b Specify the following details:

- **Country Dialing Code Filter**: This code helps to determine which subaccount needs to used to send an SMS OTP message to a user.

  For example, the administrator can configure a subaccount that delivers SMS OTP messages to all users in India using the code +91 as the Country Dialing Code Filter. So that Advanced Authentication server automatically uses a specify subaccount to send all messages to users requesting from India.

- **Subaccount SID**: 34 digits unique String Identifier (SID) of the subaccount to recognize the resource.
- **Subaccount Auth Token**: Authentication Token to verify the user's identity and indicates the level of access.
- **Sender Phone**: From phone number that is displayed on recipients phone.

3c Click Save icon .

11.22.3 MessageBird

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.

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.

11.23 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 **Authentication token**: In Twilio, the Account SID acts as a username, and the Authentication Token acts as a password.

- **Sender phone**: The phone number of the sender.
- **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.

NOTE: Only a top administrator can set and enforce the Service URL setting. Tenants cannot edit this setting in the tenant administrator console.

---

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

2a Specify the phone number in **Phone** to which you want to send the Voice OTP.

2b Specify a message to be sent to the phone in **Message**.

2c Click **Send test message!**.

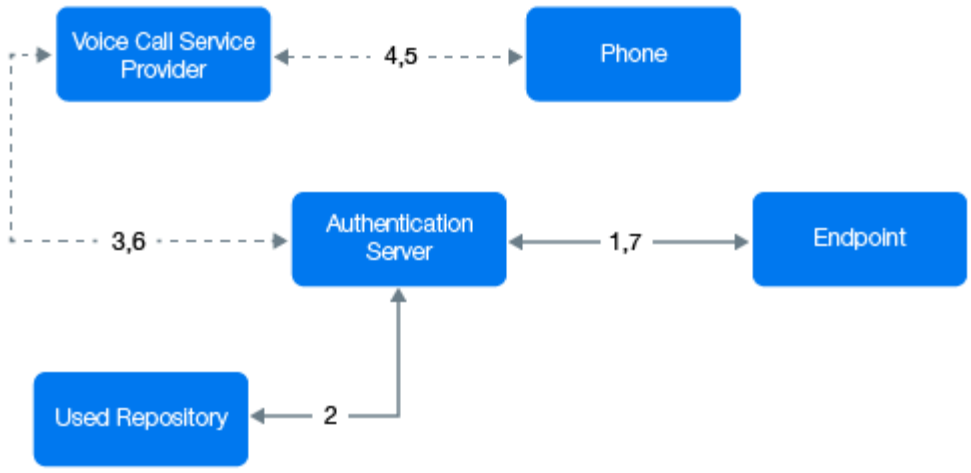
3 Click **Save**.

Real messaging uses async sender. Ensure that you have configured a chain with the **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).

11.24 Web Authentication

This policy replaces the **SAML 2.0 options** policy. The Web Authentication policy allows you to configure the following settings:

- Configuring the identity provider
- Downloading the identity provider SAML metadata
- Configuring timeout
- Enabling the client event selection
- Enabling the client chain selection
- Customizing messages and authentication method names for the web authentication events

11.24.1 Configuring the identity provider

The **Identity provider URL** displays the default Public external URL. Also, the tenant_base URL if configured in the Public external URLs policy. If the Identity provider URL is set to default Public external URL, then the users must prefix `TENANT\repository\` to the username during login. If you select the URL in the `https://tenantName.tenant_base/` format, then the users can access the Advanced Authentication portals using the short username without prefixing the `TENANT\repository\`. The `tenant_base` represents the domain name. The domain name is shared among the tenants to generate a unique URL for each tenant.

For example, consider top tenant URL is `caf.aacloud.com`, here the domain base is `aacloud.com`. If the name of a new tenant is `cyberres` then the tenant URL becomes `cyberres.aacloud.com`.

11.24.2 Downloading the identity provider SAML metadata

Click IdP SAML 2.0 Metadata to download SAML 2.0 metadata. You can download the SAML 2.0 metadata file only when the Identity Provider's URL is configured. The downloaded SAML 2.0 metadata file is used to configure the Service Provider.

For more information about SAML 2.0, see SAML 2.0.

11.24.3 Configuring timeout

Specify the following details to configure timeouts:

1. **Session Timeout:** Specify the time in seconds. By default, this value is set to 1200 seconds. This is the timeout value for authenticating to the Web Authentication session. If the session is idle for more than the specified time, then the session expires and the user must authenticate again before any action which requires an authenticated session. This timeout value is applicable for OAuth2 / OpenID Connect and SAML events.

   NOTE: Advanced Authentication user enrollment uses Web Authentication for authenticating users. After the authentication, the OSP session is not in use and Advanced Authentication manages its sessions. An OAuth2 / OpenID Connect application might use the Web Authentication session.

   ---

2. **Authorization Code Timeout:** Specify the time in seconds. By default, this value is set to 120 seconds. This timeout value indicates how long the authorization code is valid. The request for an Access Token or an ID Token fails if the Authorization Code has expired and is no longer valid. The Authorization code becomes invalid if the client does not request for Token ID from the server within the specified time.

   For security reasons, some OAuth2 / OpenID Connect code flow schemes require that first an Authorization Code be requested. The Authorization Code is then used to request an Access Token and ID Token.

3. **Access Token Timeout**: Specify the time in seconds till when the access token is valid. By default, this value is set to 120 seconds. Once the token expires, a new token is required before accessing the protected resources. The application might create a new token by using a Refresh Token and the client secret, or else the user is required to authenticate again.

4. **Refresh Token Timeout**: Specify the time in seconds till when the token is valid. Once the token expires it can no longer be used to create a new Access Token. By default, this value is set to 2592000 seconds.

5. **Public Refresh Token Timeout**: This timeout value is for refreshing token timeout for public clients. When there are two client types, private and public. By default, this value is set to 3600 seconds.

6. **Session Token Revocation Timeout**: Specify the timeout value till when the session-based refresh token revocation entries are retained. Retained entries are removed when the session is properly logged out or after the refresh token expires. By default, this value is set to 172800 seconds.

11.24.4 Enabling the client event selection

The **Enable client event selection** option is set to **OFF** by default and the third-party service provider cannot select a preferred event for any client such as Windows Client, Mac OS X Client, and Linux PAM Client workstation. You can set this option is set to **ON** to allow third-party service providers to select a preferred event that is configured in the `authcfg.xml` file that user can access post authentication.

The syntax to select a specific event in the `authcfg.xml` file is as follows:`'AuthnContextClassRef' => array( 'AuthnContextClassRef' => 'urn:uuid:519a6c73-f092-43d3-ab11-8d789ebc2f79?=internal.osp.oidp.aa.event-name=<event name>')`

NOTE: If you configure an incorrect event name and a user tries to authenticate to the event, an error message `authentication failed` appears.

11.24.5 Enabling the client chain selection

The **Enable client chain selection** option is set to **OFF** by default and the third-party service provider cannot select a preferred chain for any client such as Windows Client, Mac OS X Client, and Linux PAM Client workstation. You can set this option is set to **ON** to allow third-party service providers to select a preferred chain that is configured in the `authcfg.xml` file that user can use during authentication.

The syntax to select a specific chain in the `authcfg.xml` file is as follows:`'AuthnContextClassRef' => array( 'AuthnContextClassRef' => 'urn:uuid:519a6c73-f092-43d3-ab11-8d789ebc2f79?=internal.osp.oidp.aa.chain-name=<chain name>')`

NOTE: If you configure an incorrect chain and a user tries to authenticate by using that chain, an error message `authentication failed` appears.

11.24.6 Customizing messages and authentication method names for the web authentication events

You can customize the messages and authentication methods name for the Web Authentication events in the Custom Messages policy. By default, **Use Custom Messages** is set to **OFF**. Set **Use Custom Messages** to **ON** to enable using the custom messages for the OAuth, SAML 2.0, or Open ID Connect events. You must customize the messages in the "Custom messages" policy.

NOTE: If you modify non-English messages or method name in the **Custom Messages** policy With **Use Custom Messages** is set to **OFF**, then customized messages does not reflect on the Web Authentication events page.

11.25 Windows Credential Providers

NOTE: This Windows Credential Provider policy is applicable only for Windows Client.

In this policy, you can configure the settings that enables the third-party credential provider to confirm the identity of a user and authentication.

When using Advanced Authentication on windows workstation where Windows client is installed, the client functions as the Credential Provider during user logon. This policy allows you to enforce the use of third-party Credential Provider for ensuring the user's identity during logon process and grant access.

Additionally, users can enable or add third-party Credential Provider by specifying the `allowedproviders` parameter in the `config.properties` file of the windows client.

To configure the Windows Credential Providers policy, perform the following steps:

1. Set the **Enable** option to **ON**.
2. Specify the Credential Provider Data name in **CP Name**.

   To get the **CP Name**:

   2a Navigate to the path `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Authentication\Credential Providers`

   2b Search the **Data** name of the required Credential Provider.

   For example, the Data name of Sophos SafeGuard is NPProvider.
3. Specify the class ID of provider in **CP Globally Unique Identifier (GUID)**.

   To get the class ID:

   3a Navigate to the path `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Authentication\Credential Providers`

   3b Search the class id of the required Credential Provider.

   For example, the class ID of Sophos SafeGuard is {5CDFA681-61C8-423d-999E-32EA10C5F7ED}.
4. Click **Save**.

12 Configuring the server options

Perform the following configurations to configure the Advanced Authentication server settings:

- Section 12.1, “Generating OSP keystores,” on page 251

12.1 Generating OSP keystores

You can generate the signing and encrypting certificates for the SAML federation based third-party integrations. By default, the Advanced Authentication server has a signing and encrypting certificates. You can use the default certificates or generate new certificates based on your requirements. Generating new certificates delete the existing certificates and replace them with new certificates.

NOTE: The existing SAML2 federations disrupt if you generate new OSP Keystores. Therefore, you must update the existing SAML2 federations with the new keys to re-establish the trust.

---

13 Adding a Report

Report provides you pictorial representation of collected data. You can examine data in different combinations, display report in easy-to-understand graphs, track data at different time intervals and export the report in JSON and CSV formats to share the result with others. With reports, you can track all logins (failed or successful), users' enrollment status, authentication methods used for specific event, license information, number of active users and so on.

You can add a report with specific report type as described in Table 13-1.

Table 13-1 *Report types*

Report Type	Description	Available Attributes
Pie chart	This report displays the information collected on a specific parameter and represents information in the Pie chart format. You can sort the parameter in ascending and descending order.	◆ **Name**: Title of the report. ◆ **Relative Time Interval**: Set this option to **ON** to select a specific time interval from the **Relative Interval**. Set this option to **OFF** to select preferred From and To dates from the **Date range**. ◆ **Event Type**: Types of events to display in the report. Options available are **All logon events**, **Failed logon events** and **Successful logon events**. ◆ **Size**: Number of records to filter in the report. ◆ **Order**: Sorting order of selected parameter in the **Field**. Options available are **Ascending** and **Descending**. ◆ **Field**: The parameter on which the data is collected to display in the report. Options available are **Event Name**, **Chain Name**, **Method Name**, **Endpoint Name** and so on. ◆ **Users**: To filter records of specific user from directory. ◆ **Events**: To filter records of specific event. ◆ **Chains**: To filter records of specific chain.

Report Type	Description	Available Attributes
Stacked chart	This report displays a stacked bar chart that classifies and compares different categories of Field 1 and 2 parameters to track the maximum and minimum number of logons. X-axis represents categories of the Field 2 parameter. Y-axis represents logon count. Segments in each vertical bar represents categories of Field 1 parameter. Different colors are used to depict different categories and label for each category is displayed in upper-right corner of the report.	◆ **Name**: Title of the report. ◆ **Relative Time Interval**: Set this option to **ON** to select a specific time interval from the **Relative Interval**. Set this option to **OFF** to select preferred From and To dates from the **Date range**. ◆ **Event Type**: Types of events to display in the report. Options available are **All logon events**, **Failed logon events** and **Successful logon events**. ◆ **Field 1**: The parameter to represent on X-axis of the report. Options available are **Event Name**, **Chain Name**, **Method Name**, **Endpoint Name** and so on. ◆ **Size 1**: Number of records to display on the X-axis. ◆ **Order 1**: To sort the parameter selected in the **Field 1**. Options available are **Ascending** and **Descending**. ◆ **Field 2**: The parameter to represent on Y-axis of the report. Options available are **Event Name**, **Chain Name**, **Method Name**, **Endpoint Name** and so on. ◆ **Size 2**: Number of records to display on the Y-axis. ◆ **Order 2**: To sort the parameter selected in the **Field 2**. Options available are **Ascending** and **Descending**. ◆ **Users**: To filter records of specific user from directory. ◆ **Events**: To filter records of specific event. ◆ **Chains**: To filter records of specific chain.
Activity stream	This report displays information about user, tenant, chain, method used for authentication, and the result.	◆ **Name**: Title of the report. ◆ **Relative Time Interval**: Set this option to **ON**, select a specific time interval from the **Relative Interval**. Set this option to **OFF**, select preferred From and To dates from the **Date range**. ◆ **Event Type**: Types of events to display in the report. Options available are **All logon events**, **Failed logon events** and **Successful logon events**. ◆ **Users**: To filter records of specific user from directory. ◆ **Events**: To filter records of specific event. ◆ **Chains**: To filter records of specific chain.

Report Type	Description	Available Attributes
Enroll activity stream	This report displays information about enrolled users: last log on time, tenant, user, method used for authentication, and event type.	◆ **Name**: Title of the report. ◆ **Relative Time Interval**: Set this option to **ON** to select a specific time interval from the **Relative Interval**. Set this option to **OFF** to select preferred From and To dates from the **Date range**. ◆ **Users**: To filter records of specific user from directory.
Users	This report displays information about the enrolled users: tenant name, user name, enrollment status and last log on time.	◆ **Name**: Title of the report. ◆ **Relative Time Interval**: Set this option to **ON** to select a specific time interval from the **Relative Interval**. Set this option to **OFF** to select preferred From and To dates from the **Date range**.
Authenticators	This report displays information about the enrolled authenticators: tenant name, user name, event category, method, comment and owner of the account.	◆ **Name**: Title of the report. ◆ **Relative Time Interval**: Set this option to **ON** to select a specific time interval from the **Relative Interval**. Set this option to **OFF** to select preferred From and To dates from the **Date range**.
Event count line chart	This report tracks and displays logon count of all events in the appliance. X-axis represents time and Y-axis represents logon count. Each data point on the chart represents numbers of user logged on at a specific time. All the data points are plotted and connected with a line to track the maximum and minimum number of logons.	◆ **Name**: Title of the report. ◆ **Relative Time Interval**: Set this option to **ON** to select a specific time interval from the **Relative Interval**. Set this option to **OFF** to select preferred From and To dates from the **Date range**. ◆ **Event Type**: Types of events to display in the report. Options available are **All logon events**, **Failed logon events** and **Successful logon events**. ◆ **Interval**: Regular interval to track the data point on the chart. ◆ **Users**: To filter records of specific user from directory. ◆ **Events**: To filter records of specific event. ◆ **Chains**: To filter records of specific chain.

Report Type	Description	Available Attributes
Event count line chart group by field	This report tracks and displays logon count of specific parameter. X-axis represents time and Y-axis represents logon count. Data points of different colors represent specific category of the selected parameter. The label for each category is displayed in upper-right corner of the widget. All the data points are plotted and connected with a line to track the maximum and minimum number of logons.	◆ **Name**: Title of the report. ◆ **Relative Time Interval**: Set this option to **ON** to select a specific time interval from the **Relative Interval**. Set this option to **OFF** to select preferred From and To dates from the **Date range**. ◆ **Event Type**: Types of events to display in the report. Options available are **All logon events**, **Failed logon events** and **Successful logon events**. ◆ **Interval**: Regular interval to track the data point on the chart. ◆ **Size**: Number of records to filter in the report. ◆ **Order**: Sorting order of the parameter selected in the **Field**. Options available are **Ascending** and **Descending**. ◆ **Field**: The parameter on which the data is collected to display in the report. Options available are **Event Name**, **Chain Name**, **Method Name**, **Endpoint Name** and so on. ◆ **Users**: To filter records of specific user from directory. ◆ **Events**: To filter records of specific event. ◆ **Chains**: To filter records of specific chain.

Report Type	Description	Available Attributes
Distinct events count line chart	This report tracks and displays distinct count of all categories in the selected parameter (**Distinct values by field**). X-axis represents time and Y-axis represents distinct logon count. Each data point on the chart represents unique logon count at a specific time. All the data points are plotted and connected with a line to track the maximum and minimum number of distinct logons.	◆ **Name**: Title of the report. ◆ **Relative Time Interval**: Set this option to **ON**, select a specific time interval from the **Relative Interval**. Set this option to **OFF**, select preferred From and To dates from the **Date range**. ◆ **Event Type**: Events to display in the report. Options available are **All logon events**, **Failed logon events** and **Successful logon events**. ◆ **Interval**: Regular interval to track the data point on the chart. ◆ **Distinct values by field**: The parameter on which the data is collected to display in the report. Options available are **Event Name**, **Chain Name**, **Method Name**, **Endpoint Name** and so on. ◆ **Size**: Number of records to filter in the report. ◆ **Order**: Sorting order of the parameter selected in the **Field**. Options available are **Ascending** and **Descending**. ◆ **Users**: To filter records of specific user from directory. ◆ **Events**: To filter records of specific event. ◆ **Chains**: To filter records of specific chain.

Report Type	Description	Available Attributes
Distinct events count line chart group by fields	This report displays and classifies distinct logon count of each event. X-axis represents time and Y-axis represents distinct logon count. Each data point on the chart represents unique logon count of particular event at a specific time. All the data points are plotted and connected with a line to track the maximum and minimum number of distinct logons to particular event.	◆ **Name**: Title of the report. ◆ **Relative Time Interval**: Set this option to **ON**, select a specific time interval from the **Relative Interval**. Set this option to **OFF**, select preferred From and To dates from the **Date range**. ◆ **Event Type**: Events to display in the report. Options available are **All logon events**, **Failed logon events** and **Successful logon events**. ◆ **Interval**: Regular interval to track the data point on the chart. ◆ **Size**: Number of records to filter in the report. ◆ **Order**: Sorting order of the parameter selected in the **Field**. Options available are **Ascending** and **Descending**. ◆ **Field**: The parameter on which the data is collected to display in the report. Options available are **Event Name**, **Chain Name**, **Method Name**, **Endpoint Name** and so on. ◆ **Distinct values by field**: The parameter on which the data is collected to display in the report. Options available are **Event Name**, **Chain Name**, **Method Name**, **Endpoint Name** and so on. ◆ **Users**: To filter records of specific user from directory. ◆ **Events**: To filter records of specific event. ◆ **Chains**: To filter records of specific chain.

Report Type	Description	Available Attributes
Billing-Unique user per month **IMPORTANT:** The Billing-Unique user per month reports is available only in Advanced Authentication as a Service (SaaS) version.	This is a SaaS report. This report displays the unique user logon count in the selected period. X- axis represents time and the Y-axis represents the unique user logon count.	◆ **Name**: Title of the report. ◆ **Billing Period**: Select the billing period from the following options: ◆ **This Month**: To get the current moth logon count. ◆ **Last Month**: To get the logon count of last month. ◆ **Last 7-days**: To get the last seven days logon count of a specific tenant. ◆ **Monthly History**: To get the logon count of each month. ◆ **Counting Method**: Select the counting method from the following options: ◆ **Estimate count (Quicker)**: To get the approximate logon count. ◆ **Full Count**: To get the exact logon count.
User validations	This report displays information about the user validation result for the HANIS Face and HANIS Fingerprint methods.	◆ **Name**: Title of the report. ◆ **Relative Time Interval**: Set this option to **ON**, select a specific time interval from the **Relative Interval**. Set this option to **OFF**, select preferred From and To dates from the **Date range**. ◆ **Validation type**: To filter records of the validation type: HANIS Face, HANIS Fingerprint, or both. ◆ **Interval**: Regular interval to track the data point on the chart. ◆ **Users**: To filter records of specific user whose details are available on the directory or not. Usage of partial username input is not supported for users who do not exist in the corporate directory. The search using incomplete username input does not fetch results. ◆ **Validation result**: To filter records of specific result such as Successful, Failed, and All validations.

Following are the generic steps to add a custom report:

1. Click **Reports** in the Administration portal.
2. Click **Add**.
3. Specify the report title in the **Name**.

4. Select the preferred **Report type**. Options available are:
   - Pie chart
   - Stacked chart
   - Activity stream
   - Enroll activity stream
   - Users
   - Authenticators
   - Licenses
   - Servers
   - Events count line chart
   - Events count line chart grouped by field
   - Distinct events count line chart
   - Distinct events count line chart grouped by field
   - User validations
5. When the **Relative time interval** is set to **ON**, the **Relative Interval** is displayed to select a specific time interval. When set to **OFF**, the date range is displayed to select preferred From and To dates.
6. Select the preferred **Event type**. Options available are **All logon events**, **Failed logon events**, and **Successful logon events**.
7. Select number of records from the **Size** to display in the report.
8. Select sorting order from the **Order**. Options available are **Ascending** or **Descending**.
9. Select the preferred parameter from the **Field**. Based on the selected parameter, the data is collected to display on the report. Options available are **Event Name**, **Chain Name**, **Method Name**, **Endpoint Name** and so on.
10. Specify and select the preferred domain joined user from the **Users** to filter records in the report.
11. Specify and select the preferred event from the **Events** to filter records in the report.
12. Specify and select the preferred chain from the **Chains** to filter records in the report.
13. Click **Save**.
14. Click **Reload** to generate and display the report based on the selected values.

14 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.

  These methods are enrolled automatically if the chains containing them are assigned to any event and a user performs the first authentication using the method.

- **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.

15 Scripts option

Scripts Option allows you to generate a script to install your RADIUS Agent.

- Section 15.1, “Generating RADIUS script,” on page 263

15.1 Generating RADIUS script

Your can generate a RADIUS script to install RADIUS Agents. Before generating the RADIUS script, ensure that the following requirements are met:

- All required RADIUS Clients are added in **Policies > RADIUS Options**. For more information, see Adding clients.
- At least one chain is assigned to any RADIUS Server event.

To generate a RADIUS Script, perform the following step:

1. Click **Scripts Option > RADIUS Script**.
2. Select **Radius** from **Script**.
3. Select **Linux** from **Platform**.
4. Specify the expiration time (in hours) of generated script in **Expiration time (hours)**.

   NOTE: Due to security reasons, the expiration time should not be more than 48 hours.
5. Specify the docker image name (optional) and path to save the generated script in **Docker Image**.
6. Click **Generate Script**.

NOTE: The RADIUS script contains the RADIUS Client information. Therefore, later, if you need to add a new RADIUS Client for RADIUS Agent or modify the existing RADIUS Client, you need to uninstall the RADIUS Agent and re-install the RADIUS Agent with the new script where the new RADIUS Clients are configured.

For more information about starting, stopping, and restarting, see Starting, stopping, restating RADIUS Agent.

