BIG-IP® Access Policy Manager®: Authentication and Single Sign-On

Version 11.5

Table of Contents

Legal Notices....13
Acknowledgments....15

Chapter 1: Authentication Concepts....19
About AAA server support....20
About AAA high availability support....20
About AAA and load balancing....20
About AAA traffic and route domains....20
About APM support for multiple authentication types....21
About APM certificate authentication support....21
About SSL certificates on the BIG-IP system....21
About local user database support....22
About guest access (one-time password) support....22
About authentication for Microsoft Exchange clients....22
Documentation for Access Policy Manager authentication....22

Chapter 2: Active Directory Authentication....25
About Active Directory authentication....26
About Active Directory password management....26
About AAA high availability....26
About how APM handles binary values in Active Directory attributes....27
Task summary for Active Directory authentication....27
Configuring an Active Directory AAA server....28
Creating an access profile....29
Configuring Active Directory authentication....29
Creating a virtual server....30
Testing AAA high availability for supported authentication servers....31
Example access policy using Active Directory authentication and query....31
Active Directory authentication session variables....32
Active Directory cross-domain support rules....32
Active Directory authentication and query troubleshooting tips....33
Overview: Using Active Directory Trusted Domains....34
Configuring an Active Directory Trusted Domain....34

Chapter 3: Active Directory Query....37
About Active Directory queries....38
About nested groups in Active Directory and LDAP queries....38
About Active Directory password management....38
About how APM handles binary values in Active Directory attributes....39
Adding an Active Directory query to an access policy....39

Using AD query with IPv6....40
Active Directory query session variables....42
Active Directory authentication and query troubleshooting tips....43

Chapter 4: LDAP and LDAPS Authentication....45
About LDAP and LDAPS authentication....46
About how APM handles binary values in LDAP attributes....46
About AAA high availability....47
Task summary for configuring for LDAPS authentication....47
Configuring an LDAPS AAA server in APM....48
Creating an access profile....48
Configuring LDAPS authentication....49
Creating a virtual server for LDAPS....50
Testing LDAPS authentication....50
Testing AAA high availability for supported authentication servers....51
Example of LDAP auth and query default rules....51
LDAP authentication session variables....51
UserDN settings in LDAP....52
LDAP authentication and query troubleshooting tips....52

Chapter 5: LDAP Query....55
About LDAP queries....56
About nested groups in Active Directory and LDAP queries....56
About how APM handles binary values in LDAP attributes....56
Adding an LDAP query to an access policy....57
Example of LDAP auth and query default rules....57
Session variables in LDAP query properties....58
LDAP query session variables....58
LDAP authentication and query troubleshooting tips....59

Chapter 6: RSA SecurID Authentication....61
About RSA SecurID authentication....62
About RSA SecurID configuration requirements for APM AAA....62
Task summary for configuring for RSA SecurID authentication....63
Configuring a SecurID AAA server in APM....63
Creating an access profile....64
Configuring RSA SecurID authentication in an access policy....64
Creating a virtual server....66
Access policy example for RSA and AD authentication....67
RSA SecurID session variables for access policy rules....68
RSA SecurID on Windows using RADIUS configuration troubleshooting tips....68

Chapter 7: RADIUS Authentication....71

About RADIUS authentication....72
About AAA high availability....72
Guidelines for setting up RADIUS authentication for AAA high availability....72
About how APM handles binary values in RADIUS attributes....73
Task summary for RADIUS authentication....73
Configuring a RADIUS AAA server in APM....74
Creating an access profile....74
Using RADIUS authentication in an access policy....75
Creating a virtual server....76
Testing AAA high availability for supported authentication servers....76
RADIUS attributes....77
RADIUS session variables for access policy rules....77
RADIUS authentication and accounting troubleshooting tips....78