III Configuring risk settings

Advanced Authentication uses Risk Service to assess the risk based on the contextual information associated with an access attempt. You can configure this capability through Risk Settings.

IMPORTANT: To configure Risk Settings and use Risk Service, you must first purchase and add the license for it.

IMPORTANT: In the cluster environment, Risk Service is available on the Global Master Server only. The Web Server and other nodes transfer the request to the Global Master Server to evaluate the risk level. Therefore, it is required to use a valid CA issued certificate instead of the self-signed certificate. In case, the self-signed certificate is in use, then other nodes cannot establish a connection with the Global Master Server.

NOTE: The External Parameters rule and Behavioral Analytics are available in Advanced Authentication 6.3 Service Pack 3 and later versions.

For information about...	See
Risk Service, its process, and its benefits	Introduction to Risk Service (https://www.netiq.com/documentation/risk-service-2.0/admin/data/ovrvw_risk_srv.html) in the Risk Service Guide (https://www.netiq.com/documentation/risk-service-2.0/admin/data/bookinfo.html).
How to configure Risk Service	Configuring Risk Service (https://www.netiq.com/documentation/risk-service-2.0/admin/data/configuring-risk-service.html) in the Risk Service Guide (https://www.netiq.com/documentation/risk-service-2.0/admin/data/bookinfo.html).

- Chapter 16, “Configuring Risk Service,” on page 267
- Chapter 17, “Understanding how Risk Service works through scenarios,” on page 269
- Chapter 18, “Troubleshooting Risk Service configuration,” on page 275

16 Configuring Risk Service

To see general Risk Service configuration details, see Configuring Risk Service (https://www.netiq.com/documentation/risk-service-2.0/admin/data/configuring-risk-service.html) in the Risk Service Guide (https://www.netiq.com/documentation/risk-service-2.0/admin/data/bookinfo.html).

- Section 16.1, “Monitoring risk audit logs,” on page 267

16.1 Monitoring risk audit logs

Risk logs include information about the risk service events. The logs message is displayed in the following CEF format:

```
Date host CEF:Version|Device Vendor|Device Product|Device Version|Device
Event Class ID|Name|Severity|[Extension]
```

The Extension part of the message displays additional details associated with an audit event. Extension can include the following fields:

- Custom string label: Indicates the name of the audit field.
- Custom string: Indicates the value of custom string label.
- Custom number label: Indicates the name of the audit field.
- Custom number: Indicates the value of respective custom number label.

EventID	Name	Severity	Example
receivedRequest	Received request at Risk Service	LOW	INFO RiskService_collector CEF:0|NetIQ|Risk Service|1.0|receivedRequest|Received request at Risk Service|LOW| suid=123 cs1Label=correlationID cs1=abcdef_123456 cs2Label=containerID cs2=f6811eb7c2e2 cs3Label=tenantID cs3=tenant_1 cs4Label=policyID cs4=Demo_Risk Policy cn1Label=mode cn1=0 msg=Request received at the Risk service for risk evaluation

EventID	Name	Severity	Example
successfulRiskEvaluated	Successful Response sent from Risk	LOW	INFO RiskService_collector CEF:0|NetIQ|Risk Service|1.0|successfulRiskEvaluated|Successful Response sent from Risk Service|LOW| suid=123 cs1Label=correlationID cs1=abcdef_123456 cs2Label=containerID cs2=f6811eb7c2e2 cs3Label=tenantID cs3=tenant_1 cs4Label=policyID cs4=RPH cn1Label=mode cn1=0 cn2Label=riskscore cn2=100 cs5Label=risklevel cs5=Medium msg= Response of the risk evaluation request sent successfully
riskResponseFailure	Risk Service response failed	HIGH	INFO RiskService_collector CEF:0|NetIQ|Risk Service|1.0|riskResponseFailure|Risk Service response failed|HIGH| cs1Label=correlationID cs1=abcdef_123456 cs2Label=containerID cs2=f6811eb7c2e2 cs3Label=tenantID cs3=tenant_1 cs4Label=policyID cs4=Demo_Risk Policy cn1Label=mode msg=Failed to provide the response of the risk evaluation request at Risk Service : {"error":"Policy not found for tenant."}
configurationChanged	Risk configuration has been modified	LOW	INFO RiskService_ui CEF:0|NetIQ|Risk Service|1.0|configurationChanged|Risk configuration has been modified|LOW| suid=admin cs1Label=correlationID cs1=2660c5a5-60b8-44b8-aafe-589a77bc7561 cs2Label=containerID cs2=e272e8f5f6ca cs3Label=tenantID cs3=tenant_1 cs4Label=policyID cn1Label=mode cs5Label=configName cs5=1574318009646 cs6Label=configType cs6=RISKPOLICY cs7Label=action cs7=MODIFY msg=Risk policy updated

17 Understanding how Risk Service works through scenarios

The following example scenarios describe how to use Risk Service in Advanced Authentication:

- Section 17.1, “Assessing risks based on the IP Address,” on page 269
- Section 17.2, “Allowing employees to access the human resources Portal outside the corporate network,” on page 271

17.1 Assessing risks based on the IP Address

Your organization wants to allow its employees to access the Payroll portal only from the corporate network.

For this requirement, you need to perform the following tasks:

1. Configure a risk policy with IP Address Rule.
2. Configure a chain for the low risk level.
3. Configure or modify the event for the payroll portal and map the risk policy and the chain to this event.

NOTE: If you do not configure a chain for the high risk level, no chain is prompted to the user for authentication in the high-risk scenarios. The access is denied in such a case.

Configure a risk policy

1 Click **Risk Settings** > **Create a Risk Policy** icon.

2 Specify the following details:

- **Policy Name**: Specify the name. For example, `Risk-Service-Internal-Network`.
- **Description**: Specify the purpose of this policy.

3 Configure IP Address Rule as follows:

3a Click **Add Rule**.

3b Specify the rule name and the description.

3c Select **IP Address Rule** from **Choose a Rule Type**.

3d Select **Is** from **Allow if IP address in the list**.

3e Select **IP address range** in **Manually enter the Data source**.

3f Specify the range of the IP address.

For example, 10.0.0.0 to 10.255.255.255

3g Click **Save**.

4 Set the green slider to 0 to indicate the low risk level.

5 Click **Save**.

Configure a chain

1 Click **Chains** > **Add**.

2 Specify a name for the chain in **Name**. For example, `LowRisk`.

3 Set **Is enabled** to **ON**.

4 Select methods that you want to add to the chain in **Methods**. For example, Password.

5 Specify the groups that will use the authentication chain in **Roles and Groups**.

6 Expand **Risk Settings** by clicking **+**.

7 In **Minimum Risk Level,** select **Low**.

8 Click **Save**.

For more information about chains, see Chapter 8, "Creating a chain," on page 169.

Configure or modify the event for the payroll portal

1 To create a new event:

1a Click **Events** > **Add**.

1b Specify a name for the event.

1c Set **Is enabled** to **ON**.

1d Select the type in **Event type**. For example, `Generic`.

1e Select the `LowRisk` chain that you created in Configure a chain.

1f In **Risk Policy**, select the `Risk-Service-Internal-Network` policy.

1g Click **Save**.

2 To modify an existing event:

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

2b Select the `LowRisk` chain that you created in Configure a chain.

2c In **Risk Policy**, select the `Risk-Service-Internal-Network` policy.

For more information about creating and editing an event, see Configuring events.

After you implement this risk policy, the following are possible scenarios:

Scenario	Risk Level	Result
An employee accesses the Payroll portal in the corporate network.	Low	The user is required to authenticate using the `LowRisk` chain.
An employee accesses the Payroll outside the corporate network. IP Address Rule is failed.	High	No chain is configured for the high risk. So, access is denied.

17.2 Allowing employees to access the human resources Portal outside the corporate network

Inside the corporate network and within business hours, all employees can access the Human Resources (HR) portal using their password.

You want to secure the HR portal when it is accessed beyond business hours and from an external network.

To meet this requirement, you need to perform the following tasks:

1. Configure a risk policy with IP Address Rule and User Time of Login Rule.
2. Configure chains for low risk and medium risk levels.
3. Configure or modify an event for the HR portal and map the risk policy and chains to this event.

NOTE: If you do not configure a chain for the high risk level, no chain is prompted to the user for authentication in the high-risk scenarios. The access is denied in such a case.

Configure a risk policy

1 Click **Risk Settings** > **Create a Risk Policy** icon.

2 Specify the following details:

- **Policy Name**: Specify the name. For example, `Risk-Service-Employees-Access`.
- **Description**: Specify the purpose of this policy.

3 Configure **IP Address Rule** and **User Time of Login Rule** in the same sequence as follows. The rules are executed in the top to bottom sequence.

Rule	Configuration Steps
IP Address Rule	1. Click **Add Rule**. 2. Specify the rule name and the description. 3. Select **IP Address Rule** from **Choose a Rule Type**. 4. Select **Is** from **Allow if IP address in the list**. 5. Select **IP address range** in **Manually enter the Data source**. 6. Specify the range of the IP address. For example, 10.0.0.0 to 10.255.255.255 7. Click **Save**.
User Time of Login Rule	1. Click **Add Rule**. 2. Specify the rule name and the description. 3. Select **User Time of Login Rule** from **Choose a Rule Type**. 4. Select **Is** from **User time of login**. 5. Select the date range from **Monday** to **Friday**. 6. Select the time range from **9:00 AM** to **6:00 PM**. 7. Click **Save**.

4 Set up the risk levels:

- Move the blue slider to 1 to indicate that if one rule fails, the risk is medium.
- Move the green slider to 0 to indicate if no rules fail, the risk is low.
- If both rules fail, then the risk is high.

5 Click **Save**.

Configure chains

1 Create the following chains:

Chain	Steps
For the low risk level	1. Click **Chains** > **Add**. 2. Specify a name for the chain in **Name**. For example, `LowRisk`. 3. Specify a **Short name**. 4. Set **Is enabled** to **ON** to enable the chain. 5. Select **Methods** you want to add to the chain. For example, Password. 6. Specify the groups that will use the authentication chain in **Roles and Groups**. 7. Expand **Risk Settings** by clicking **+**. 8. In **Minimum Risk Level,** select **Low**. 9. Click **Save**.
For the medium risk level	1. Click **Chains** > **Add**. 2. Specify a name for the chain in **Name**. For example, `MediumRisk`. 3. Specify a **Short name**. 4. Set **Is enabled** to **ON** to enable the chain. 5. Select **Methods** you want to add to the chain. For example, Password and SMS OTP. 6. Specify the groups that will use the authentication chain in **Roles and Groups**. 7. Expand **Risk Settings** by clicking **+**. 8. In **Minimum Risk Level,** select **Medium**. 9. Click **Save**.

For more information about chains, see Chapter 8, “Creating a chain,” on page 169.

2 Click **Save**.

Configure or modify an event for the HR portal

1 To create a new event:

1a Click **Events** > **Add**.

1b Specify a name for the event.

1c Set **Is enabled** to **ON**.

1d Select the type in **Event type**. For example, `Generic`.

1e Select `MediumRisk` and `LowRisk` chains that you created in “Configure chains” on page 272.

1f In **Risk Policy**, select the `Risk-Service-Employees-Access` policy.

1g Click **Save**.

2 **To modify an existing event:**

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

2b Select `MediumRisk` and `LowRisk` chains that you created in "Configure chains" on page 272.

2c In **Risk Policy**, select the `Risk-Service-Employees-Access` policy.

For more information about creating and editing an event, see Chapter 9, "Configuring events," on page 173.

After you configure and implement this risk policy, the following are possible scenarios:

Scenario	Number of Failed Rules	Risk	Result
An employee access the HR portal during business hours from the corporate network.	Zero	Low	The user can authenticate using `LowRisk` or `MediumRisk` chain.
An employee access the HR portal after business hours from the corporate network.	One (User Time of Login Rule)	Mediu m	The user is required to authenticate using the `MediumRisk` chain.
An employee accesses the HR portal during business hours but from an external network.	One (IP Address Rule)	Mediu m	The user is required to authenticate using the `MediumRisk` chain.
An employee accesses the HR portal after business hours from an external network.	Two (IP Address Rule and User Time of Login Rule)	High	Access is denied.

18 Troubleshooting Risk Service configuration

- Section 18.1, “An error in Syslog when the Risk Service license is not Applied,” on page 275
- Section 18.2, “Cannot read the log file error in risk logs,” on page 275

18.1 An error in Syslog when the Risk Service license is not Applied

Issue: When the license of Risk Service is not applied on the Advanced Authentication server, the health check for Risk Service fails. An error message is displayed in the Syslog stating the context deadline exceeded.

Workaround: Run the following command to stop Risk Service:

```
systemctl stop risk-service
```

You can also ignore the error as it does not affect the actual functionality of the Advanced Authentication server.

Later, if you want to use Risk Service, run the following command to start the service:

```
systemctl start risk-service
```

18.2 Cannot read the log file error in risk logs

Issue: When you modify the CEF Log Forward policy on the Administration portal, an error message is displayed in the **Logs > Risk Service** page. The message indicates that the server cannot read the log file. This issue occurs due to a delay in reloading the Risk Service configuration.

Workaround: Wait for two or three minutes and refresh the **Risk Service** tab to view the latest logs. The old logs are archived.

IV 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

This chapter includes the following examples of third-party integrations. The integration procedures were written based on the latest software available and might require modification with current versions to work as expected. The following instructions assist you in integration. However, there will be changes in the third-party versions and cannot be assured to be completely accurate. We recommend you consult the company that has integrated its product with Advanced Authentication to make these integrations work appropriately:

- 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 Palo Alto GlobalProtect Gateway
- Configuring integration with Salesforce
- Configuring integration with ADFS
- Configuring integration with Google G suite
- Configuring integration with Office 365
- Configuring integration with Sentinel
- Configuring integration with Office 365 without using ADFS
- Configuring integration with Cisco AnyConnect
- Configuring integration with GitLab
- Configuring integration with Filr
- Configuring integration with DUO authentication proxy
- Configuring integration with ArcSight

19 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.

You can find the public key to verify the JWT in the following path:

`https://<AAserver>/osp/a/<tenant_id>/auth/oauth2/.well-known/openid-configuration`. It contains the `jwks_uri`.

You can specify TOP for the `tenant_id` parameter, if the Multitenancy (https://www.netiq.com/documentation/advanced-authentication-63/server-administrator-guide/data/multitncy_opts.html) mode is disabled or you are not in Advanced Authentication as a Service (SaaS).

This section contains the following topics:

- Section 19.1, “Building blocks of OAuth 2.0,” on page 279
- Section 19.2, “Sample OAuth 2.0 Application integrated with Advanced Authentication,” on page 282
- Section 19.3, “OAuth 2.0 attributes,” on page 289
- Section 19.4, “Non standard endpoints,” on page 290

For information on the following see the respective link:

- To create an OAuth 2.0 event, see Creating an OAuth 2.0 / openid connect event.

19.1 Building blocks of OAuth 2.0

The following are the building blocks of OAuth 2.0.

- OAuth 2.0 roles
- OAuth 2.0 grants

19.1.1 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.

19.1.2 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).

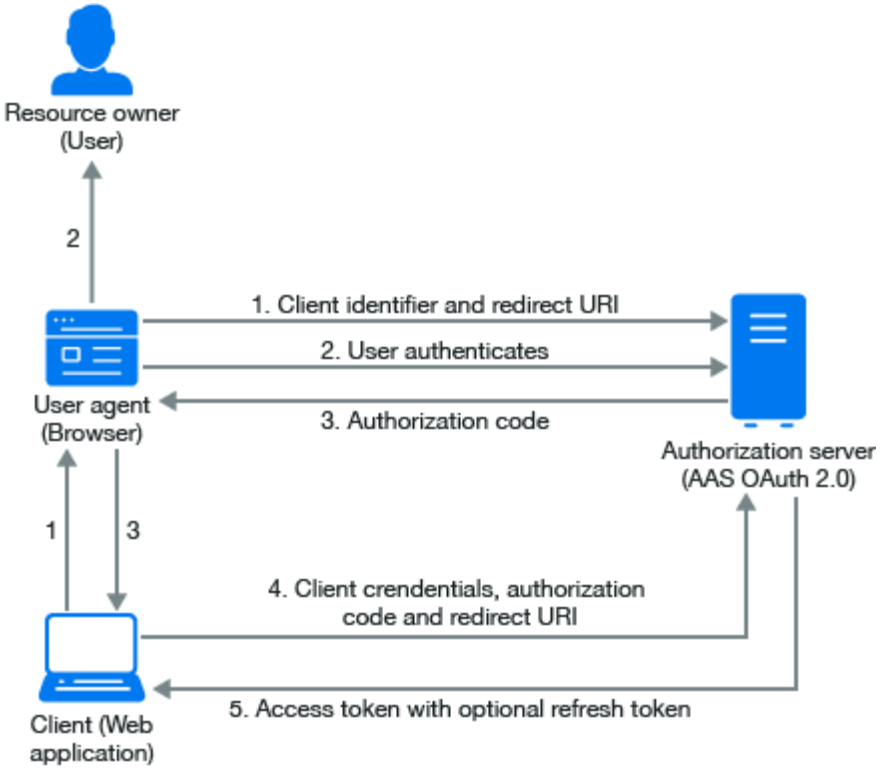
- "Authorization code" on page 280
- "Implicit grant" on page 281

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:

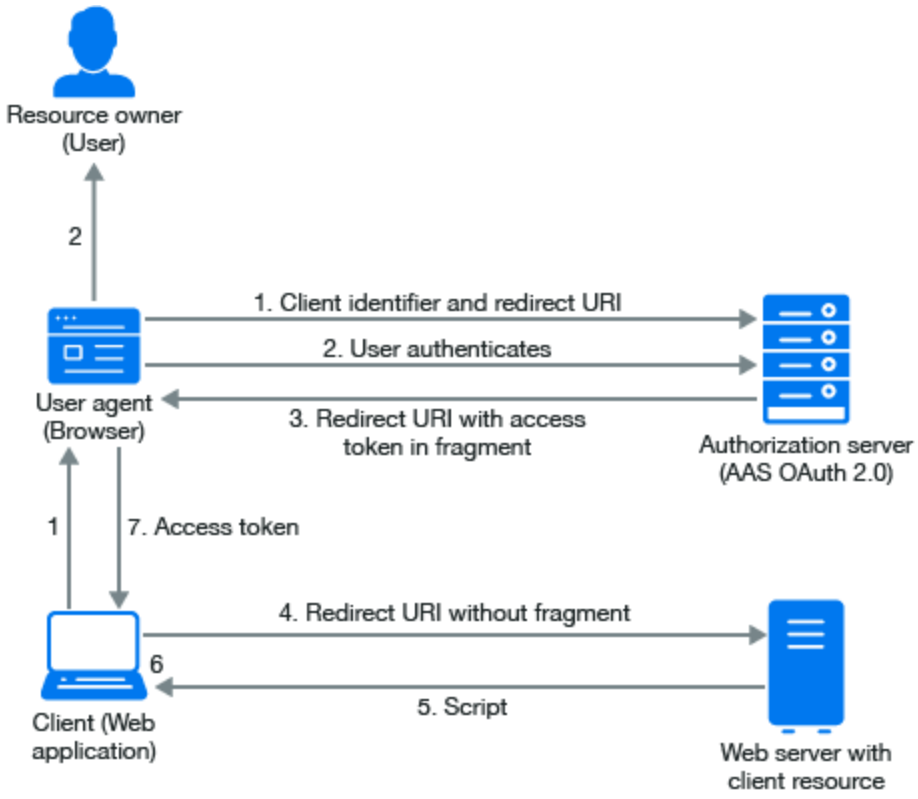
1. The OAuth client initiates the flow when it directs the user agent of the resource owner to the authorization endpoint. The OAuth client includes its client identifier, requested scope, local state, and a redirection URI.
2. The authorization server authenticates the resource owner through the user 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 Impicit 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.

19.2 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).

NOTE: Ensure to install necessary components like package manager and modules according to the Python version in use as a prerequisite.

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: {}\n'.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{}\ndata:\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">
         //<![CDATA[
            function getHashParam(name) {
                var hash = window.location.hash;
                if (hash) {
                    if (name = (new RegExp('[#&]' +
encodeURIComponent(name) + '=([^&]*)')).exec(hash))
                    return decodeURIComponent(name[1]);
                }
            }
            function showResult() {
                if (window.location.hash) {
                  document.getElementById('result').innerHTML = '<table
border="1">'+
                        '<tr><td>access_token</
td><td>'+getHashParam('access_token')+'</td></tr>'+
                        '<tr><td>token_type</
td><td>'+getHashParam('token_type')+'</td></tr>'+
                        '<tr><td>expires_in</
td><td>'+getHashParam('expires_in')+'</td></tr>'+
                        '</table>';
                } else {
                    document.getElementById('result').innerHTML =
'Implicit granted token is not found';
                }
            }
          ]]//>
```

```
        </script>
</head>
<body onload="showResult();">
<div id="result">result</div><br/>
<br/>
Click <a href="/logon">here</a> to obtain an authentication token
through Authorization Code Grant<br/>
Click <a href="/logon-implicit">here</a> to obtain an authentication
token through Implicit Grant (the token will be received in hash part of
THIS page)<br/>
Click <a href="/logon-creds">here</a> to obtain an authentication token
through Resource Owner Password Credentials Grant<br/>
</body>
</html>
```

3. **Token information (token.html)**

```
<!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>
```

19.2.1 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 OpenText 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 OpenText 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**.

     NOTE: The Authorization code grant and Implicit grant fail when the **Use for Owner Password Credentials** is set 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`.

19.3 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

Attribute	Value
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-0TRljvJEe3qKwJiXvy3Ibjvcixfii Y1Q
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-0TRljvJEe3qKwJiXvy3Ibjvcixfii Y1Q

The `client` attribute is a **Client ID** specified in the OAuth 2.0 settings.

19.4 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. For example, Advanced Authentication uses OSP for authentication after Advanced Authentication has obtained 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.

  The following JSON request object code is an example to sign an endpoint.

  { "username" : "< username >" "LoginParameters" : { "internal.osp.oidp.aa.chain-name" : "<chain name>" } }

  where username is name of the user trying to authenticate and chain name is name of the chain configured in the Advanced Authentication server.
- **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.

HTTP status codes

The following table describes the HTTP status codes.

HTTP Status Code	Description
200	The operation was successful.
400	The operation was unsuccessful. Additional error information may be found in the response content.
401	Client authentication missing or invalid.
500	A server error occurred.

The cause of the error can be determined from the additional error information found in the response content.

Response content

The response to a successful request is a serialized JSON object (XML is not currently supported). The **data** field is the signed and encrypted data to be used with another endpoint. The **exp** field is the expiration time of the data as defined by RFC 7519. For more information, see the Data tracker (https://datatracker.ietf.org/doc/html/rfc7519#section-4.1.4).

The following sample code in javascript is an example of the response content.

```
{ "data":"_TXNCmy8ocXUg3Hg7u1TmRRJ3-2JQHcv3XggLbzhX2l6TcM-
11sfYlVatE6KIhP1.e1lJXX3Gj5UlFPoo03ig-4vczT2UtrAzbV4poyN592s~",
"exp":1488210079}
```

NOTE: The web authentication does not query the LDAP directly for users. Web authentication routes the request to the Advanced Authentication server internally. Therefore, if the Advanced Authentication server can match the inbound username with an appropriate attribute in the LDAP server, it would be same as what Advanced Authentication provides.

20 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 the PAP and EAP-TTLS/PAP methods.

- The RADIUS server supports the following authentication methods: **Email OTP**, **Emergency Password**, **LDAP Password**, **OATH OTP**, **Out-of-Band, Password**, **RADIUS Client**, **Security Questions**, **Smartphone**, **SMS OTP**, **Voice OTP**, **Flex OTP** and **Voice** methods. It is possible to use any method using the Out-of-Band method.

  For more information, see Out-of-band.

- By design, Advanced Authentication does not support the single-factor authentication with the **Smartphone**, **Email OTP**, **SMS OTP**, **Security Questions**, **Voice OTP**, or **Voice** method for RADIUS. These methods cannot be the first or single method in a chain. Also, the OATH TOTP and OATH HOTP methods cannot be the first methods in the chain. It is recommended to use these methods as the second-factor in a two-factor chain after the **LDAP Password** method.

To configure pre-defined RADIUS Server event, 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. Specify endpoint name in **Endpoints whitelist**.
6. Set **Bypass user lockout in repository** to **ON**, if you want to allow repository locked-out users to be authenticated on the Advanced Authentication. By default, **Bypass user lockout in repository** is set to **OFF** and users locked on repository is not allowed to authenticate.
7. Set **Return groups on logon** to **ON** if you want to retrieve the group details of users who authenticated to the event in the authentication response.

   With **Return groups on logon** set to **ON**, if Groups is empty, all the groups that the users are associated with are returned in the response. However, to return the required groups, specify the preferred groups in **Groups**.

   The RADIUS protocol according to RFC (https://datatracker.ietf.org/doc/html/rfc2865) has a 4KB limit of response size. The authentication response might exceed the set limit, if a user is a member of several groups. Therefore, it is recommended to use **Groups** to limit the groups' in the response.

   By default, **Return groups on logon** is set to **OFF**, the groups of users authenticated to the event are not returned in the response.
8. Configure Input rule.
9. Configure Chain selection rule.
10. Configure Result specification rule.

You can configure the above RADIUS rules in RADIUS Options policy also. For more information about configuring the RADIUS rules in RADIUS Options Policy, see RADIUS Options.

The rules configured in RADIUS Options policy are called Global level rules and rules configured in RADIUS event are called Event level rules. All the RADIUS rules are executed in the following order.

10a Input rule configured in Global level rules.

10b Event Selection rule configured in Global level rules.

10c Input rule configured in Event level rules.

10d Chain selection rule configured in Event level rules.

10e Chain selection rule configured in Global level rules (if no chain in Event level rules).

10f Authenticate the user.

10g Result specification configured in Global level rules.

10h Result specification configured in Event level rules.

11 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 R**ADIUS 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 and OpenVPN applications do not support this option.

NOTE: If you use the **LDAP Password+Smartphone** chain, you can use an offline authentication by specifying the password in the format `<LDAP Password>&<Smartphone OTP>`. For example, Q1w2e3r4&512385. This option is supported for **LDAP Password+OATH TOTP, Password+Smartphone, Password+OATH TOTP, Password+OATH HOTP.** It is required configure the Input Rule before you use another delimiter or no delimiter.

Before using ampersand or any other special character as a delimiter, you must configure the Input Rule and Chain selection rule in the **RADIUS Options** policy.

NOTE: If the RADIUS log files are overflown of records with the error `Discarding duplicate request from client`, you can increase the timeout on the RADIUS Client. The optimal timeout value needs to be determined by experimenting. It must not exceed 60 seconds.

Customizing prompt messages for RADIUS event

You can customize prompt messages of the authentication methods that are configured for the RADIUS event. The customized prompt messages are displayed when a user initiates authentication to RADIUS event using the configured methods.

For more information about customizing prompt message for RADIUS event, see Customizing prompt messages of the authentication methods for RADIUS event.

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.

Using RADIUS in multitenancy mode

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

21 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).

This chapter contains the following section:

- Section 21.1, “Integrating Advanced Authentication with SAML 2.0,” on page 297

21.1 Integrating Advanced Authentication with SAML 2.0

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

1. Create a **SAML 2.0** event. For more information on how to create a event, see Creating a SAML 2.0 event.
2. Click **Policies > Web Authentication**.
3. (Conditional) Specify the Identity Provider’s URL in **Identity provider URL**. For more information, see Configuring the identity provider.
4. (Conditional) Click **IdP SAML 2.0 Metadata** to download the SAML 2.0 metadata. You can download the SAML 2.0 metadata file only when the Identity Provider's URL is configured. The downloaded SAML 2.0 metadata file is used to configure the Service Provider.
5. Click **Save**.
6. Click **Server Options > Signing Certificate** and save the certificate content in a notepad file for further use.

   NOTE: Use the Identity Provider Signing certificate in your Service Provider.

NOTE: The logout URL must follow the below format:

```
https://<AAServer>/osp/a/TOP/auth/app/logout
```

where TOP is the name of the tenant.

However, it is possible to perform the logout from both Identity Provider and Service Provider using the following URL:

```
https://<AAServer>/osp/a/TOP/auth/app/logout?target=https://<Service Provider>/app/logout
```

For example: `https://<AAServer>/osp/a/TOP/auth/app/logout?target=https://<NAMServer>/nidp/app/logout`

The following are the examples of integration with SAML 2.0.

- Configuring integration with Salesforce
- Configuring integration with ADFS

21.1.1 Requesting Advanced Authentication methods and chains through a SAML authnrequest

SAML 2.0 provides a mechanism to request an **authentication class reference**. For more information, see the SAML 2.0 Core specification (https://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf) in **section 3.3.2.2.1**.

The Service Provider sends the following sample code in the <AuthnRequest>:

```
<samlp:RequestedAuthnContext>
<saml:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorContract</saml:AuthnContextClassRef> </samlp:RequestedAuthnContext>
```

SAML 2.0 defines a bunch of URNs that corresponds to authentication **classes**. For more information, see SAML 2.0 Authentication Context (http://docs.oasis-open.org/security/saml/v2.0/saml-authn-context-2.0-os.pdf).

Some of the authentication class types of Advanced Authentication match the SAML 2.0 references. The Advanced Authentication auth class types are defined in an enum named `AuthClassType`.

In this XML example, the SAML class reference URN maps to the Advanced Authentication's `AuthClassType.MOBILE_ONE_FACTOR_CONTRACT`. The Advanced Authentication value is mapped to `NaafAuthMethod.SMARTPHONE (or NaafAuthMethod.SWISSCOM)`.The code in `NaafEventContractExecutable.filterChains` selects from the available chains any chain that contains one of its methods (in this example) SMARTPHONE or SWISSCOM. (The map from Advanced Authentication methods to OSP auth class type is `NaafContractExecutable.METHOD_TO_TYPE_MAP`.)

In this example, after the user is identified, if there is a chain available with the Smartphone or Swisscom methods, then the authentication proceeds. If not, the authentication fails and Advanced Authentication returns a `no requested authentication context` status to the Service Provider.

An optional `Comparison` attribute can be set on the <RequestedAuthnContext>. This attribute is defined in the SAML 2.0 Core specification (https://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf) in **section 3.3.2.2.1**.

In addition to requesting the Advanced Authentication methods using the SAML 2.0-defined URNs, Advanced Authentication also has a special **contract parameters** class reference URN. The URN is: `urn:uuid:519a6c73-f092-43d3-ab11-8d789ebc2f79`.

The **contract parameters** are added through the URN **q-component**. The URN syntax is defined at RFC 8141 (https://tools.ietf.org/html/rfc8141).

The <NaafEvent> contract executable contains attributes named `allowClientChainSelection` and `allowClientEventSelection`. These attributes allow the authentication chain and the authentication event to be selected through a **contract parameter** from the client, which in this example, is the SAML Service Provider. In the Advanced Authentication `authcfg.xml`, the default value of `allowClientEventSelection` is `false` and `allowClientChainSelection` is `true`.

For example, **ISM** is an event name with the following chains: LDAP+Smartphone, LDAP+SMS_OTP, LDAP+TOTP, LDAP+SecQuest, LDAP+U2F, and LDAP+Voice.

If the `<NaafEvent>` contract executable is configured with the **ISM** event, then the following code will request the LDAP+SMS_OTP chain.

```
<samlp:RequestedAuthnContext>
<saml:AuthnContextClassRef>urn:uuid:519a6c73-f092-43d3-ab11-
8d789ebc2f79?=internal.osp.oidp.aa.chain-name=LDAP%2BSMS_OTP</
saml:AuthnContextClassRef> </samlp:RequestedAuthnContext>
```

The plus sign '`+`' is encoded as '`%2B`'. Advanced Authentication considers that the **q-component,** which starts with `?=`, is in the `x-www-form-urlencoded` format and '+' is a reserved character for this syntax.

The two contract parameters that are defined in the Advanced Authentication class `CFGNaafEvent` are:

- `internal.osp.oidp.aa.chain-name`
- `internal.osp.oidp.aa.event-name`

22 Examples of integrations

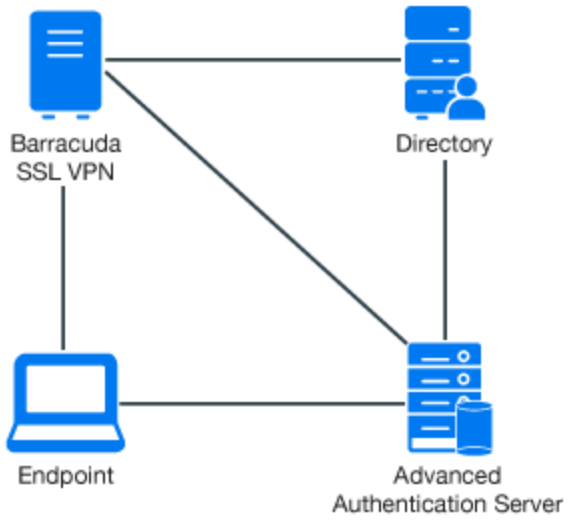
This chapter includes the following examples of third-party integrations. The integration procedures were written based on the latest software available and might require modification with current versions to work as expected. The following instructions assist you in integration. However, there will be changes in the third-party versions and cannot be assured to be completely accurate. We recommend you consult the company that has integrated its product with Advanced Authentication to make these integrations work appropriately:

- Section 22.1, “Configuring integration with Barracuda,” on page 301
- Section 22.2, “Configuring integration with Citrix NetScaler,” on page 303
- Section 22.3, “Configuring integration with dell sonicwall SRA EX-Virtual Appliance,” on page 305
- Section 22.4, “Configuring integration with FortiGate,” on page 307
- Section 22.5, “Configuring integration with OpenVPN,” on page 309
- Section 22.6, “Configuring integration with Palo Alto GlobalProtect Gateway,” on page 311
- Section 22.7, “Configuring integration with Salesforce,” on page 312
- Section 22.8, “Configuring integration with ADFS,” on page 315
- Section 22.9, “Configuring integration with Google G suite,” on page 317
- Section 22.10, “Configuring integration with Office 365,” on page 320
- Section 22.11, “Configuring integration with Sentinel,” on page 322
- Section 22.12, “Configuring integration with Office 365 without using ADFS,” on page 323
- Section 22.13, “Configuring integration with Cisco AnyConnect,” on page 326
- Section 22.14, “Configuring integration with GitLab,” on page 329
- Section 22.15, “Configuring integration with Filr,” on page 334
- Section 22.16, “Configuring integration with DUO authentication proxy,” on page 334
- Section 22.17, “Configuring integration with ArcSight,” on page 335
- Section 22.18, “Configuring integration with Azure,” on page 337
- Section 22.19, “Configuring integration with Amazon Web Services Single Sign-On,” on page 339

22.1 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:

- Section 22.1.1, “Configuring the Advanced Authentication RADIUS server,” on page 302
- Section 22.1.2, “Configuring the Barracuda SSL VPN Appliance,” on page 303
- Section 22.1.3, “Authenticating on Barracuda SSL VPN using Advanced Authentication,” on page 303

22.1.1 Configuring 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 **Save** in **Edit Event**.
6. Click **Policies > Radius Options**.
7. Click **Add** in **Clients**.
8. Specify an **IP address** of the Barracuda SSL VPN appliance.
9. Specify **Name** of the Client.
10. Specify a secret and confirm it.
11. Set **Enabled** to **ON**.
12. Click icon to save the Client details.
13. Click **Save** in **Radius Options**.

22.1.2 Configuring 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.

22.1.3 Authenticating on Barracuda SSL VPN using Advanced Authentication

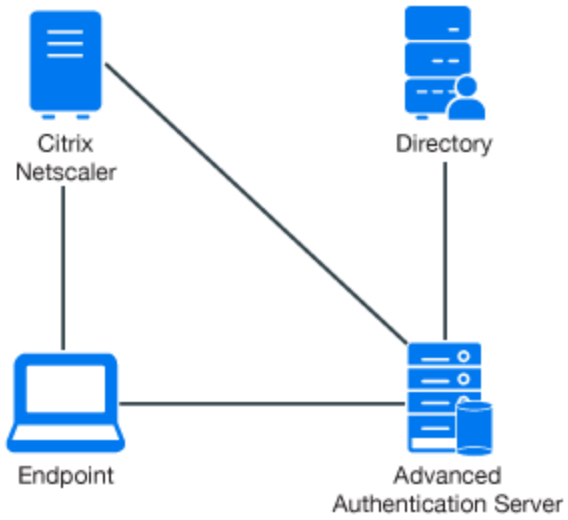
1. Specify the user's credentials.
2. Click **More** and select the configured user database (if the database is not selected by default).
3. Click **Log In** and approve the authentication on the user's smartphone.

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

22.2 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:

- Section 22.2.1, “Configuring the Advanced Authentication RADIUS server,” on page 304
- Section 22.2.2, “Configuring the Citrix NetScaler Appliance,” on page 305
- Section 22.2.3, “Authenticating on the Citrix NetScaler using Advanced Authentication,” on page 305

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.

22.2.1 Configuring 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 **Save** in **Edit Event**.
6. Click **Policies > Radius Options**.
7. Click **Add** in **Clients**.
8. Specify an **IP address** of the Citrix NetScaler appliance.
9. Specify **Name** of the Client.
10. Specify a secret and confirm it.
11. Set **Enabled** to **ON**.
12. Click icon to save the Client details.
13. Click **Save** in **Radius Options**.

22.2.2 Configuring 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.

22.2.3 Authenticating on the Citrix NetScaler using Advanced Authentication

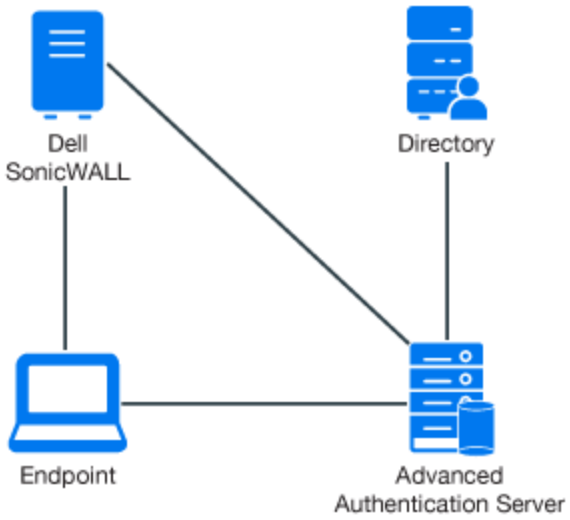
1. Specify the user's credentials then click **Login**.
2. Accept the authentication on your smartphone.

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

22.3 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:

- Section 22.3.1, “Configuring the Advanced Authentication RADIUS server,” on page 306
- Section 22.3.2, “Configuring the dell sonicwall SRA Appliance,” on page 307
- Section 22.3.3, “Authenticating on Dell SonicWall Workspace using Advanced Authentication,” on page 307

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.

22.3.1 Configuring 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 **Save** in **Edit Event**.
6. Click **Policies > Radius Options**.
7. Click **Add** in **Clients**.
8. Specify an **IP address** of the Dell SonicWall appliance.
9. Specify **Name** of the Client.
10. Specify a secret and confirm it.
11. Set **Enabled** to **ON**.
12. Click icon to save the Client details.
13. Click **Save** in **Radius Options**.

22.3.2 Configuring 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.

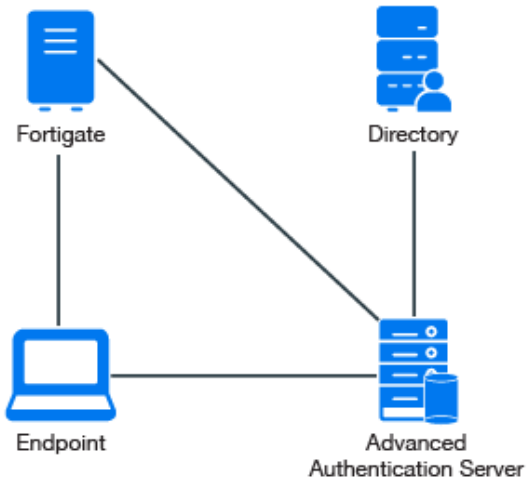
22.3.3 Authenticating on Dell SonicWall Workspace using Advanced Authentication

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

22.4 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:

- Section 22.4.1, “Configuring the Advanced Authentication RADIUS server,” on page 308
- Section 22.4.2, “Configuring the FortiGate Appliance,” on page 308
- Section 22.4.3, “Authenticating on FortiGate using Advanced Authentication,” on page 309

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.

22.4.1 Configuring 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 **Save** in **Edit Event**.
6. Click **Policies > Radius Options**.
7. Click **Add** in **Clients**.
8. Specify an **IP address** of the FortiGate appliance.
9. Specify **Name** of the Client.
10. Specify then RADIUS shared secret and confirm it.
11. Set **Enabled** to **ON**.
12. Click  icon to save the Client details.
13. Click **Save** in **Radius Options**.

22.4.2 Configuring 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.
7. Set the Remote Authentication timeout. Default timeout is 5 seconds. Run the following commands in the command line:

```
config system global
```

```
set remoteauthtimeout 45
```

8. Set the RADIUS client timeout using the following commands:

```
config user radius
edit
set timeout 60
```

22.4.3 Authenticating on FortiGate using Advanced Authentication

1 Specify the user's credentials and click **Login**.

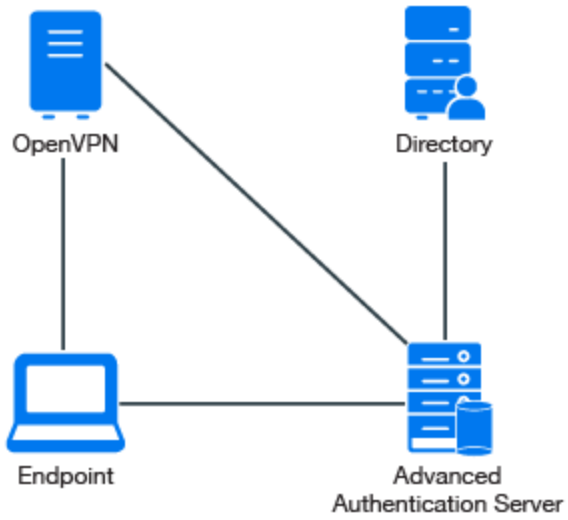
2 Specify the OTP and click **Login**.

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.

22.5 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:

- Section 22.5.1, "Configuring the Advanced Authentication RADIUS server," on page 310
- Section 22.5.2, "Configuring the OpenVPN Appliance," on page 310

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.

You can watch the OpenVPN integration video here:

http://www.youtube.com/watch?v=K-h28nU2vs0

22.5.1 Configuring 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 **Save** in **Edit Event**.
6. Click **Policies > Radius Options**.
7. Click **Add** in **Clients**.
8. Specify an **IP address** of the OpenVPN appliance.
9. Specify **Name** of the Client.
10. Specify a secret and confirm it.
11. Set **Enabled** to **ON**.
12. Click  icon to save the Client details.
13. Click **Save** in **Radius Options**.

22.5.2 Configuring 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.

If you have assigned multiple chains in the **Used** section of the RADIUS event for connecting to OpenVPN, then you must specify *<username>&<chain shortname>* in the **username**.

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 (https://docs.openvpn.net/docs/access-server/openvpn-access-server-command-line-tools.html#authentication-failure-lockout-policy).

22.6 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:

- Section 22.6.1, “Adding the RADIUS server,” on page 311
- Section 22.6.2, “Adding an authentication profile,” on page 312
- Section 22.6.3, “Configuring globalprotect gateway,” on page 312

22.6.1 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 **OpenText 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**.

22.6.2 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.

22.6.3 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 **OpenText authentication profile** created in Add an Authentication Profile from the list.
4. Click **OK** to save the GlobalProtect Gateway settings.

22.7 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:

- Section 22.7.1, “Configuring the Advanced Authentication SAML 2.0 event,” on page 312
- Section 22.7.2, “Configuring to authenticate on Salesforce with SAML 2.0,” on page 313
- Section 22.7.3, “Obtaining the signing certificate of Advanced Authentication,” on page 313
- Section 22.7.4, “Configuring the Salesforce domain name,” on page 313
- Section 22.7.5, “Configuring the SAML provider,” on page 314
- Section 22.7.6, “Verifying Single Sign-On to Salesforce,” on page 315

22.7.1 Configuring 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**.

22.7.2 Configuring to authenticate on Salesforce with SAML 2.0

1 Click **Policies > Web Authentication.**

2 Set **Identity provider 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. Configure an external .
2. Specify the address with port number in **Identity provider 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 Click **Save**.

22.7.3 Obtaining the signing certificate of Advanced Authentication

1 Click **Server Options** in the Advanced Authentication Administration portal.

2 Verify whether the Signing Certificate is available and use the certificate.

3 Navigate to **Policies > Web Authentication** and click **Download IdP SAML 2.0 Metadata**.

A new tab launches with the SAML 2.0 metadata that includes the certificate in `x.509` format.

4 Find the tag `<ds:X509Certificate>` and copy the certificate that follows to a notepad file.

5 Add the `---BEGIN CERTIFICATE ----------` at the beginning and `---END CERTIFICATE----------` at end of the certificate in the notepad file.

6 Save the notepad file for further use.

22.7.4 Configuring the 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.

22.7.5 Configuring the SAML provider

1 Click **Settings > Identity > Single Sign-On Settings**.

2 Upload the Identity Provider Signing Certificate that you obtained in Step 6 of section 27.7.3.

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 **Browse** 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 appears, 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**.

22.7.6 Verifying Single Sign-On to Salesforce

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

NOTE: While logging in to Salesforce if an error message `Single Sign-on error` is displayed after succeeding all methods in the chain, you must change the **SAML Identity Type** in the Salesforce console.

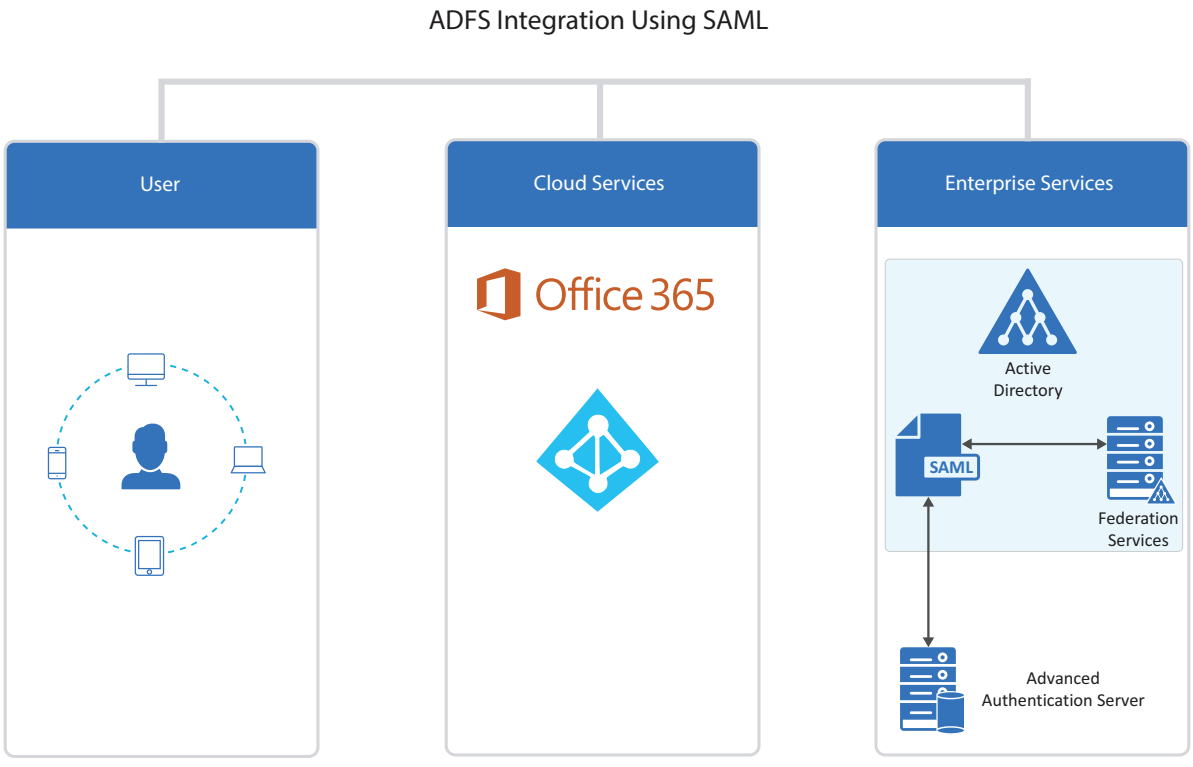
For more information, see Error while logging in to Salesforce.

22.8 Configuring integration with ADFS

This section provides the configuration information on integrating Advanced Authentication with ADFS (Active Directory Federation Services). This integration secures the ADFS connection.

The following diagram represents Advanced Authentication and ADFS integration using SAML.

Figure 22-1



To configure the Advanced Authentication integration with ADFS using SAML 2.0 perform the following configuration tasks:

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

- Section 22.8.1, “Configuring the Advanced Authentication SAML 2.0 event,” on page 316
- Section 22.8.2, “Making the corresponding changes in ADFS,” on page 317

22.8.1 Configuring the Advanced Authentication SAML 2.0 event

1. Open the Advanced Authentication Administration portal.
2. Click **Events > Add** to add a new event.
3. 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**.

     You can perform one of the following, instead of pasting the metadata:
     - Click **Browse** and upload the saved XML file.
     - Get the endpoints including full URLs through powershell using the following command:

       ```
       get-adfsendpoint
       ```

       For more information, see Get ADFS Endpoint (https://docs.microsoft.com/en-us/powershell/module/adfs/get-adfsendpoint?view=win10-ps)
   - 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.
4. Click **Policies > Web Authentication**.
5. Set **Identity provider 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. Configure an external .
   2. Specify the address in **Identity provider URL** instead of specifying an address of a single Advanced Authentication server.
6. 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.

22.8.2 Making 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);
```

22.9 Configuring integration with Google G suite

This section provides the configuration information on integrating Advanced Authentication with Google G Suite. This integration secures the connection.

The following diagram represents Advanced Authentication in Google G Suite.



To configure the Advanced Authentication integration with Google G Suite using SAML 2.0, perform the following configuration tasks:

- Section 22.9.1, “Obtaining the signing certificate of Advanced Authentication,” on page 318
- Section 22.9.2, “Configuring Google G suite,” on page 318
- Section 22.9.3, “Configuring the Advanced Authentication event,” on page 319
- Section 22.9.4, “Configuring to authenticate on Google G-Suite with SAML 2.0,” on page 319
- Section 22.9.5, “Verifying single sign-on to Google suite,” on page 319

NOTE: As a prerequisite, ensure that you finalize the setup of G Suite by accepting the agreement and clicking **Finalize setup**.

22.9.1 Obtaining the signing certificate of Advanced Authentication

1. Click **Server Options** in the Advanced Authentication Administration portal.
2. Verify whether the Signing Certificate is available. Use the certificate.
3. Navigate to **Policies > Web Authentication** and click **Download IdP SAML 2.0 Metadata**.

   A new tab launches with the SAML 2.0 metadata that includes the certificate in `x.509` format.
4. Find the tag `<ds:X509Certificate>` and copy the certificate that follows to a notepad file.
5. Add the `---BEGIN CERTIFICATE ----------` at the beginning and `---END CERTIFICATE----------` at end of the certificate in the notepad file.
6. Save the notepad file for further use.

22.9.2 Configuring Google G suite

1. Login to the Google's Administration console.
2. Open the **Security** section.
3. Expand **Set up single sign-on (SSO)**.
4. Enable **Setup SSO with third party identity provider**.
5. Specify the following parameters:

   5a **Sign-in page URL**: `https://<AdvancedAuthenticationServerAddress>/osp/a/TOP/auth/saml2/sso`. Replace `AdvancedAuthenticationServerAddress` with the domain name or IP address of your Advanced Authentication server.

   5b **Sign-out page URL**: `https://<AdvancedAuthenticationServerAddress>/osp/a/TOP/auth/app/logout`.

   5c **Change password URL**: `https://<AdvancedAuthenticationServerAddress>` or Self-Service Password Reset URL.

   5d Upload the Identity Provider Signing Certificate that you downloaded in Step 5 on page 318.
6. Clear **Use a domain specific issuer** if you have one domain in G Suite or select the option if you have more than one domain in G Suite.

   Ensure that you have a user account in a repository that corresponds to a user account in Google. An email address specified in the **Contact information** for the Google account must be the same as an address from email attribute for the corresponding account of your repository.

   NOTE: You cannot use the Google administrator account with SAML.

22.9.3 Configuring the Advanced Authentication event

1 Open the Advanced Authentication Administration portal.

2 Click **Events > New Event** to add a new event with the following options:

2a **Name**: Google

2b **Event Type**: SAML2

2c **Chains**: Select the required chains.

2d **SP SAML 2.0 metadata**: Paste the Service Provider metadata.

Sample metadata is as follows:

```
<EntityDescriptor entityID="google.com"
xmlns="urn:oasis:names:tc:SAML:2.0:metadata"> <SPSSODescriptor
protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
<NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-
format:emailAddress</NameIDFormat> <AssertionConsumerService
index="1" Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://www.google.com/a/mycompany.com" /> </
SPSSODescriptor> </EntityDescriptor>
```

Replace `mycompany.com` in the Location URL to your primary domain from the **Domains** settings in Google.

NOTE: It is not recommended to use the sample metadata in the production environment.

NOTE: You must use the Service Provider metadata when one domain exists in the G Suite. If you have more than one domain in G Suite, then every Service Provider metadata for each domain must have `google.com` as an entityID replaced with `google.com/mycompany.com`, where `mycompany.com` is your domain name.

2e Select **Send E-Mail as NameID (suitable for G-Suite)** from the **NameID formatting options**. This is applicable for the G-Suite.

2f Click **Save**.

22.9.4 Configuring to authenticate on Google G-Suite with SAML 2.0

In **Policies > Web Authentication**, set **Identity provider URL** to `https://AdvancedAuthenticationServerAddress/` and replace `AdvancedAuthenticationServerAddress` with domain name or IP address of your Advanced Authentication server.

22.9.5 Verifying single sign-on to Google suite

Open the Google Sign in page and specify an email address of the user from **Basic information** of the Google account (email address of Google account). Google redirects to the Advanced Authentication server, where the user must authenticate. After successful authentication, the Advanced Authentication server redirects the user back to Google.

22.10 Configuring integration with Office 365

This section provides the configuration information on integrating Advanced Authentication with Office 365. This integration secures the connection.

The following diagram represents integration of Advanced Authentication with Office 365.



To configure the integration of Advanced Authentication with Office 365, perform the following tasks:

- Section 22.10.1, “Configuring Advanced Authentication SAML 2.0 event,” on page 320
- Section 22.10.2, “Making the corresponding changes in ADFS,” on page 321
- Section 22.10.3, “Authenticating on Office 365,” on page 322

Ensure that the following requirements are met:

- ADFS v4.0, Domain Controller, and other components must be configured to work with Microsoft Office 365.

22.10.1 Configuring Advanced Authentication SAML 2.0 event

1. Open the Advanced Authentication Administration portal.
2. Click **Events > Add** to add a new event.
3. Create an event with the following parameters:
   - Name: **Office 365**
   - 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.
   - Select **Send ImmutableId (User objectId) as NameID (required for Microsoft Office 365)** from the **NameID formatting options**. This is required for integration with Microsoft Office 365.

   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.
4. Click **Save**.
5. Click **Policies > Web Authentication**.

6 Set the **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. Configure an external .
2. Specify the address in **External URL** instead of specifying an address of a single Advanced Authentication server.

---

7 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.

---

8 Click **Save**.

22.10.2 Making the corresponding changes in ADFS

1 Open the ADFS management console.

2 Click **Claims Provider Trusts > Add Claims Provider trust**.

3 Click **Start** in the **Add Claims Provider Trust Wizard**.

4 Click **Import data about the claims provider from a file** in the **Select Data Source** tab.

5 Browse the **Federation metadata file**.

You can download the Federation metadata from the Advanced Authentication metadata URL: `https://<aaf-server>/osp/a/TOP/auth/saml2/metadata`.

6 Click **Next**.

7 Specify the **Display name**.

8 Click **Next**.

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

10 Click **Close**.

11 Right-click the **Display name** and click **Edit Claim Rules**.

12 Click **Add Rule**.

13 Select **Send Claims Using a Custom Rule from Claim rule template** in the **Add Transform Claim Rule Wizard**.

14 Click **Next**.

15 Specify the **Claim rule name**.

16 Paste the following in **Custom rule**:

```
c:[Type == "netbiosName"] => issue(Type = "http://
schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname",
Issuer = c.Issuer, OriginalIssuer = c.OriginalIssuer, Value = c.Value,
ValueType = c.ValueType);
```

17 Click **OK**.

18 Launch Windows Powershell and run the following command to connect to your Office 365 tenant:

```
Connect-MsolService
```

19 Run the following command to disable the `PromptLoginBehavior` parameter and to send `wfresh=0` to AD FS for fresh authentication of federated users.

```
Set-MsolDomainFederationSettings -DomainName <domain_name> -
PromptLoginBehavior Disabled
```

22.10.3 Authenticating on Office 365

1 Launch `http://office.com/`.

2 Login with your credentials.

3 Select **Advanced Authentication** to go through the multi-factor authentication.

4 You will be redirected to the OAuth or SAML Login page.

5 You must go through the specified chains for authentication.

You might face an issue when authenticating to Microsoft teams and Outlook apps on a smartphone. For the workaround, see “Issue with authenticating on Office 365”.

22.11 Configuring integration with Sentinel

This section provides the configuration information about integrating Advanced Authentication with Sentinel for managing logs. With this integration the syslog files are gathered and transmitted from Advanced Authentication to Sentinel sever, where an administrator can search the events to analyze, monitor, and generate a report.

To configure the integration of Advanced Authentication with Sentinel, perform the following tasks:

- Section 22.11.1, “Configuring the CEF log forward policy on Advanced Authentication,” on page 322
- Section 22.11.2, “Searching the events on Sentinel,” on page 323

22.11.1 Configuring the CEF log forward policy on Advanced Authentication

To forward the syslog details to Sentinel, you must configure the **CEF log Forward** policy by performing the following steps:

1 Open the Advanced Authentication Administration portal.

2 Click **Policy > CEF Log Forward**.

3 Specify the Sentinel server IP address in **Syslog server**.

4 Specify the port number in **Port**.

For example, you can specify 1443.

5 Select the transport layer details in **Transport**.

For example, you can select **TCP with TLS.**

6 Click **Save**.

7 Restart the Advanced Authentication server to apply the changes.

22.11.2 Searching the events on Sentinel

1 Open the Sentinel console.

2 Specify the query `((sev:[0 TO 5])) AND (sp:"CEF")` in the Search bar, then click **Search**.

The events with severity 0 to 5 are displayed. You can download the events in the `csv` format.

22.12 Configuring integration with Office 365 without using ADFS

This section provides the configuration information about integrating Advanced Authentication with Microsoft Office 365. This integration allows users to log in to Office 365 by using their corporate password. During authentication, the specified password is validated by using the federated on-premises Active Directory.



NOTE: The SAML 2.0 supports web-based clients and email-rich clients. With this integration, only limited clients are available for single sign-on.

For example, the Microsoft Teams desktop client does not support SAML; therefore, the client cannot automatically sign in after this integration.

To configure the Advanced Authentication integration with Office 365 using SAML 2.0 perform the following tasks:

- Section 22.12.1, “Configuring the Advanced Authentication SAML 2.0 event,” on page 324
- Section 22.12.2, “Configuring the identity provider URL,” on page 324
- Section 22.12.3, “Obtaining the signing certificate of Advanced Authentication,” on page 324
- Section 22.12.4, “Enabling Single Sign-On to Office 365,” on page 324
- Section 22.12.5, “Verifying Single Sign-On to Office 365,” on page 326

Before integration ensure to download the Office 365 SAML Metadata from Microsoft Online Service (https://nexus.microsoftonline-p.com/federationmetadata/saml20/federationmetadata.xml).

You can watch the Office 365 integration video here:

http://www.youtube.com/watch?v=wJChEJrbnHk

22.12.1 Configuring the Advanced Authentication SAML 2.0 event

1. Log in to the Advanced Authentication Administration portal.
2. Click **Events > Add**.
3. Create an event with the following parameters:
   - **Name**: Office365
   - **Event Type**: SAML 2
   - **Chains**: Select the preferred chains
   - Perform one of the following to import the metadata:
     - Paste the content of the file `https://nexus.microsoftonline-p.com/federationmetadata/saml20/federationmetadata.xml` to **SP SAML 2.0 meta data**.

       Or
     - Click **Browse** and upload the saved XML file.
   - Select **Send ImmutableId (User objectId) as NameID (required for Microsoft Office 365)** from the **NameID formatting options**. This is required for integration with Microsoft Office 365 without ADFS.
4. Click **Save**.

22.12.2 Configuring the identity provider URL

1. Click **Policies > Web Authentication** in the Advanced Authentication Administration portal.
2. Set the **Identity Provider URL** to `https://AdvancedAuthenticationServerAddress/` and replace `AdvancedAuthenticationServerAddress` with domain name or IP address of your Advanced Authentication server.
3. Click **Save**.

22.12.3 Obtaining the signing certificate of Advanced Authentication

1. Click **Server Options** in the Advanced Authentication Administration portal.
2. Click **Signing Certificate** and save the certificate content in a notepad file for further use.

22.12.4 Enabling Single Sign-On to Office 365

It is required to add a custom domain to Office 365 to federate your Office 365 tenant with Advanced Authentication as the external identity provider. You cannot federate your `onmicrosoft.com` domain. It is not recommended to set the custom domain that you have added to Office 365 as the default domain. However, if you set the custom domain as default then you cannot federate it.

To enable single sign-on to Office 365 perform the following tasks:

- “Enabling directory synchronization in Office 365” on page 325
- “Federating the custom domain using Advanced Authentication” on page 325

Enabling directory synchronization in Office 365

1. Log in to the Office 365 Identity Federation Setup page (https://portal.office.com/IdentityFederation/IdentityFederation.aspx) as the tenant administrator. We recommend you to follow and complete the described ten steps to achieve SSO.
2. Review and prepare for SSO as described in the step 1 (https://docs.microsoft.com/en-us/previous-versions/azure/azure-services/jj151786(v=azure.100)) of Identity Federation Setup page.
3. Skip step 2 to integrate without AD FS.

   NOTE: In this integration, it is not required to deploy AD FS. Here, Advanced Authentication replaces AD FS and acts as Security Token Service (STS) for SSO. Ensure to make note of the UPN requirements for SSO.

4. Do not install the Windows Azure Active Directory Federation Services 2.0 as described in step 3. Instead, install the Microsoft Online Services Sign-in Assistant on a computer joined to your AD domain then open PowerShell and run the following command to install the Microsoft Azure Active Directory Module for Windows PowerShell:

   ```
   Install-Module MSOnline
   ```

   For more information about Office 365 PowerShell, see Connect to Office 365 PowerShell (https://docs.microsoft.com/en-us/office365/enterprise/powershell/connect-to-office-365-powershell).
5. Review the prerequisites for Active Directory synchronization and activate the Active Directory synchronization for your domain as described in step 5 and 6.
6. Install and configure the Directory Sync tool on the same server where you have installed the Microsoft Azure Active Directory Module for Windows PowerShell.
7. Launch Azure Active Directory Connect.
8. In the **Express settings** page, click **Custom Settings**.
9. In the **User sign-in** page, select **Do not configure** as **Sign On method**.
10. In the **Identifying Users** page, select **objectGUID** from **Source Anchor**.
11. Verify the Active Directory Synchronization and activate the Office 365 licensing for unlicensed but synchronized users.

Federating the custom domain using Advanced Authentication

1. Log in to the domain-joined computer where you have installed the following components:
   - Microsoft Online Services Sign-in Assistant
   - Microsoft Azure Active Directory Module for Windows PowerShell
   - Azure AD Connect tool
2. Launch Windows Powershell and then run the following command to connect to your Office 365 tenant:

   ```
   Connect-MsolService
   ```
3. Run the following command to verify whether your Office 365 domain is federated:

   ```
   get-msoldomain -domain samplecompany.com
   ```

In case the authentication type of your Office 365 domain is set to Federated, you must convert the authentication type to Managed using the following command:

```
Set-MsolDomainAuthentication –DomainName samplecompany.com -
Authentication Managed
```

4 Set the identity provider details in the PowerShell variables as follows:

- `$dom="fully_qualified_domain_name"`

  For example, `$dom="samplecompany.com"`
- `$uri="https://AdvancedAuthenticationServerAddress/osp/a/TOP/auth/saml2/metadata"`
- `$url="https://AdvancedAuthenticationServerAddress/osp/a/TOP/auth/saml2/sso"`
- `$logoutUrl="https://AdvancedAuthenticationServerAddress/osp/a/TOP/auth/saml2/slo"`
- `$protocol="SAMLP"`
- `$cert="paste the signing certificate that you have saved in a notepad file"`

5 Run the following command to convert your Office 365 domain to Federated authentication:

```
Set-MsolDomainAuthentication –DomainName $dom -Authentication Federated
-PassiveLogOnUri $url -IssuerUri $uri -LogOffUri $logoutUrl -
PreferredAuthenticationProtocol SAMLP -SigningCertificate $cert
```

6 Run the following command to verify the federation settings of your Office 365 domain:

```
Get-MsolDomainFederationSettings -domain samplecompany.com
```

22.12.5 Verifying Single Sign-On to Office 365

1 On the Microsoft Office page (http://office.com/), log in with your credentials.

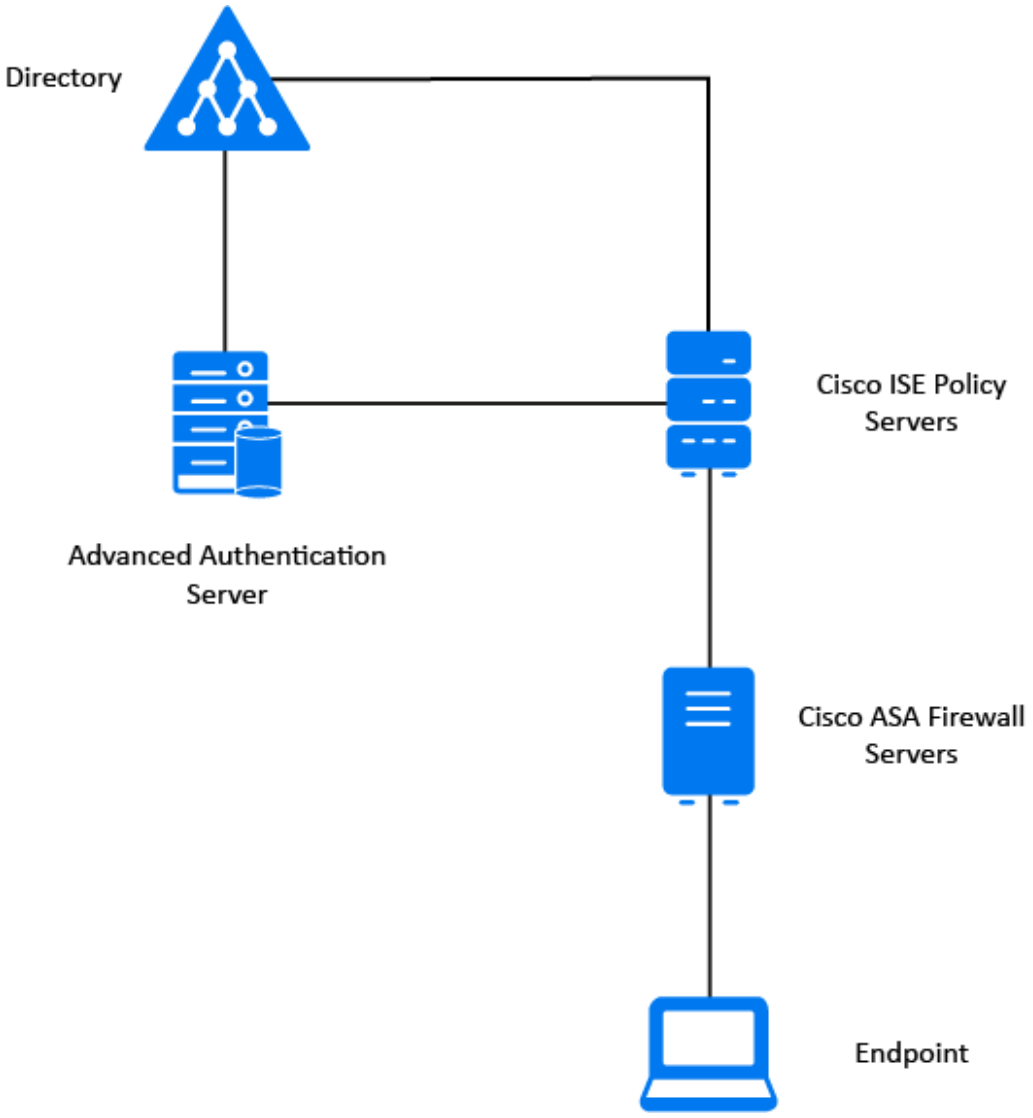
The page redirects to the Advanced Authentication SAML Login page.

2 Select the preferred chain for authentication.

You must pass all methods in the chain to authenticate successfully.

22.13 Configuring integration with Cisco AnyConnect

This section provides the configuration information on integrating Advanced Authentication with Cisco AnyConnect. This integration secures the Cisco AnyConnect VPN connection.



To configure the Advanced Authentication integration with Cisco AnyConnect perform the following tasks:

- Configuring the Advanced Authentication RADIUS server
- Enabling the Connection profile in cisco ASA
- Creating a group policy in cisco ASA
- Adding a RADIUS token server in cisco ISE
- Configuring policy sets in cisco ISE

Ensure that you meet the following requirements:

- Install and configure Cisco ASA 5555-X with Firepower
- Install Cisco ISE
- Install Advanced Authentication appliance
- Configure a repository with the user data in the Advanced Authentication server

22.13.1 Configuring 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 **Save** in **Edit Event**.
6. Click **Policies > Radius Options**.
7. Click **Add** in **Clients**.
8. Specify an **IP address** of the Cisco ISE server.
9. Specify **Name** of the Client.
10. Specify the RADIUS shared secret and confirm it.
11. Set **Enabled** to **ON**.
12. Click  icon to save the Client details.
13. Click **Save** in **Radius Options**.

22.13.2 Enabling the Connection profile in cisco ASA

1. Navigate to **Configuration > Remote Access VPN > Network (Client) Access > AnyConnect Connection Profiles**.
2. Select the **AnyConnect VPN profile** in **Connection Profiles** and click **Edit**.

   The Edit AnyConnect Connection Profile window is displayed.
3. Set the **Method** as **AAA** in the **Authentication**.
4. Select the group created for Advanced Authentication server from **AAA Server Group**.
5. Click **OK**.
6. Click **Apply**.

22.13.3 Creating a group policy in cisco ASA

1. Navigate to **Configuration > Remote Access VPN > Network (Client) Access > Group Policies > Add > Servers**.
2. Specify the name of policy in **Name**.
3. Specify the text to display as message in **Banner**.
4. Click **More Options** then select **Clientless SSL VPN** and **SSL VPN client** as the **Tunneling Protocols**.
5. Click **OK** and **Apply**.

22.13.4 Adding a RADIUS token server in cisco ISE

1. Navigate to **Administration > Identity Management > External Identity Sources** in Cisco ISE.
2. Click **RADIUS Token** from the External Identity Sources navigation pane on the left.

3 Click **Add**.

4 Specify the following details in the **Connection** tab:

- **Host IP**: IP address or host name of the Advanced Authentication server.
- **Shared Secret**: Secret set in the RADIUS server to establish a connection.
- **Authentication Port**: Port to communicate with the RADIUS server. The default port is 1812.
- **Server Timeout**: Time in seconds that Cisco ISE should wait for a response from the RADIUS token server before it determines that the primary server is down. The default timeout value is 5 secs.
- **Connection Attempts**: The number of times that Cisco ISE should reconnect to the primary server before moving on to the secondary server (if configured) or dropping the request if there is no secondary server. The default is 3.

5 Click **Save** and **Submit**.

22.13.5 Configuring policy sets in cisco ISE

1 Navigate to **Work Centers > Network Access > Policy Sets**.

2 From the Status column, click the current **Status** icon and from the dropdown list update the status for the policy set as necessary.

3 Specify Policy Set Name and Description.

4 Select the **Network Access: Device IP Address** attribute and **Equals** operator.

5 Click **Save**.

After you complete all the above tasks, configure an authorization policy for the preferred VPN profile and user group in the repository.

22.13.6 Authenticating to Cisco AnyConnect using Advanced Authentication

1 Launch Cisco AnyConnect Client.

2 Specify the credentials and click **Login**.

3 Specify the input for second-factor authenticator as the administrator has configured.

4 Click **Login**.

22.14 Configuring integration with GitLab

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

To configure the integration of Advanced Authentication appliance with GitLab using SAML 2.0 perform following tasks:

- Section 22.14.1, “Configuring GitLab for Advanced Authentication,” on page 330
- Section 22.14.2, “Creating the relying party trust on ADFS,” on page 331

- Section 22.14.3, “Creating the claims party trust on ADFS,” on page 332
- Section 22.14.4, “Configuring the SAML 2.0 event on Advanced Authentication,” on page 333

Ensure that the following requirements are met:

- Advanced Authentication is configured with a repository (Active Directory).
- A user account has been created in a repository that corresponds to a user account in GitLab. The email address used for logging in to the GitLab account must be the same as an address from email attribute for the corresponding account of your repository.

22.14.1 Configuring GitLab for Advanced Authentication

GitLab can be configured to act as a SAML 2.0 Service Provider (SP). This allows GitLab to consume assertions from a SAML 2.0 Identity Provider (which is Advanced Authentication here).

First configure SAML 2.0 support in GitLab, then register the GitLab application in the Identity Provider (IdP).

On your GitLab server, perform the following steps:

1. In the `vi /etc/gitlab/gitlab.rb` file, perform the following steps:
2. To allow users to use SAML to sign up without having to manually create an account first, add the following values to your configuration for omnibus package:

```
gitlab_rails['omniauth_enabled'] = true
gitlab_rails['omniauth_allow_single_sign_on'] = ['saml']
gitlab_rails['omniauth_block_auto_created_users'] = false
```

3. You can automatically link SAML users with existing GitLab users if their email addresses match by adding the following setting:

```
gitlab_rails['omniauth_auto_link_saml_user'] = true
```

4. Add the provider configuration:

```
 gitlab_rails['omniauth_providers'] = [
 {
    name: 'saml',
args:{
 assertion_consumer_service_url: 'https://<gitlabserver address>/users/
auth/saml/callback',
             idp_cert_fingerprint:
'A3:8D:36:9E:9C:B7:31:0E:14:26:A5:10:68:73:07:A7:CA:7C:9E:BB',
             idp_sso_target_url: 'https://<adfs-serveraddress>/adfs/ls/
',
             idp_slo_target_url: 'https://<adfs-serveraddress>/adfs/ls/
',
             issuer: 'https://<gitlab_serveraddress>',
             name_identifier_format:
```

```
'urn:oasis:names:tc:SAML:2.0:nameid-format:transient',
            attribute_statements: {
               username: ['http://schemas.xmlsoap.org/ws/2005/05/
identity/claims/upn'],
              email: ['http://schemas.xmlsoap.org/ws/2005/05/identity/
claims/emailaddress'],
               name: ['http://schemas.xmlsoap.org/ws/2005/05/identity/
claims/name'],
               first_name: ['http://schemas.xmlsoap.org/ws/2005/05/
identity/claims/givenname'],
               last_name: ['http://schemas.xmlsoap.org/ws/2005/05/
identity/claims/surname'],
            }
         }
```

5 Change the value for `assertion_consumer_service_url` to match the HTTPS endpoint of GitLab (append `users/auth/saml/callback` to the HTTPS URL of your GitLab installation to generate the correct value).

6 Change the values of `idp_cert_fingerprint`, `idp_sso_target_url`, `name_identifier_format` to match your IdP. If a fingerprint is used, it must be a SHA1 fingerprint. For more information, see the omniauth-saml documentation (https://github.com/omniauth/omniauth-saml).

7 Change the value of `issuer` to a unique name, which will identify the application to the IdP. Ensure to configure the `issuer` with the GitLab server address.

8 For the changes to take effect, you must reconfigure GitLab if you installed through Omnibus.

9 Register the GitLab SP in the IdP(Advanced Authentication). For more information, see Configuring the SAML 2.0 event on Advanced Authentication.

22.14.2 Creating the relying party trust on ADFS

1 On the ADFS Management console, click **Relying Party Trusts > Add Relying Party Trust**.

2 Click **Start**.

3 To import GitLab metadata, perform the following:

3a Select **Import data about the relying party from a file**.

3b Specify the **GitLab URL** in `https://<gitlab_serveraddress>/users/auth/saml/metadata` format.

3c Click **Next**.

4 Specify **Display Name** and **Notes** for GitLab and click **Next**.

5 Select **Permit everyone** from **Choose an access control policy list** to configure access control policy for ADFS and click **Next**.

6 Verify the values imported from the GitLab metadata and click **Next**.

7 Select **Configure claims issuance policy for this application** and click **Close**.

8 Select the trust created for GitLab on the Relying Party Trusts and click **Edit Claim Rules**.

9 In the **Issuance Transform Rule** tab, add two rules:

- To add the first rule, perform the following steps:
  1. Click **Add Rule**.

2. Select **Transform an incoming Claim** from **Claim Rule Template**.
3. Specify the **Claim rule name**.
4. Select **Name ID** from **Incoming claim type**.
5. Select **Unspecified** from **Incoming name ID format**.
6. Select **Name ID** from **Outgoing claim type**.
7. Select **Transient Identifier** from **Outgoing name ID format**.
8. Select **Pass through all claim values**.
9. Click **Finish**.

- To add the second rule, perform the following steps:
  1. Click **Add Rule**.
  2. Select **Pass Through or Filter an Incoming Claim** from **Claim Rule Template** and click **Next**.
  3. Specify the **Claim rule name**.
  4. Select **E-mail Address** from **Incoming claim type**.
  5. Select **Pass through all claim values**.
  6. Click **Finish**.

22.14.3 Creating the claims party trust on ADFS

1. Open the ADFS management console.
2. Expand the **Trust Relationships** menu.
3. Click **Add Claims Provider trust**.
4. Select **Import data about the claims provider**.
5. Paste **OSP metadata URL** in `https://<AAF_server_hostname>/osp/a/TOP/auth/saml2/metadata` format or import the file manually.

   It may not work for the self-signed certificate. You can copy metadata from OSP URL to an XML file and provide the file name.
6. Specify the **Display name**.
7. **Edit Claim Rules** for the created claims provider trust.
8. In the **Acceptance Transform Rules** tab, add two rules:
   - To add the first rule, perform the following steps:
     1. Click **Add Rule**.
     2. Select **Transform an incoming Claim** from **Claim Rule Template** and click **Next**.
     3. Specify **Claim rule name**.
     4. Select **Name ID** from **Incoming claim type**.
     5. Select **Transient Identifier** from **Incoming name ID format**.
     6. Select **Name ID** from **Outgoing claim type**.
     7. Select **Unspecified** from **Outgoing name ID format**.

8. Select **Pass through all claim values**.
9. Click **Finish**.

- To add the second rule, perform the following steps:
  1. Click **Add Rule**.
  2. Select **Transform an incoming Claim** from **Claim Rule Template** and click **Next**.
  3. Specify the **Claim rule name**.
  4. Select **mail** from **Incoming Claim Type**.
  5. Select **E-mail Address** from **Outgoing claim type**.
  6. Select **Pass through all claim values**
  7. click **Finish**.

9 Open **Properties** for the created claims provider trust and navigate to the **Endpoints** tab.

10 Ensure that the **Binding** of all endpoints is set to `POST`.

WARNING: While removing the existing endpoints from the **Endpoints** tab, make a note of configuration to re-create an endpoint and set the **Binding** to `POST`.

---

11 Click **OK**.

22.14.4 Configuring the SAML 2.0 event on Advanced Authentication

1 Open the Advanced Authentication Administration portal.

2 Click **Events > Add.**

3 Create an event with the following parameters:

- Name: GitLab
- Chains: select the required chains.
- Paste the content of the file `https://<adfs_hostname>/FederationMetadata/2007-06/FederationMetadata.xml` to the **SP SAML 2.0 meta data**.

  or

  - Click **Choose File** 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, then you have an issue on ADFS that you need to resolve.

---

4 Click **Policies > Web Authentication.**

5 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. Configure an external .
2. Specify the address in **External URL** instead of specifying an address of a single Advanced Authentication server.

6 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.

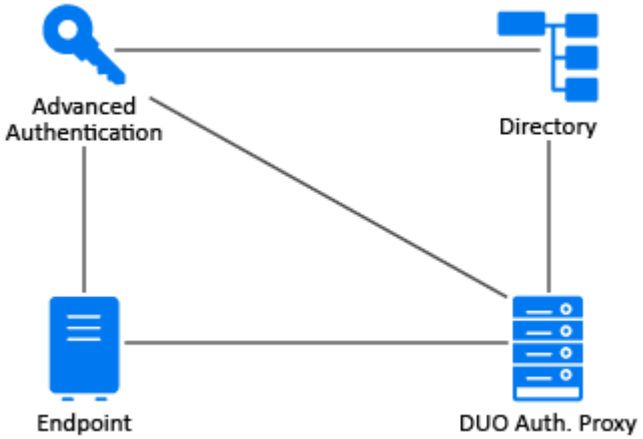
22.15 Configuring integration with Filr

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

For more information about using Advanced Authentication with Filr, see the Filr documentation (https://www.microfocus.com/documentation/filr/filr-4/filr-bp-maint/index.html?page=/documentation/filr/filr-4/filr-bp-maint/multi-factor-advanced-auth.html).

22.16 Configuring integration with DUO authentication proxy

This section provides the configuration information of Advanced Authentication integration with the DUO Authentication Proxy. The following diagram represents Advanced Authentication in DUO Authentication Proxy.



22.16.1 Configuring the Advanced Authentication RADIUS client

1 Open the Advanced Authentication Administration portal.

2 Click **Methods > RADIUS Client**.

3 Specify the IP address of DUO Proxy server in **Server**.

4 Specify the secret key for DUO Authentication Proxy in **Secret**.

5 Specify the port to where the RADIUS authentication request is sent. The default port is 1812.

6 Click **Save**.

7 Click **Chain > New Chain**.

8 Specify Radius Client in **Name**.

9 Set **Is enabled** to **ON**.

10 Move **LDAP Password** and **RADIUS Client** from Available to Used list

11 Specify the repositories, roles and groups in **Repos, Roles and Groups**.

12 Click **Save**.

13 Click **Events > Authenticators Management**.

14 Move **LDAP Password** and **RADIUS Client** from Available to Used list.

15 Set **Show the chain selection** to **ON**.

16 Click **Save**.

22.16.2 Configuring the DUO authentication proxy

1 Sign-in to the DUO Authentication Proxy as an administrator.

2 Click **Applications > RADIUS**.

3 Specify a shared secret keyword in **Secret Key**.

4 Specify the Advanced Authentication IP address in **API Hostname**.

5 Specify the port. The default port is 1812.

6 Click **Save**.

22.17 Configuring integration with ArcSight

This section provides the information on integrating Advanced Authentication with ArcSight and to achieve single sign-on (SSO) to ArcSight.

To configure the integration of Advanced Authentication with ArcSight, perform the following tasks:

- Configuring ArcSight
- Configuring the SAML 2.0 event on Advanced Authentication
- Authenticating on ArcSight with SAML 2.0

22.17.1 Configuring ArcSight

1 On the NFS server, open the `sso-configuration.properties` file, located by default in the `<arcsight_nfs_vol_path>/sso/default` directory.

`<arcsight_nfs_vol_path>` is the nfs volume used for CDF installation.

For example: /opt/NFS_volume/arcsight-volume. This location might vary based on the version of ArcSight.

2 In the configuration directory, open the `sso-configuration.properties` file and add the following properties:

```
com.microfocus.sso.default.login.method = saml2
com.microfocus.sso.default.saml2.enabled = true
com.microfocus.sso.default.login.saml2.mapping-attr = mail
com.microfocus.sso.default.login.saml2.identifierFormat = emailAddress
```

3 Download the SAML2 metadata from Advanced Authentication server.The URL to download the metadata:

```
https://<AA Server hostname>/osp/a/<Tenant Name>/auth/saml2/metadata
```

4 Convert the metadata xml file to base64 string and set the following variable:

```
com.microfocus.sso.default.login.saml2.metadata = <base64 encoded
metadata xml>
```

5 Save the changes in the `sso-configuration.properties` file.

Ensure, there are no additional spaces at the end of properties.

6 Restart the pod to apply the new configuration.

- Get the pod information using following command:

  ```
  kubectl get pods --all-namespaces | grep fusion-single-sign-on
  ```

- Delete the current running pod using following command:

  ```
  kubectl delete pod fusion-single-sign-on-xxxxxxxxxx-xxxxx -n
  arcsight-installer-xxxxx
  ```

New pod is initiated with new configuration.

7 Retrieve the Fusion SSO SAML service provider metadata from the server.

```
https://EXTERNAL_ACCESS_HOST/osp/a/default/auth/saml2/spmetadata
```

where, `EXTERNAL_ACCESS_HOST` is the hostname of the server.

This metadata must be uploaded in Advanced authentication SAML2 configuration.

For more information, see Configuring SAML Authentication in ArcSight (https://www.microfocus.com/documentation/arcsight/arcsight-platform-22.1/arcsight-admin-guide-22.1/#deployment_post_perform/saml_config.htm).

22.17.2 Configuring the SAML 2.0 event on Advanced Authentication

1 Open the Advanced Authentication Administration portal.

2 Click **Events > New Event.**

3 Create an event with the following parameters:

- **Name**: specify a preferred name that indicates the use of this event. For example, ArcSight
- **Event Type**: SAML2
- **Chains**: select the required chains.
- **SP SAML 2.0 meta data**: Paste the content of the file `https://EXTERNAL_ACCESS_HOST/osp/a/default/auth/saml2/spmetadata`

  or

  - Click **Choose File** and upload the saved XML file.

4 Select **Send E-Mail as NameID (suitable for G-Suite)** from **NameID formatting options**.

5 Click **Save**.

22.17.3 Authenticating on ArcSight with SAML 2.0

Open the ArcSight login page, the page redirects to the Advanced Authentication server, where the user must authenticate. After successful authentication, the Advanced Authentication server redirects the user back to ArcSight Dashboard page.

22.18 Configuring integration with Azure

This section provides the configuration information on integrating Advanced Authentication with Azure MFA. This integration secures the connection with Advanced Authentication verification methods and allow users succeed the methods to seamlessly access Azure services.

The following diagram represents integration of Advanced Authentication with Azure.



To configure the integration of Advanced Authentication with Azure, perform the following tasks:

- Configuring Advanced Authentication SAML 2.0 event
- Configuring ADFS
- Authenticating on Azure

Ensure that the following requirements are met:

- Create an account in Azure.
- Install Azure AD connect and synchronize the directory with cloud account.
- Register the custom domain name and verify that through public DNS registrar service.

22.18.1 Configuring Advanced Authentication SAML 2.0 event

1 Click **Events > New Event** to add a new event in the Administration portal.

2 Create an event with the following parameters:

- **Name**: Office 365
- **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.

3 Click **Save**.

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

4 Click **Policies > Web Authentication**.

5 Set the External URL to https://AdvancedAuthenticationServerAddress/ and replace AdvancedAuthenticationServerAddress with domain name or IP address of your Advanced Authentication server.

6 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.

7 Click **Save**.

22.18.2 Configuring ADFS

1 Open the ADFS management console.

2 Click **Claims Provider Trusts > Add Claims Provider trust**.

3 Click **Start** in the **Add Claims Provider Trust Wizard**.

4 Click **Import data about the claims provider from a file** in the **Select Data Source** tab.

5 Browse the **Federation metadata file**.

You can download the Federation metadata from the Advanced Authentication metadata URL: `https://<aaf-server>/osp/a/TOP/auth/saml2/metadata`.

6 Click **Next**.

7 Specify the **Display name**.

8 Click **Next**.

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

10 Click **Close**.

11 Right-click the **Display name** and click **Edit Claim Rules**.

12 Click **Add Rule**.

13 Select **Send Claims Using a Custom Rule from Claim rule template** in the **Add Transform Claim Rule Wizard**.

14 Click **Next**.

15 Specify the **Claim rule name**.

16 Paste the following in **Custom rule**:

```
c:[Type == "netbiosName"] => issue(Type = "http://
schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname",
Issuer = c.Issuer, OriginalIssuer = c.OriginalIssuer, Value = c.Value,
ValueType = c.ValueType);
```

17 Click **OK**.

22.18.3 Authenticating on Azure

1. Launch https://portal.azure.com//.
2. Login with your credentials.
3. Select Advanced Authentication to go through the multi-factor authentication.

   Page redirects to the SAML Login page.
4. You must pass the specified chains for authentication.

22.19 Configuring integration with Amazon Web Services Single Sign-On

This section includes the configuration required to integrate Advanced Authentication with Amazon Web Services Single Sign-On (AWS SSO). This integration secures the connection with Advanced Authentication verification methods and allow users to seamlessly access AWS services after the successful SAML authentication.

After the integration, Advanced Authentication serves as an Identity Provider and AWS Single Sign-On serves as a Service Provider.

To configure the integration of Advanced Authentication with AWS Single Sign-On, perform the following tasks:

- Downloading the SAML metadata of Advanced Authentication
- Setting-up AWS Single Sign-On
- Configuring a SAML 2.0 event on Advanced Authentication
- Verifying the integration

Ensure that the following requirements are met:

- AWS SSO is intended for organizations. Use your existing organization or create one.

  For more information, see Create an Organization (https://docs.aws.amazon.com/organizations/latest/userguide/orgs_manage_org_create.html).
- Make sure the repository that you are using contains users with email addresses specified. You cannot use the LOCAL repository in Advanced Authentication because the local users do not have email address.
- Add users' details to AWS.

  NOTE: The username of each user in AWS must be user's email address else you cannot map the user account. If the mapping is not valid then SAML authentication might fail.

22.19.1 Downloading the SAML metadata of Advanced Authentication

1. Navigate to **Policies > Web Authentication** on the Administration Portal.
2. Click **Download IdP SAML 2.0 Metadata**.

A new tab launches with the SAML 2.0 metadata that includes the certificate in x.509 format.

3 Save the file in `.xml` format. Keep the file for further use.

22.19.2 Setting-up AWS Single Sign-On

1 Log in to AWS Management Console with your organization management credentials.

2 Open AWS SSO Console.

3 Select **Enable AWS SSO**.

If you have not yet created AWS Organizations, a prompt to create an organization appears.

4 Click **Go to Settings**.

5 Under **Identity source** section, click **Actions > Change identity source**.

6 Select **External Identity Provider** and click **Next**.

7 Click **Download metadata file** and save the file for further use.

The AWS SSO SAML metadata file must be uploaded in Advanced Authentication that serves as the external identity provider.

8 Under **Identity provider metadata**, click **Choose file**, and select the metadata file that you downloaded from Advanced Authentication Step 3.

9 Click **Next**.

10 Specify `ACCEPT` to confirm the change and click **Change identity source**.

22.19.3 Configuring a SAML 2.0 event on Advanced Authentication

1 Click **Events > Add** on the Administration Portal.

2 Specify the following details:

2a **Name**: AWS SSO (or any other string).

2b **Chains**: select the required chains.

2c Under SAML 2.0 settings, click **Choose File** and select the metadata of AWS Single Sign-On that you obtained in Step 7.

2d Select **Send E-Mail as NameID (suitable for G-Suite)** from **NameID formatting options**. This is required to prevent errors in the SAML response.

3 Click **Save**.

22.19.4 Verifying the integration

1 Ensure that users are available on Advanced Authentication server and AWS account of your organization.

2 Navigate to **AWS SSO > Dashboard**.

3 Copy the **User portal URL** under **Settings summary**.

4 Navigate to the User portal URL in a browser and check whether you are redirected to Advanced Authentication login page.

5 Complete the authentication as per the chains configured in the SAML event.

For example, specify the user credentials in **Username** and **Password** if Password Only chain is available.

6 Check whether you can access the Single Sign-On page.

V Maintaining Advanced Authentication

This chapter contains the following sections:

- Chapter 23, “Logging,” on page 345
- Chapter 24, “Reporting,” on page 381
- Chapter 25, “Searching a card holder’s information,” on page 383
- Chapter 26, “Troubleshooting,” on page 385
- Chapter 27, “General best practices,” on page 393

23 Logging

Advanced Authentication provides the logging functionality. All administrative and user actions and events are logged.

Logs help to debug a problem based on the event or action performed.

The log rotation is hard coded based on the file size. The maximum size of a log file is 20 MB. For RADIUS logs, the size of the file is 50 MB. For WebAuth logs, the size of the file is 10 MB. Advanced Authentication stores the last ten log files of each type.

Advanced Authentication supports the following types of logs:

- Syslog
- RADIUS logs
- Async logs
- Long tasks logs
- Long scheduler logs
- Fingerprint logs
- Risk Service logs

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.

The **Debug logging** is set to **OFF** by default. You can enable **Debug logging** to generate detailed logs for each events and the logs are available in the respective tabs.

After enabling the **Debug logging**, you can apply the change to other Advanced Authentication servers in a cluster with the **Apply to all Servers** option.

Exporting the logs

To export logs, perform the following steps:

1. Click **Logs**.
2. Select the log you want to export.
3. Click **Export**.
4. Specify a **Start date** and **End date** to determine the required logging period.
5. Click **Export**.

   The exported log files are displayed in the **File Name** section.
6. Click the exported log file package that is exported in the format `aucore-logs_<logging_period>.tar.gz` to download it.

Clearing the logs

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 logs to save as a backup before you delete them.

2. Click **OK** to clear the logs.

23.1 Syslog

These logs contain information about the system events and actions. The log message is displayed in the format:

```
<PRI>VERSION TIMESTAMP HOSTNAME APP-NAME PROCID MSGID STRUCTURED-DATA
CEF:Version|Device Vendor|Device Product|Device Version|Device Event Class
ID|Name|Severity|[Extension]
```

On the server, the Syslog is stored in `/var/log/messages`.

After you export the logs, you can find the messages file in the `\var\log\host\` of the exported logs packages.

NOTE: Previous versions of Advanced Authentication were not aligned to the standards of CEF. `CEF Name`, `Severity` and `Extension` have been changed to conform to the standard. New logs are available when you enable ArcSight CEF standard in **Policies > CEF log forward**. Disabling this policy allows you to use older versions of CEF. Ensure that any existing CEF integration is familiar with this change.

The CEF extensions are mapped as follows:

ArcSight CEF Field	Advanced Authentication Event Field	Field Type
dvc	device address	Required
dvchost	device host name	Required
dvcpid	device process id	Required
dtz	device time zone	Required
rt	device receipt time	Required
cs	custom string- Depends on the event	Optional
flexString		
deviceCustomDate1	custom date - Depends on the event	Optional
deviceExternalId	endpoint id	Optional

ArcSight CEF Field	Advanced Authentication Event Field	Field Type
duser	destination user name	Optional
externalId	session id	Optional
oldFileId	Depends on the event	Optional
outcome	Display the outcome, ‘success’ or ‘failure’	Optional
reason	The reason an audit event was generated	Optional
sourceServiceName	endpoint name	Optional
src	endpoint address	Optional
suser	source user name	Optional

For more information about Syslog rules, see The Syslog Protocol (https://www.rfc-editor.org/rfc/rfc5424.html).

For more information about CEF rules, see Implement ArcSight Common Event Format (CEF) - Version 26 (https://www.microfocus.com/documentation/arcsight/arcsight-smartconnectors-8.3/cef-implementation-standard/).

The Syslogs are classified as follows:

- 0 - 99: Maintenance
- 100 - 199: Access
- 200 - 299: App data
- 300 - 399: Endpoints
- 400 - 499: Repositories
- 500 - 599: Local Users
- 600 - 699: Repository Users
- 700 - 799: User templates
- 800 - 899: Policies
- 900 - 999: Licenses
- 1000 - 1099: Settings
- 1100 - 1199: Password filter
- 1200 - 1299: Cached logon
- 1300 - 1399: Events
- 1400 - 1499: Chains
- 1500 - 1599: Identity validations

To monitor the risk related audit logs, see Monitoring risk audit logs.

Code	Name	Class	Severity	Optional Parameters	Example
2	Request failed	operational	4	duser, externalId, outcome, reason	CEF:0|NetIQ|AA|6.5.1.0|2|Request failed|4|duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=failure reason=request fail dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
10	Server started	operational	4		CEF:0|NetIQ|AA|6.5.1.0|10|Server started|4|dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
12	Server stopped	operational	7		CEF:0|NetIQ|AA|6.5.1.0|12|Server stopped|7|dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
13	Server unexpectedly stopped	operational	9		CEF:0|NetIQ|AA|6.5.1.0|13|Server unexpectedly stopped|9|dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
50	Server Message	operational	4	outcome, reason	CEF:0|NetIQ|AA|6.5.1.0|50|Server Message|4|reason=unknown event 125 dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
100	User logon started	security	1	cs1(tenant_id), cs2(tenant_name), cs3(method_id), cs4(event_name), cs5(unit_id), duser, externalId, sourceServiceName, src	CEF:0|NetIQ|AA|6.5.1.0|100|User logon started|1|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=PASSWORD:1 cs3Label=method_id cs4=AdminUI cs4Label=event_name cs5=PSlpIe12Jn30JpXLSzXWfKRzwLpHV2nu cs5Label=unit_id duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W sourceServiceName=SampleEp src=10.20.22.23 dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275

Code	Name	Class	Severity	Optional Parameters	Example
101	User was successfully logged on	security	1	cs1(tenant_id), cs2(tenant_name), cs3(method_id), cs4(event_name), cs5(template_owner), cs6(chain_name), duser, externalId, flexString1(method_info), outcome, sourceServiceName, src	CEF:0|NetIQ|AA|6.5.1.0|101|User successfully logged on|1|cs1=def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=PASSWORD:1 cs3Label=method_id cs4=AdminUI cs4Label=event_name cs5=LOCAL\USER cs5Label=template_owner cs6=password-chain cs6Label=chain_name duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W flexString1=shared-authenticator-used flexString1Label=method_info outcome=success sourceServiceName=SampleEp src=10.20.22.23 dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
102	User was failed to authenticate	security	4	cs1(tenant_id), cs2(tenant_name), cs3(method_id), cs4(event_name), duser, externalId, outcome, reason, sourceServiceName, src	CEF:0|NetIQ|AA|6.5.1.0|102|User failed to authenticate|4|cs1=def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=PASSWORD:1 cs3Label=method_id duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=failure reason=PASSWORD_WRONG sourceServiceName=SampleEp src=10.20.22.23 dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275

Code	Name	Class	Severity	Optional Parameters	Example
103	User was switched logo method	security	2	cs1(tenant_id), cs2(tenant_name), cs3(method_id), cs4(event_name), duser, externalId, oldFileId(old_method_id), outcome, sourceServiceName, src	CEF:0|NetIQ|AA|6.5.1.0|103|User switched logon method|2|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=SMARTPHONE:1 cs3Label=method_id duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W oldFileId=PASSWORD:1 outcome=success sourceServiceName=SampleEp src=10.20.22.23 dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
104	User logon session ended	security	2	cs1(tenant_id), cs2(tenant_name), cs4(event_name), duser, externalId, sourceServiceName, src	CEF:0|NetIQ|AA|6.5.1.0|104|User logon session ended|2|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W sourceServiceName=SampleEp src=10.20.22.23 dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
105	User cancelled the logon	security	7	cs1(tenant_id), cs2(tenant_name), cs3(method_id), duser, externalId, sourceServiceName, src	CEF:0|NetIQ|AA|6.5.1.0|105|User canceled the logon|7|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=SMARTPHONE:1 cs3Label=method_id duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W sourceServiceName=SampleEp src=10.20.22.23 dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275

Code	Name	Class	Severity	Optional Parameters	Example
106	User failed to switch logon method	security	4	cs1(tenant_id), cs2(tenant_name), cs3(method_id), cs4(event_name), duser, externalId, outcome, reason, sourceServiceName, src	CEF:0|NetIQ|AA|6.5.1.0|106|User failed to switch logon method|4|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=SMARTPHONE:1 cs3Label=method_id duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=failure reason=transaction aborted sourceServiceName=SampleEp src=10.20.22.23 dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
107	User locked	security	7	cs1(tenant_id), cs2(tenant_name), cs3(method_id), cs4(event_name), duser, externalId, reason, sourceServiceName, src	CEF:0|NetIQ|AA|6.5.1.0|107|User locked|7|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=SMARTPHONE:1 cs3Label=method_id duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W sourceServiceName=SampleEp src=10.20.22.23 reason=Too many authentication failures dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
108	User logout	security	4	cs1(tenant_id), cs2(tenant_name), cs3(event_name), duser, externalId	CEF:0|NetIQ|AA|6.5.1.0|108|User logout|4|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=AdminUI cs3Label=event_name dtz=Etc/UTC duser=LOCAL\\ADMIN dvc=127.0.0.1 dvchost=dev-comp dvcpid=391 externalId=b6f8db3c82aa11f0b5630242ac110002 outcome=success rt=1756233203352

Code	Name	Class	Severity	Optional Parameters	Example
200	User read data	security	3	cs1(tenant_id), cs2(tenant_name), cs3(data_id), cs4(record_id), duser, externalId, sourceServiceName, src	CEF:0|NetIQ|AA|6.5.1.0|200|User read data|3|cs1=def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=OSLogon cs3Label=data_id cs4=WtxZyc6bynIFdKOw02Fgm CQUAEcFuua0 cs4Label=record_id duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNY NlGgpRpd7W sourceServiceName=SampleEp src=10.20.22.23 dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
201	User wrote data	security	4	cs1(tenant_id), cs2(tenant_name), cs3(data_id), cs4(record_id), duser, externalId, sourceServiceName, src	CEF:0|NetIQ|AA|6.5.1.0|201|User wrote data|4|cs1=def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=OSLogon cs3Label=data_id cs4=WtxZyc6bynIFdKOw02Fgm CQUAEcFuua0 cs4Label=record_id duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNY NlGgpRpd7W sourceServiceName=SampleEp src=10.20.22.23 dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
300	Endpoint created	security	4	cs1(tenant_id), cs2(tenant_name), cs3(endpoint_name), deviceExternalId, duser, outcome, src	CEF:0|NetIQ|AA|6.5.1.0|300|Endpoint created|4|cs1=def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=SampleEp cs3Label=endpoint_name deviceExternalId=F6EP7N0eIqKWjn28z QXi7cQcRNIMuT2m duser=LOCAL\USER outcome=success src=10.20.22.23 dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275

Code	Name	Class	Severity	Optional Parameters	Example
301	No rights to create endpoint	security	7	cs1(tenant_id), cs2(tenant_name), cs3(endpoint_name), deviceExternalId, duser, src	CEF:0|NetIQ|AA|6.5.1.0|301|No rights to create endpoint|7|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=SampleEp cs3Label=endpoint_name deviceExternalId=F6EP7N0eIqKWjn28zQXi7cQcRNIMuT2m duser=LOCAL\USER src=10.20.22.23 dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
302	Failed to create endpoint	operational	7	cs1(tenant_id), cs2(tenant_name), cs3(endpoint_name), deviceExternalId, duser, outcome, reason, src	CEF:0|NetIQ|AA|6.5.1.0|302|Failed to create endpoint|7|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=SampleEp cs3Label=endpoint_name deviceExternalId=F6EP7N0eIqKWjn28zQXi7cQcRNIMuT2m duser=LOCAL\USER outcome=failure reason=transaction aborted src=10.20.22.23 dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
303	Endpoint removed	security	4	cs1(tenant_id), cs2(tenant_name), cs3(endpoint_name), deviceExternalId, duser, outcome, src	CEF:0|NetIQ|AA|6.5.1.0|303|Endpoint removed|4|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=SampleEp cs3Label=endpoint_name deviceExternalId=F6EP7N0eIqKWjn28zQXi7cQcRNIMuT2m duser=LOCAL\USER outcome=success src=10.20.22.23 dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275

Code	Name	Class	Severity	Optional Parameters	Example
304	No rights to remove endpoint	security	7	cs1(tenant_id), cs2(tenant_name), cs3(endpoint_name), deviceExternalId, duser, src	CEF:0|NetIQ|AA|6.5.1.0|304|No rights to remove endpoint|7|cs1=def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=SampleEp cs3Label=endpoint_name deviceExternalId=F6EP7N0eIqKWjn28zQXi7cQcRNIMuT2m duser=LOCAL\USER src=10.20.22.23 dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
305	Failed to remove endpoint	operational	7	cs1(tenant_id), cs2(tenant_name), cs3(endpoint_name), deviceExternalId, duser, outcome, reason, src	CEF:0|NetIQ|AA|6.5.1.0|305|Failed to remove endpoint|7|cs1=def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=SampleEp cs3Label=endpoint_name deviceExternalId=F6EP7N0eIqKWjn28zQXi7cQcRNIMuT2m duser=LOCAL\USER outcome=failure reason=transaction aborted src=10.20.22.23 dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
306	Endpoint session started	operational	1	cs1(tenant_id), cs2(tenant_name), cs3(endpoint_name), deviceExternalId, outcome, src	CEF:0|NetIQ|AA|6.5.1.0|306|Endpoint session started|1|cs1=def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=SampleEp cs3Label=endpoint_name deviceExternalId=F6EP7N0eIqKWjn28zQXi7cQcRNIMuT2m outcome=success src=10.20.22.23 dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275

Code	Name	Class	Severity	Optional Parameters	Example
307	Endpoint session ended	operational	1	cs1(tenant_id), cs2(tenant_name), cs3(endpoint_name), deviceExternalId, outcome, src	CEF:0|NetIQ|AA|6.5.1.0|307|Endpoint session ended|1|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=SampleEp cs3Label=endpoint_name deviceExternalId=F6EP7N0eIqKWjn28zQXi7cQcRNIMuT2m outcome=success src=10.20.22.23 dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
308	Invalid endpoint session secret	security	7	cs1(tenant_id), cs2(tenant_name), cs3(endpoint_name), deviceExternalId, src	CEF:0|NetIQ|AA|6.5.1.0|308|Invalid endpoint session secret|7|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=SampleEp cs3Label=endpoint_name deviceExternalId=F6EP7N0eIqKWjn28zQXi7cQcRNIMuT2m src=10.20.22.23 dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
309	Failed to create endpoint session	operational	7	cs1(tenant_id), cs2(tenant_name), cs3(endpoint_name), deviceExternalId, outcome, reason, src	CEF:0|NetIQ|AA|6.5.1.0|309|Failed to create endpoint session|7|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=SampleEp cs3Label=endpoint_name deviceExternalId=F6EP7N0eIqKWjn28zQXi7cQcRNIMuT2m outcome=failure reason=transaction aborted src=10.20.22.23 dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
310	Failed to end endpoint session	operational	7	cs1(tenant_id), cs2(tenant_name), cs3(endpoint_name), deviceExternalId, outcome, reason, src	CEF:0|NetIQ|AA|6.5.1.0|310|Failed to end endpoint session|7|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=SampleEp cs3Label=endpoint_name deviceExternalId=F6EP7N0eIqKWjn28zQXi7cQcRNIMuT2m outcome=failure reason=transaction aborted src=10.20.22.23 dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275

Code	Name	Class	Severity	Optional Parameters	Example
311	Endpoint changed	security	4	cs1(tenant_id), cs2(tenant_name), cs3(endpoint_name), deviceExternalId, duser, outcome, src	CEF:0|NetIQ|AA|6.5.1.0|311|Endpoint changed|4|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=SampleEp cs3Label=endpoint_name deviceExternalId=F6EP7N0eIqKWjn28zQXi7cQcRNIMuT2m duser=LOCAL\USER outcome=success src=10.20.22.23 dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
312	Failed to change endpoint	operational	7	cs1(tenant_id), cs2(tenant_name), cs3(endpoint_name), deviceExternalId, duser, outcome, reason, src	CEF:0|NetIQ|AA|6.5.1.0|312|Failed to change endpoint|7|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=SampleEp cs3Label=endpoint_name deviceExternalId=F6EP7N0eIqKWjn28zQXi7cQcRNIMuT2m duser=LOCAL\USER outcome=failure reason=transaction aborted src=10.20.22.23 dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
313	Endpoint re-created	security	4	cs1(tenant_id), cs2(tenant_name), cs3(endpoint_name), deviceExternalId, duser, oldFileId(old_endpoint_id), outcome, src	CEF:0|NetIQ|AA|6.5.1.0|313|Endpoint re-created|4|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=SampleEp cs3Label=endpoint_name deviceExternalId=F6EP7N0eIqKWjn28zQXi7cQcRNIMuT2m duser=LOCAL\USER oldFileId=AZXSCViJjJc2bukT3mUkORc0BoJevQ67 outcome=success src=10.20.22.23 dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275

Code	Name	Class	Severity	Optional Parameters	Example
314	Failed to re-create endpoint	operational	7	cs1(tenant_id), cs2(tenant_name), cs3(endpoint_name), deviceExternalId, duser, oldFileId(old_endpoint_id), outcome, reason, src	CEF:0|NetIQ|AA|6.5.1.0|314|Failed to re-create endpoint|7|cs1=def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=SampleEp cs3Label=endpoint_name deviceExternalId=F6EP7N0eIqKWjn28zQXi7cQcRNIMuT2m duser=LOCAL\USER oldFileId=AZXSCViJjJc2bukT3mUkORc0BoJevQ67 outcome=failure reason=transaction aborted src=10.20.22.23 dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
401	Repository created	operational	4	cs1(tenant_id), cs2(tenant_name), cs3(repo_name), cs4(repo_type), duser, externalId, outcome	CEF:0|NetIQ|AA|6.5.1.0|401|Repository created|4|cs1=def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=LOCAL cs3Label=repo_name cs4=LDAP cs4Label=repo_type duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=success dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
402	Failed to create repository	operational	7	cs1(tenant_id), cs2(tenant_name), cs3(repo_name), cs4(repo_type), duser, externalId, outcome, reason	CEF:0|NetIQ|AA|6.5.1.0|402|Failed to create repository|7|cs1=def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=LOCAL cs3Label=repo_name cs4=LDAP cs4Label=repo_type duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=failure reason=transaction aborted dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275

Code	Name	Class	Severity	Optional Parameters	Example
403	Repository removed	operational	4	cs1(tenant_id), cs2(tenant_name), cs3(repo_name), cs4(repo_type), duser, externalId, outcome	CEF:0|NetIQ|AA|6.5.1.0|403|Repository removed|4|cs1=def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=LOCAL cs3Label=repo_name cs4=LDAP cs4Label=repo_type duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=success dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
404	Failed to remove repository	operational	7	cs1(tenant_id), cs2(tenant_name), cs3(repo_name), cs4(repo_type), duser, externalId, outcome, reason	CEF:0|NetIQ|AA|6.5.1.0|404|Failed to remove repository|7|cs1=def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=LOCAL cs3Label=repo_name cs4=LDAP cs4Label=repo_type duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=failure reason=transaction aborted dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
405	Repository configuration changed	operational	4	cs1(tenant_id), cs2(tenant_name), cs3(repo_name), cs4(repo_type), duser, externalId, outcome	CEF:0|NetIQ|AA|6.5.1.0|405|Repository configuration changed|4|cs1=def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=LOCAL cs3Label=repo_name cs4=LDAP cs4Label=repo_type duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=success dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275

Code	Name	Class	Severity	Optional Parameters	Example
406	Failed to change repository configuration	operational	7	cs1(tenant_id), cs2(tenant_name), cs3(repo_name), cs4(repo_type), duser, externalId, outcome, reason	CEF:0|NetIQ|AA|6.5.1.0|406|Failed to change repository configuration|7|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=LOCAL cs3Label=repo_name cs4=LDAP cs4Label=repo_type duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=failure reason=transaction aborted dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
501	Local user created	operational	4	cs1(tenant_id), cs2(tenant_name), duser, externalId, outcome, suser	CEF:0|NetIQ|AA|6.5.1.0|501|Local user created|4|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=success suser=LOCAL\ADMIN dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
502	Local user removed	operational	4	cs1(tenant_id), cs2(tenant_name), duser, externalId, outcome, suser	CEF:0|NetIQ|AA|6.5.1.0|502|Local user removed|4|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=success suser=LOCAL\ADMIN dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
503	Failed to create local user	operational	7	cs1(tenant_id), cs2(tenant_name), duser, externalId, outcome, reason, suser	CEF:0|NetIQ|AA|6.5.1.0|503|Failed to create local user|7|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=failure reason=transaction aborted suser=LOCAL\ADMIN dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275

Code	Name	Class	Severity	Optional Parameters	Example
504	No rights to remove local user	security	7	cs1(tenant_id), cs2(tenant_name), duser, externalId, suser	CEF:0|NetIQ|AA|6.5.1.0|504|No rights to remove local user|7|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W suser=LOCAL\ADMIN dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
505	Failed to remove local user	operational	7	cs1(tenant_id), cs2(tenant_name), duser, externalId, outcome, reason, suser	CEF:0|NetIQ|AA|6.5.1.0|505|Failed to remove local user|7|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=failure reason=transaction aborted suser=LOCAL\ADMIN dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
506	No rights to create local user	security	7	cs1(tenant_id), cs2(tenant_name), duser, externalId, suser	CEF:0|NetIQ|AA|6.5.1.0|506|No rights to create local user|7|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W suser=LOCAL\ADMIN dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
507	Local user changed	operational	4	cs1(tenant_id), cs2(tenant_name), duser, externalId, outcome, suser	CEF:0|NetIQ|AA|6.5.1.0|507|Local user changed|4|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=success suser=LOCAL\ADMIN dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275

Code	Name	Class	Severity	Optional Parameters	Example
508	Failed to change local user	operational	7	cs1(tenant_id), cs2(tenant_name), duser, externalId, outcome, reason, suser	CEF:0|NetIQ|AA|6.5.1.0|508|Failed to change local user|7|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=failure reason=transaction aborted suser=LOCAL\ADMIN dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
601	User created	operational	4	cs1(tenant_id), cs2(tenant_name), cs3(repo_name), duser, externalId, outcome, suser	CEF:0|NetIQ|AA|6.5.1.0|601|User created|4|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=LOCAL cs3Label=repo_name duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=success suser=LOCAL\ADMIN dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
602	No rights to create user	security	7	cs1(tenant_id), cs2(tenant_name), cs3(repo_name), duser, externalId, suser	CEF:0|NetIQ|AA|6.5.1.0|602|No rights to create user|7|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=LOCAL cs3Label=repo_name duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W suser=LOCAL\ADMIN dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
603	Failed to create user	operational	7	cs1(tenant_id), cs2(tenant_name), cs3(repo_name), duser, externalId, outcome, reason, suser	CEF:0|NetIQ|AA|6.5.1.0|603|Failed to create user|7|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=LOCAL cs3Label=repo_name duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=failure reason=transaction aborted suser=LOCAL\ADMIN dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275

Code	Name	Class	Severity	Optional Parameters	Example
604	User removed	operational	4	cs1(tenant_id), cs2(tenant_name), duser, externalId, outcome, suser	CEF:0|NetIQ|AA|6.5.1.0|604|User removed|4|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=success suser=LOCAL\ADMIN dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
605	No rights to remove user	security	7	cs1(tenant_id), cs2(tenant_name), cs3(repo_name), duser, externalId, suser	CEF:0|NetIQ|AA|6.5.1.0|605|No rights to remove user|7|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=LOCAL cs3Label=repo_name duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W suser=LOCAL\ADMIN dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
606	Failed to remove user	operational	7	cs1(tenant_id), cs2(tenant_name), duser, externalId, outcome, reason, suser	CEF:0|NetIQ|AA|6.5.1.0|606|Failed to remove user|7|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=failure reason=transaction aborted suser=LOCAL\ADMIN dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
607	Role granted to user	security	4	cs1(tenant_id), cs2(tenant_name), cs3(role_name), duser, externalId, outcome, suser	CEF:0|NetIQ|AA|6.5.1.0|607|Role granted to user|4|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=ENROLL ADMINS cs3Label=role_name duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=success suser=LOCAL\ADMIN dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275

Code	Name	Class	Severity	Optional Parameters	Example
608	Failed to grant role to user	security	7	cs1(tenant_id), cs2(tenant_name), cs3(role_name), duser, externalId, outcome, reason, suser	CEF:0|NetIQ|AA|6.5.1.0|608|Failed to grant role to user|7|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=ENROLL ADMINS cs3Label=role_name duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=failure reason=transaction aborted suser=LOCAL\ADMIN dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
609	Role revoked from user	security	4	cs1(tenant_id), cs2(tenant_name), cs3(role_name), duser, externalId, outcome, suser	CEF:0|NetIQ|AA|6.5.1.0|609|Role revoked from user|4|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=ENROLL ADMINS cs3Label=role_name duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=success suser=LOCAL\ADMIN dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
610	Failed to revoke role from user	security	7	cs1(tenant_id), cs2(tenant_name), cs3(role_name), duser, externalId, outcome, reason, suser	CEF:0|NetIQ|AA|6.5.1.0|610|Failed to revoke role from user|7|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=ENROLL ADMINS cs3Label=role_name duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=failure reason=transaction aborted suser=LOCAL\ADMIN dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275

Code	Name	Class	Severity	Optional Parameters	Example
611	User unlocked	operational	4	cs1(tenant_id), cs2(tenant_name), duser, externalId, outcome, suser	CEF:0|NetIQ|AA|6.5.1.0|611|User unlocked|4|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=success suser=LOCAL\ADMIN dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
612	Failed to unlock user	operational	7	cs1(tenant_id), cs2(tenant_name), duser, externalId, outcome, reason, suser	CEF:0|NetIQ|AA|6.5.1.0|612|Failed to unlock user|7|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=failure reason=transaction aborted suser=LOCAL\ADMIN dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
701	Template was assigned to the user	security	4	cs1(tenant_id), cs2(tenant_name), cs3(method_id), cs4(template_owner), cs5(comment), duser, externalId, outcome, suser	CEF:0|NetIQ|AA|6.5.1.0|701|Template was assigned to the user|4|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=PASSWORD:1 cs3Label=method_id cs4=LOCAL\USER cs4Label=template_owner cs5=Sample cs5Label=comment duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W suser= LOCAL\ADMIN outcome=success dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275

Code	Name	Class	Severity	Optional Parameters	Example
702	Template was enrolled for the user	security	4	cs1(tenant_id), cs2(tenant_name), cs3(method_id), cs4(template_owner), cs5(comment), duser, externalId, outcome, suser	CEF:0|NetIQ|AA|6.5.1.0|702|Template was enrolled for the user|4|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=PASSWORD:1 cs3Label=method_id cs4=LOCAL\USER cs4Label=template_owner cs5=Sample cs5Label=comment duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W suser= LOCAL\ADMIN outcome=success dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
703	User enrolled the assigned template	security	4	cs1(tenant_id), cs2(tenant_name), cs3(method_id), cs4(template_owner), cs5(comment), duser, externalId, outcome, suser	CEF:0|NetIQ|AA|6.5.1.0|703|User enrolled the assigned template|4|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=PASSWORD:1 cs3Label=method_id cs4=LOCAL\USER cs4Label=template_owner cs5=Sample cs5Label=comment duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W suser= LOCAL\ADMIN outcome=success dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
704	Template linked	security	4	cs1(tenant_id), cs2(tenant_name), cs3(method_id), cs4(template_owner), cs5(comment), duser, externalId, outcome, suser	CEF:0|NetIQ|AA|6.5.1.0|704|Template linked|4|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=PASSWORD:1 cs3Label=method_id cs4=LOCAL\USER cs4Label=template_owner cs5=Sample cs5Label=comment duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W suser= LOCAL\ADMIN outcome=success dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275

Code	Name	Class	Severity	Optional Parameters	Example
705	Failed to assign template to the user	security	7	cs1(tenant_id), cs2(tenant_name), cs3(method_id), cs4(template_owner), cs5(comment), duser, externalId, outcome, reason, suser	CEF:0|NetIQ|AA|6.5.1.0|705|Failed to assign template to the user|7|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=PASSWORD:1 cs3Label=method_id cs4=LOCAL\USER cs4Label=template_owner cs5=Sample cs5Label=comment duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W suser= LOCAL\ADMIN outcome=failure reason=transaction aborted dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
706	Failed to enroll template for the user	security	7	cs1(tenant_id), cs2(tenant_name), cs3(method_id), cs4(template_owner), cs5(comment), duser, externalId, outcome, reason, suser	CEF:0|NetIQ|AA|6.5.1.0|706|Failed to enroll template for the user|7|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=PASSWORD:1 cs3Label=method_id cs4=LOCAL\USER cs4Label=template_owner cs5=Sample cs5Label=comment duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W suser= LOCAL\ADMIN outcome=failure reason=transaction aborted dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
707	User failed to enroll the assigned template	security	7	cs1(tenant_id), cs2(tenant_name), cs3(method_id), cs4(template_owner), cs5(comment), duser, externalId, outcome, reason, suser	CEF:0|NetIQ|AA|6.5.1.06.5.1.0|707|User failed to enroll the assigned template|7|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=PASSWORD:1 cs3Label=method_id cs4=LOCAL\USER cs4Label=template_owner cs5=Sample cs5Label=comment duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W suser= LOCAL\ADMIN outcome=failure reason=transaction aborted dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275

Code	Name	Class	Severity	Optional Parameters	Example
708	Failed to link template	security	7	cs1(tenant_id), cs2(tenant_name), cs3(method_id), cs4(template_owner), cs5(comment), duser, externalId, outcome, reason, suser	CEF:0|NetIQ|AA|6.5.1.0|708|Failed to link template|7|cs1=def0def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=PASSWORD:1 cs3Label=method_id cs4=LOCAL\USER cs4Label=template_owner cs5=Sample cs5Label=comment duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W suser= LOCAL\ADMIN outcome=failure reason=transaction aborted dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
709	Template link removed	security	4	cs1(tenant_id), cs2(tenant_name), cs3(method_id), cs4(template_owner), cs5(comment), duser, externalId, outcome, suser	CEF:0|NetIQ|AA|6.5.1.0|709|Template link removed|4|cs1=def0def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=PASSWORD:1 cs3Label=method_id cs4=LOCAL\USER cs4Label=template_owner cs5=Sample cs5Label=comment duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W suser= LOCAL\ADMIN outcome=success dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
710	Failed to remove template link	security	7	cs1(tenant_id), cs2(tenant_name), cs3(method_id), cs4(template_owner), cs5(comment), duser, externalId, outcome, reason, suser	CEF:0|NetIQ|AA|6.5.1.0|710|Failed to remove template link|7|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=PASSWORD:1 cs3Label=method_id cs4=LOCAL\USER cs4Label=template_owner cs5=Sample cs5Label=comment duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W suser= LOCAL\ADMIN outcome=failure reason=transaction aborted dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275

Code	Name	Class	Severity	Optional Parameters	Example
711	Template removed	security	4	cs1(tenant_id), cs2(tenant_name), cs3(method_id), cs4(template_owner), cs5(comment), duser, externalId, outcome, suser	CEF:0|NetIQ|AA|6.5.1.0|711|Template removed|4|cs1=def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=PASSWORD:1 cs3Label=method_id cs4=LOCAL\USER cs4Label=template_owner cs5=Sample cs5Label=comment duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W suser= LOCAL\ADMIN outcome=success dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
712	Failed to remove template	security	7	cs1(tenant_id), cs2(tenant_name), cs3(method_id), cs4(template_owner), cs5(comment), duser, externalId, outcome, reason, suser	CEF:0|NetIQ|AA|6.5.1.0|712|Failed to remove template|7|cs1=def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=PASSWORD:1 cs3Label=method_id cs4=LOCAL\USER cs4Label=template_owner cs5=Sample cs5Label=comment duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W suser= LOCAL\ADMIN outcome=failure reason=transaction aborted dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
713	Template changed	security	4	cs1(tenant_id), cs2(tenant_name), cs3(method_id), cs4(template_owner), cs5(comment), duser, externalId, outcome, suser	CEF:0|NetIQ|AA|6.5.1.0|713|Template changed|4|cs1=def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=PASSWORD:1 cs3Label=method_id cs4=LOCAL\USER cs4Label=template_owner cs5=Sample cs5Label=comment duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W suser= LOCAL\ADMIN outcome=success dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275

Code	Name	Class	Severity	Optional Parameters	Example
714	Failed to change template	security	7	cs1(tenant_id), cs2(tenant_name), cs3(method_id), cs4(template_owner), cs5(comment), duser, externalId, outcome, reason, suser	CEF:0|NetIQ|AA|6.5.1.0|714|Failed to change template|7|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=PASSWORD:1 cs3Label=method_id cs4=LOCAL\USER cs4Label=template_owner cs5=Sample cs5Label=comment duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W suser= LOCAL\ADMIN outcome=failure reason=transaction aborted dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
715	Template changed during logon	security	4	cs1(tenant_id), cs2(tenant_name), cs3(method_id), cs4(template_owner), cs5(comment), duser, externalId, suser	CEF:0|NetIQ|AA|6.5.1.0|715|Template changed during logon|4|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=PASSWORD:1 cs3Label=method_id cs4=LOCAL\USER cs4Label=template_owner cs5=Sample cs5Label=comment duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W suser= LOCAL\ADMIN dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
801	Policy changed	security	4	cs1(tenant_id), cs2(tenant_name), cs3(component_id), cs4(scope), duser, externalId, outcome	CEF:0|NetIQ|AA|6.5.1.0|801|Policy changed|4|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=LoginOptions cs3Label=component_id cs4=global cs4Label=scope duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=success dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275

Code	Name	Class	Severity	Optional Parameters	Example
802	No rights to change policy	security	7	cs1(tenant_id), cs2(tenant_name), cs3(component_id), cs4(scope), duser, externalId	CEF:0|NetIQ|AA|6.5.1.0|802|No rights to change policy|7|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=LoginOptions cs3Label=component_id cs4=global cs4Label=scope duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
803	Failed to change policy	operational	7	cs1(tenant_id), cs2(tenant_name), cs3(component_id), cs4(scope), duser, externalId, outcome, reason	CEF:0|NetIQ|AA|6.5.1.0|803|Failed to change policy|7|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=LoginOptions cs3Label=component_id cs4=global cs4Label=scope duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=failure reason=transaction aborted dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
804	Object policy changed	security	4	cs1(tenant_id), cs2(tenant_name), cs3(component_id), cs4(scope), cs5(object_id), cs6(object_type), duser, externalId, flexString1(object_name), outcome	CEF:0|NetIQ|AA|6.5.1.0|804|Object policy changed|4|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=LoginOptions cs3Label=component_id cs4=object cs4Label=scope cs5=fc157e1cfe2f11ec81840242ac110002 cs5Label=object_id cs6=User cs6Label=object_type duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W flexString1=testUser flexString1Label=object_name outcome=success dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275

Code	Name	Class	Severity	Optional Parameters	Example
805	No rights to change object policy	security	7	cs1(tenant_id), cs2(tenant_name), cs3(component_id), cs4(scope), cs5(object_id), cs6(object_type), duser, externalId, flexString1(object_name)	CEF:0|NetIQ|AA|6.5.1.0|805|No rights to change object policy|7|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=LoginOptions cs3Label=component_id cs4=object cs4Label=scope cs5=fc157e1cfe2f11ec81840242ac110002 cs5Label=object_id cs6=User cs6Label=object_type duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W flexString1=testUser flexString1Label=object_name dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
806	Failed to change object policy	operational	7	cs1(tenant_id), cs2(tenant_name), cs3(component_id), cs4(scope), cs5(object_id), cs6(object_type), duser, externalId, flexString1(object_name), outcome, reason	CEF:0|NetIQ|AA|6.5.1.0|806|Failed to change object policy|7|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=LoginOptions cs3Label=component_id cs4=object cs4Label=scope cs5=fc157e1cfe2f11ec81840242ac110002 cs5Label=object_id cs6=User cs6Label=object_type duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W flexString1=testUser flexString1Label=object_name outcome=failure reason=transaction aborted dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275

Code	Name	Class	Severity	Optional Parameters	Example
901	License added	operational	4	cs1(tenant_id), cs2(tenant_name),cs3 (license_id), cs4(enabled_features), cs5(user_count), deviceCustomDate1 (expire_date), externalId, outcome	CEF:0|NetIQ|AA|6.5.1.0|901|License added|4|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=kAi22UNwgKJnldwQ30okbPRBduoveSD2 cs3Label=license_id cs4=super cs4Label=enabled_features cs5=42 cs5Label=user_count deviceCustomDate1=Dec 25 2022 20:30:00 deviceCustomDate1Label=expire_date externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=success dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
902	Failed to add license	operational	7	cs1(tenant_id), cs2(tenant_name), cs3(license_id), cs4(enabled_features), cs5(user_count), deviceCustomDate1(expire_date), externalId, outcome, reason	CEF:0|NetIQ|AA|6.5.1.0|902|Failed to add license|7|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=kAi22UNwgKJnldwQ30okbPRBduoveSD2 cs3Label=license_id cs4=super cs4Label=enabled_features cs5=42 cs5Label=user_count deviceCustomDate1=Dec 25 2022 20:30:00 deviceCustomDate1Label=expire_date externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=failure reason=transaction aborted dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
1001	Global setting changed	security	7	cs1(tenant_id), cs2(tenant_name), cs3(setting_name), duser, externalId, outcome	CEF:0|NetIQ|AA|6.5.1.0|1001|Global setting changed|7|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=HTTPCert cs3Label=setting_name duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=success dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275

Code	Name	Class	Severity	Optional Parameters	Example
1002	No rights to change global setting	security	9	cs1(tenant_id), cs2(tenant_name), cs3(setting_name), duser, externalId	CEF:0|NetIQ|AA|6.5.1.0|1002|No rights to change global setting|9|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=HTTPCert cs3Label=setting_name duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
1003	Failed to change global setting	operational	9	cs1(tenant_id), cs2(tenant_name), cs3(setting_name), duser, externalId, outcome, reason	CEF:0|NetIQ|AA|6.5.1.0|1003|Failed to change global setting|9|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=HTTPCert cs3Label=setting_name duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=failure reason=transaction aborted dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
1101	Password changed	security	3	cs1(tenant_id), cs2(tenant_name), duser, sourceServiceName, src	CEF:0|NetIQ|AA|6.5.1.0|1101|Password changed|3|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name duser=LOCAL\USER sourceServiceName=SampleEp src=10.20.22.23 dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
1102	Password reset	security	6	cs1(tenant_id), cs2(tenant_name), duser, sourceServiceName, src	CEF:0|NetIQ|AA|6.5.1.0|1101|Password reset|6|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name duser=LOCAL\USER sourceServiceName=SampleEp src=10.20.22.23 dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275

Code	Name	Class	Severity	Optional Parameters	Example
1201	User logged on using local cache	security	1	cs1(tenant_id), cs2(tenant_name), cs3(event_name), cs4(chain_name), deviceCustomDate1(logon_time), duser, sourceServiceName, src	CEF:0|NetIQ|AA|6.5.1.0|1201|User logged on using local cache|1|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=Portal cs3Label=event_name cs4=password-chain cs4Label=chain_name deviceCustomDate1=1660662337275 deviceCustomDate1Label=logon_time duser=LOCAL\USER sourceServiceName=SampleEp src=10.20.22.23 dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
1301	Event created	security	4	cs1(tenant_id), cs2(tenant_name), cs3(event_name), duser, externalId, outcome	CEF:0|NetIQ|AA|6.5.1.0|1301|Event created|4|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=Portal cs3Label=event_name duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=success dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
1302	Failed to create event	operational	7	cs1(tenant_id), cs2(tenant_name), cs3(event_name), duser, externalId, outcome, reason	CEF:0|NetIQ|AA|6.5.1.0|1302|Failed to create event|7|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=Portal cs3Label=event_name duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=failure reason=transaction aborted dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
1303	Event changed	security	4	cs1(tenant_id), cs2(tenant_name), cs3(event_name), duser, externalId, outcome	CEF:0|NetIQ|AA|6.5.1.0|1303|Event changed|4|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=Portal cs3Label=event_name duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=success dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275

Code	Name	Class	Severity	Optional Parameters	Example
1304	Failed to change event	operational	7	cs1(tenant_id), cs2(tenant_name), cs3(event_name), duser, externalId, outcome, reason	CEF:0|NetIQ|AA|6.5.1.0|1304|Failed to change event|7|cs1=def0def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=Portal cs3Label=event_name duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=failure reason=transaction aborted dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
1305	Event removed	security	4	cs1(tenant_id), cs2(tenant_name), cs3(event_name), duser, externalId, outcome	CEF:0|NetIQ|AA|6.5.1.0|1305|Event removed|4|cs1=def0def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=Portal cs3Label=event_name duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=success dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
1306	Failed to remove event	operational	7	cs1(tenant_id), cs2(tenant_name), cs3(event_name), duser, externalId, outcome, reason	CEF:0|NetIQ|AA|6.5.1.0|1306|Failed to remove event|7|cs1=def0def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=Portal cs3Label=event_name duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=failure reason=transaction aborted dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
1401	Chain created	security	4	cs1(tenant_id), cs2(tenant_name), cs3(chain_name), duser, externalId, outcome	CEF:0|NetIQ|AA|6.5.1.0|1401|Chain created|4|cs1=def0def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=password-chain cs3Label=chain_name duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=success dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275

Code	Name	Class	Severity	Optional Parameters	Example
1402	Failed to create chain	operational	7	cs1(tenant_id), cs2(tenant_name), cs3(chain_name), duser, externalId, outcome, reason	CEF:0|NetIQ|AA|6.5.1.0|1402|Failed to create chain|7|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=password-chain cs3Label=chain_name duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=failure dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275 reason=transaction aborted
1403	Chain changed	security	4	cs1(tenant_id), cs2(tenant_name), cs3(chain_name), duser, externalId, outcome	CEF:0|NetIQ|AA|6.5.1.0|1403|Chain changed|4|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=password-chain cs3Label=chain_name duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=success dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
1404	Failed to change chain	operational	7	cs1(tenant_id), cs2(tenant_name), cs3(chain_name), duser, externalId, outcome, reason	CEF:0|NetIQ|AA|6.5.1.0|1404|Failed to change chain|7|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=password-chain cs3Label=chain_name duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=failure dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275 reason=transaction aborted
1405	Chain removed	security	4	cs1(tenant_id), cs2(tenant_name), cs3(chain_name), duser, externalId, outcome	CEF:0|NetIQ|AA|6.5.1.0|1405|Chain removed|4|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=password-chain cs3Label=chain_name duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=success dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275

Code	Name	Class	Severity	Optional Parameters	Example
1406	Failed to remove chain	operational	7	cs1(tenant_id), cs2(tenant_name), cs3(chain_name), duser, externalId, outcome, reason	CEF:0|NetIQ|AA|6.5.1.0|1406|Failed to remove chain|7|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=password-chain cs3Label=chain_name duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=failure dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275 reason=transaction aborted
1501	HANIS validation succeeded	security	4	cs1(tenant_id), cs2(tenant_name), cs3(method_id), cs4(id_number), cs5(phone_number),duser, external_id, outcome, sourceServiceName, src	CEF:0|NetIQ|AA|6.5.1.0|1501|HANIS validation succeeded|4|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=HANIS:1 cs3Label=method_id cs4=92***86 cs4Label=id_number cs5=+123456789 cs5Label=phone_number duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=success sourceServiceName=SampleEp src=10.20.22.23 dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275
1502	Failed to validate HANIS	operational	7	cs1(tenant_id), cs2(tenant_name), cs3(method_id), cs4(id_number), cs5(phone_number),duser, external_id, outcome, reason, sourceServiceName, src	CEF:0|NetIQ|AA|6.5.1.0|1502|Failed to validate HANIS|7|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=HANIS:1 cs3Label=method_id cs4=92***86 cs4Label=id_number cs5=+123456789 cs5Label=phone_number duser=LOCAL\USER externalId=G861nae15NAVC4JoxkTkNYNlGgpRpd7W outcome=success reason=FACE_TOO_SMALL sourceServiceName=SampleEp src=10.20.22.23 dvc=127.0.0.1 dvchost=dev-comp dvcpid=21 dtz=UTC rt=1660662337275

Code	Name	Class	Severity	Optional Parameters	Example
1601	Token created	security	4	cs1(tenant_id), cs2(tenant_name), cs3(guid), cs4(claims), cs5(scopes), cs6(ttl)	CEF:0|NetIQ|AA|6.5.1.0|1601|Token created|4|cs1=def0def0def0def0def0def0def0def0def0 cs1Label=tenant_id cs2=TOP cs2Label=tenant_name cs3=a2788596f607d4352999c8da474961577246dbaa cs3Label=guid cs4={'tenant_name': 'TOP', 'category_name': None, 'redirect_uri': '/account/', 'event_name': 'Authenticators Management'} cs4Label=claims cs6=0:10:00 cs6Label=ttl dtz=Etc/UTC dvc=10.204.96.19 dvchost=vlab003104.dom003100.lab dvcpid=229 outcome=success rt=1757345752611
1602	Token deleted	operational	7	cs3(guid), cs4(operation)	CEF:0|NetIQ|AA|6.5.1.0|1601|Token Created|4|cs3=a2788596f607d4352999c8da474961577246dbaa cs3Label=guid cs4= delete cs4Label=operation dtz=Etc/UTC dvc=10.204.96.19 dvchost=vlab003104.dom003100.lab dvcpid=229 outcome=success rt=1757345752611

23.2 RADIUS logs

These logs contain information about the logs that are recorded for the RADIUS server.

On the server, the `radius.log` file is stored in the `/var/lib/docker/volumes/aaf_radiusd-logs/_data/` directory.

After you export the RADIUS logs, you can find the `radius.log` file in the `/var/log/freeradius/` directory.

23.3 Async logs

These logs contain information about the asynchronized delivery of OTP messages for the SMS, Email, and Voice methods.

On the server, the `async_commander.log` and `async_commander.*.log` files are stored in the `/var/lib/docker/volumes/aaf_aucore-logs/_data/` directory.

After you export the Async logs, you can find the `async_commander.log` and `async_commander.*.log` files in the `/opt/AuCore/logs/` directory.

23.4 Long tasks logs

These logs contain information about the celery long tasks for exporting the backup files.

On the server, the `celery_long.log` file is stored in the `/var/lib/docker/volumes/aaf_aucore-logs/_data/` directory.

After you export the Long Tasks logs, you can find the `celery_long.log` file in the `/opt/AuCore/logs/` directory.

23.5 Long scheduler logs

These logs contain information about the queue tasks of scheduled export.

On the server, the `celery_long_beat.log` file is stored in the `/var/lib/docker/volumes/aaf_aucore-logs/_data/` directory.

After you export the Long Scheduler logs, you can find the `celery_long_beat.log` file in the `/opt/AuCore/logs/` directory.

23.6 Fingerprint logs

These logs contain all details from a Fingerprint service.

On the server, the `nbisd.log` file is stored in the `/var/lib/docker/volumes/aaf_afisd-logs/_data/` directory.

After you export the Fingerprint logs, you can find `nbisd.log` in the `\var\log\nbisd\` directory.

23.7 Risk Service logs

The `riskservice.b<string>.log` file contains the logs of Risk Service.

On the server, the `riskservice.b<string>.log` file is stored in the `/var/log/risk` directory.

After you export the logs, you can find the log file in the `fluentd/log/` directory.

Risk audit logs include information about the Risk Service events.

To export the Risk Audit logs, perform the following steps:

1. On the server, navigate to the `/var/log/risk` directory.
2. Copy the `auditlogs.<string>.log.gz` file.

24 Reporting

Advanced Authentication facilitates you to add and view reports according to your requirement. You can view information about the memory utilization, tenant information, successful or failed logins, licenses, and so forth in a graphical representation. You can also export these reports to JSON and CSV formats.

To log in to the Advanced Authentication Reporting portal, launch the URL: `https://<NetIQServer>/report` and log in with the FULL ADMIN credentials.

NOTE: To view reports, the user must have the FULL ADMIN role. You can view the reports of a specific tenant or all tenants on the Reporting Portal.

For more information, see “Adding a Report” section.

25 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.

   ---

26 Troubleshooting

NOTE: This chapter contains solutions for known issues. If you encounter any problems that are not mentioned here, contact the support service.

This chapter contains the following topics:

- Section 26.1, “Administration Portal is accessible without any authentication,” on page 385
- Section 26.2, “The ON/OFF switch is broken if the screen resolution is 110%,” on page 385
- Section 26.3, “Users can login using the old Password,” on page 386
- Section 26.4, “Issue with authenticating on Office 365,” on page 386
- Section 26.5, “Error while downloading logs package,” on page 386
- Section 26.6, “Error while configuring SMS OTP method,” on page 386
- Section 26.7, “Configuring the log rotation in Docker before deploying the Advanced Authentication server,” on page 387
- Section 26.8, “Error while logging in to Salesforce,” on page 387
- Section 26.9, “Analyzing performance issue using the profiling tool,” on page 387
- Section 26.10, “Validating JSON syntax in SLAnalyzer,” on page 388
- Section 26.11, “Push messages does not Appear in Smartphone,” on page 388
- Section 26.12, “Insufficient allocated disk space,” on page 389
- Section 26.13, “Issue with cluster synchronization,” on page 390
- Section 26.14, “Users with very large usergroups attributes are being rejected by the NGINX reverse proxy,” on page 390
- Section 26.15, “Error while loading the dashboard data,” on page 391

26.1 Administration Portal is accessible without any authentication

Issue: After authenticating to the enrollment portal, if a user switches to the Administration portal, access is granted without any authentication prompt.

Workaround: You must disable the Kerberos SSO option for the **Report logon** and **Admin UI** events.

26.2 The ON/OFF switch is broken if the screen resolution is 110%

While trying to edit the **Lockout options** policy, the **ON/OFF** switch is broken when the screen resolution is 110%.

As a solution, change the screen resolution to 100%.

26.3 Users can login using the old Password

Issue: When users use the **LDAP Password only** chain for authentication and change their LDAP password, they are still able to log in with their old LDAP password.

Workaround: You must disable the cache logon on Domain Controllers. To disable the cache logon, you must make the following registry changes:

1. Open the registry key: `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\.`
2. Create a `DWORD` parameter `OldPasswordAllowedPeriod` and set the parameter's value to `0`.

26.4 Issue with authenticating on Office 365

Issue: When authenticating to Microsoft teams and Outlook apps on smartphone, the OpenText claim provider fails.

Reason: By default, Azure Active Directory prompts for a fresh authentication with username and password and OpenText is unable to handle it.

Workaround: In Azure Active Directory, set the value of `PromptLoginBehaviour` to `NativeSupport`.

For more information see the Microsoft documentation (https://docs.microsoft.com/en-us/windows-server/identity/ad-fs/operations/ad-fs-prompt-login).

26.5 Error while downloading logs package

Issue: When the user tries to download the logs packages, an error message `401 Authorization Required/ openresty` is displayed.

Workaround: Right-click the log package and select **Save As**.

26.6 Error while configuring SMS OTP method

Issue: While performing SMS OTP configuration with Clickatell, an error message `Attribute Error: 'NoneType' object has no attribute 'strip'` is displayed.

Workaround: Remove **Password** from the **Parameters**

26.7 Configuring the log rotation in Docker before deploying the Advanced Authentication server

It is recommended to configure the log rotation in docker for deploying Advanced Authentication server on the public cloud. You can also define the maximum size of logs to address increasing log size.

Perform the following steps to rotate docker logs in docker:

1. Navigate to the `/etc/docker/daemon.json` path.
2. Set the parameter `log-driver` with the name of logging driver. By default, log-driver is set to `json-file`.

   For example, `"log-driver": "json-file"`
3. To configure maximum size of logs and number of log files use the key `log-opts` in `daemon.json`:

   ```
   "log-driver": "json-file", "log-opts": { "max-size": "10m", "max-file": "3" }
   ```

26.8 Error while logging in to Salesforce

Issue: When users log in to Salesforce and succeed all configured methods of Advanced Authentication, an error message `Single Sign-on error` is displayed. This is due to an invalid SAML assertion configured in the Single Sign-On Settings of the Salesforce console.

Workaround: Perform the following steps:

1. Navigate to Settings > Identity > Single Sign-On Settings in the Salesforce console.
2. Set **SAML Identity Type** to **Assertion contains the User's Salesforce username**.
3. Specify `IDPEmail` in **Attribute Name**.
4. Click **Save**.

26.9 Analyzing performance issue using the profiling tool

The Profiling tool facilitates you to evaluate performance of Advanced Authentication appliance and retrieve the detailed trace of any API request. The resultant data after evaluation helps you to enhance the performance of API calls execution. The Advanced Authentication server includes the Profiling tool.

The tool is disabled by default. Before enabling the Profiling tool, it is required to enable **Debug logging** on the **Logs** page of Advanced Authentication Administration portal. To enable the Profiling tool, append the following parameter to the API call that you want to trace:

```
?profiling=true
```

For example, to trace the records of OS logon data, append the parameter to the API call as follows:

```
api/v1/users/4f34e2882991440ddd0fd515e0d0236c/data/
OSLogon?login_session_id=rBT79CAz8AWh1o920OrHumx32iaToCU9&profiling=true
```

26.10 Validating JSON syntax in SLAnalyzer

You can check the JSON syntax in SLAnalyzer. By checking the syntax, you figure out the errors in the syntax, and know whether the syntax can be processed or not.

1. Download the SLAnalyzer by the `https://ftp.novell.com/pub/SLAnalyzer/SLAnalyzerDownloadManager.exe` path.
2. Run the program.
3. Select **JSON Console** in **Tools** tab.
4. Specify the syntax in the top Window of the **JSON Consol**.
5. Right-click on the syntax and select **Process**.

If the JSON syntax can be processed, JSON data will be displayed in the bottom Window.

26.11 Push messages does not Appear in Smartphone

Issue: When a user initiates authentication using the Smartphone method, push message does not appear on the smartphone screen. However, when the user launches the OpenText Advanced Authentication application manually, the authentication request appears and the user is able to authenticate.

Reason: This issue occurs due to the following reasons:

1. Your firewall lacks connectivity to `proxy.authasas.com` by HTTPS.
2. Temporary problems with Apple or Google push services.
3. You have a traffic inspector or another third-party software that re-signs certificates.

Solution 1: Perform the following steps for respective reasons:

1. Check the firewall settings. See Configuring Firewall (https://www.netiq.com/documentation/advanced-authentication-64/server-administrator-guide/data/firewall.html).
2. Contact Support.
3. Upload a root certificate to the AAF trusted store:
   - **3a** Copy certificate to Advanced Authentication server.
   - **3b** Copy certificate to `aaf-aucore-1` container:

     ```
     docker cp <certificatefile> aaf-aucore-1:/etc/pki/trust/anchors/
     ```

   - **3c** Update certificate using the following command:

     ```
     docker exec -it aaf-aucore-1 /bin/bash -c "update-ca-certificates;
     ln -s /var/lib/ca-certificates/ca-bundle.pem /var/lib/ca-
     certificates/pem/ca-certificates.crt"
     ```

Solution 2: From Advanced Authentication 6.3.3, docker containers OS is changed from Debian to SUSE. Perform the following steps to resolve the issue:

1. Copy certificate to `aaf-aucore-1` container:

   ```
   docker cp <certificatefile in PEM> aaf-aucore-1:/etc/pki/trust/anchors/
   ```

2 Update the certificate

```
docker exec -it aaf-aucore-1 /bin/bash -c \"update-ca-certificates; ln -s /var/lib/ca- certificates/ca-bundle.pem /var/lib/ca-certificates/pem/ca-certificates.crt\"
```

26.12 Insufficient allocated disk space

Sometimes the allocated disk space is not enough. In that case, you can increase the disk space using one of the following ways:

- Clearing the log files
- Expanding the root partition

26.12.1 Clearing the log files

1 Search the log files using the command:

```
find / -iname *.log
```

And remove those files.

2 Search the old log files like `celery.log.1, uwsgi.log.1` etc. in `/var/lib/docker/volumes/aaf_aucore-logs/_data/`, and remove those files.

26.12.2 Expanding the root partition

1 Download the SLES 12 SP5 ISO from www.suse.com/download/sles (http://www.suse.com/download/sles) and select the `SLE-12-SP5-Server-DVD-x86_64-Boothole-Respin-DVD1.iso` file.

2 In VMware, **Virtual Machine Settings > Hardware** select **CD/DVD** and **Use ISO image file** then point to the downloaded ISO.

3 Restart. Press F2 to specify the BIOS settings.

4 Select the **Boot** tab in the PhoenixBIOS setup Utility window.

5 Select **CD-ROM Drive** and press the + key to move the **CD-ROM Drive** to the top.

6 Select the **Exit** tab and select **Exit Saving Changes**.

7 Select **More** in the boot menu.

8 Select **Rescue System** and select **Language**.

9 Log in to Rescue system as the Root user.

10 To confirm disk information, use the command:

```
fdisk -l
```

11 To resize and rescue disk, use the command:

```
parted
```

12 To get device partition information, use the command:

```
print.
```

13 If you need to resize device other than the one which parted selects, use the command:

```
select <device name>
```

For Example: `select /dev/ sdb`

14 To get disk information, use the command:

```
print
```

15 To resize the disk space, use the command:

```
resize
```

And specify the partition number and required space for the selected disk.

16 To exit, use command:

```
quit
```

17 Check the partition using the command:

```
e2fsck -f <device name>
```

For example: `e2fsck -f /dev/sbdb`

18 Expand the file system on the new partition using command:

```
resize2fs <device name>
```

For example: `resize2fs /dev/sbdb`

19 Reboot the system using command:

```
reboot
```

20 After rebooting, check the updated size of disk using command:

`df -h` or `fdisk -l`

26.13 Issue with cluster synchronization

Issue: The system often crashes, and the cluster synchronization fails frequently. This is due to heavy user traffic.

Workaround: Increase the amount of RAM and the number of processor cores.

26.14 Users with very large usergroups attributes are being rejected by the NGINX reverse proxy

Issue: After upgrading to Advanced Authentication 6.4 Service Pack 2, users with very large userGroup attributes are rejected by the nginx reverse proxy and are unable to log in.

Workaround: To resolve the login issue, perform the following steps:

1 Login to the SSH session on the Advanced Authentication server.

2 Run the following command to connect to the Aucore container:

```
docker exec aaf-aucore-1 bash
```

3 Run the following commands to open the authcfg.xml file:

```
vi /opt/AuCore/static/osp_templates/WEB-INF/conf/current/tenant/
services/authcfg.xml
```

4 Set the value of the `AttributeMapEntry cachable` parameter to “True” in the below elements:

- `<AttributeMapEntry cachable="true" localName="userGroups" nativeName="naafUserGroups"/>`
- `<AttributeMapEntry cachable="true" localName="roles" nativeName="user_role_assignments"/>`

5 Save the file and exit the docker container.

6 Run the following command to remove the .json file from the container:

```
rm /var/lib/docker/volumes/aaf_aucore-data/_data/osp_settings.json
```

NOTE: Wait for 5 minutes to allow the configuration to be rewritten.

26.15 Error while loading the dashboard data

Issue: After upgrading to Advanced Authentication 6.4 Service Pack 2, the dashboard occasionally fails to load the data and displays the following error:

```
Transport Error(503, 'Search Guard not initialized (SG11). See https://
docs.search-guard.com/latest/sgadmin')(Internal Server Error)
```

Reason: This is due to the corrupted ElasticSearch configuration on an Advanced Authentication appliance.

Workaround: Repair the corrupted Elasticsearch configuration.

IMPORTANT: It will reset the Elasticsearch container to its initial configuration and result in the loss of all previously gathered event data.

To repair a corrupted Elasticsearch installation on the AA appliance, perform the following steps:

1 (Optional) Run the following command to connect to the appliance using ssh:

```
systemctl start sshd
```

2 Run the following command on the command prompt to connect to the `searchd` container:

```
docker exec -it aaf-searchd-1 /bin/bash
```

3 Run the following commands to open the `elasticsearch.yaml` file:

```
vi /.backup/config/elasticsearch.yml
```

4 Add the following properties at end of the `elasticsearch.yaml` file:

```
action.auto_create_index:
".watches,.triggered_watches,.watcher[1]history-*"
```

5 Save the file and exit the docker container.

6 Run the following command to stop the Elasticsearch container from the Advanced Authentication appliance command prompt:

```
docker stop aaf-searchd-1
```

7 Validate the delete path before removing the data and configuration info from the containers that are used by the `aaf-searchd-1` container at the AA appliance command prompt. To remove the data and configuration info, run the below command:

```
rm -r /var/lib/docker/volumes/aaf_searchd-config/_data/*
rm -r /var/lib/docker/volumes/aaf_searchd-data/_data/*
```

8 Run the following command to restart the `aaf-searchd-1` container:

```
docker start aaf-searchd-1
```

9 Run the following command to connect to the `aaf-aucore-1` container:

```
docker exec -it aaf-aucore-1 /bin/bash
```

10 Change the directory for the `aaf-aucore-1` container at the command prompt to the scripts by using the below command:

```
cd /opt/AuCore/aucore/scripts
```

11 To set the module search path for python by exporting the `PYTHONPATH` environment variable, run the below command:

```
export PYTHONPATH=/opt/AuCore:/opt/penv/lib/python3.9/site[1]packages
```

12 Run the below command to execute the `au_setup_reporting.pyc` script to reconfigure elastic search:

```
python au_setup_reporting.pyc ../../data/production.ini
```

13 Verify the data on the Advanced Authentication dashboard to make sure the Elasticsearch container is configured properly and is recording event data.

NOTE: During the `au_setup_reporting.pyc` script execution, you may see several connection errors and warnings stating that elasticsearch is not ready. It should eventually print out the following lines:

```
a2023-11-27 22:20:27 INFO [elasticsearch] GET https://127.0.0.1:9200/
_cluster/health?wait_for_status=yellow[status:200 request:0.275s] 2023-
11-27 22:20:28 INFO [elasticsearch] PUT https://127.0.0.1:9200/aucore-
stats-2 [status:200 request:0.321s]
```

---

27 General best practices

This chapter provides a comprehensive set of recommendations for strengthening security and authentication practices for administrator accounts:

- Enrolling the administrator account for multi-factor authentication is recommended to make it more difficult to compromise this account.
- Ensure to control the administrator account with Privileged Account Management (PAM) solution. This provides compliance with the password rotation and complexity standards.
- Configure account lockout policies to prevent brute force attacks. Always align with the identity corporate source of truth that helps maintain consistency in user identity management.
- It is advisable to use named accounts instead of generic "**admin**" usernames for audit trail purposes.
- Disable unnecessary accounts and keep unused accounts in a dormant state or remove if not needed. However, make sure to keep any audit historic data.
- All accounts should be enrolled to at least one factor apart from the account password.
- It is recommended to have strong passwords that are managed and rotated.
- It is advised to have chains with more than one authentication mechanism and to be composed with methods that are a combination of the different authentication types "Something you have", "Something you are", and "Something you know".
- Use generic naming convention for chains rather than "Chain name: Method 1 + Method 2…." To avoid giving all the information about that methods are expected to successfully respond to the chain. This is to make it less predictable and defined to potential attackers.
- Provide a generic naming convention for authentication chains that does not determine how many methods are in the chain (for example, TOTP Only) and does not outline the methods in the chain (for example, LDAP Password + Smartphone).

  An example of a generic chain name could be "Fingerprint Authentication".
- We do not recommend any one authentication method in the authentication chain above all others. Best practice often dictates a blended configuration of various methods depending on the context which considers factors, such as trust, cost, user-friendliness, and information value.
- Always register a minimum of two authentication methods per account, so there is always a substitute when one is lost or hacked and removed.
- The Smartphone method is practical choice as a second factor for remote access. This can replace expensive hardware tokens with better security results.
- It is advisable to disable **Allow as first authentication method** for methods, such as Email OTP, SMS OTP, and Voice OTP to reduce the risk of spamming and guessing OTP.
- A practical approach to determine the authentication method that fits a specific use case or user group is by analyzing the associated business risk to the group in case of identity theft.
- Segregating traffic based on authentication types or events simplifies the effort to troubleshoot and trace authentications.

- It is advisable to only use valid SSL certificates for Smartphone authentication. Self-signed certificates are not recommended.
- Request for membership of the Advanced Authentication administrative roles and groups should be well-motivated and follow an approval process to maintain security and control over access.
- Consider stringent processes for testing, documentation, and promotion in any deployments of a multi-factor environment to ensure the stability of each environment. The DEV (Development), QA (Quality Assurance) and PROD (Production) approach must be followed.
- QA and PROD environments must be maintained by an environment “Gate Keeper” who has the role to vet the deployment and ensure that it verifies the naming conventions, standard best practices, documentation and all overlapping use cases are considered in each deployment.
- Change or remove default chains attached to the events, especially the out-of-box events. This improves the overall security of the deployment.
- Consider having the backup of local administrative account to allow for application recovery in case there is a disconnect to the Corporate Directory. This account should follow a very strong password strategies and be enrolled for multi-factor.
- Always use groups associated to the corporate directory in order to manage the level of access for different users (enrolled users, application administrators, and so on).
- Perform periodic access reviews on the associated corporate directory groups to ensure that only valid users have the correct access particularly to the group that can perform administrative functions.
- Prepare for off-line use of laptops and other devices that may have network coverage challenges. Consider authentication mechanisms that support such use cases.
- Be aware of man-in-the-middle attack. Avoid using the same data channel of delivery of the different 2FA.

27.1 Recommendations to prevent phishing attacks

A strong MFA (Multi-Factor Authentication) solution helps mitigate the risk of attacks like Evilginx. Here are some recommendations for a robust MFA implementation:

- **Choose a reliable MFA Method**: Select a combination of authentication factors that provide a high level of security. This can include something the user knows (for example, a password or PIN), something the user possesses (for example, a hardware token or smart card), or something the user is (for example, biometrics like fingerprints or facial recognition).
- **Avoid SMS-based OTPs**: SMS-based OTPs (One-Time Passwords) can be vulnerable to SIM swapping attacks or interception. However, consider using time-based OTPs generated through authenticator apps like Google Authenticator or Authy.
- **Physical Tokens**: Hardware tokens provide an additional layer of security. These physical devices generate unique OTPs that are synchronized with the authentication server. They are immune to attacks targeting software-based OTP generators.
- **Biometric Authentication**: Biometrics, such as fingerprints or facial recognition, can be used as an MFA factor. Biometrics are unique to each individual and can be difficult to replicate, enhancing the security of the authentication process.

- **Adaptive Authentication**: Implement adaptive authentication mechanisms that assess various risk factors, including device fingerprinting, IP reputation, user behavior, and geolocation. Adjust the level of authentication required based on the risk level associated with the user's context.
- **Single Sign-On (SSO) with MFA**: If the organization provides single sign-on solutions, ensure that MFA is enabled for SSO access. This adds an extra layer of security for accessing multiple applications with a single set of credentials.
- **Regularly Update and Patch**: Ensure to keep all MFA-related software and systems up-to-date with the latest security patches. This helps protect against known vulnerabilities and ensures the effectiveness of your MFA solution.
- **Awareness of Cyber Security**: Educate your users about the importance of MFA and how to use it effectively. Provide clear instructions on setting up and using MFA, and emphasize the significance of protecting their authentication factors.

NOTE: While implementing a strong MFA is crucial, it is also necessary to regularly monitor and assess the security of your systems, conduct security awareness training, and prepare for emerging threats and best practices to stay one step ahead of attackers.
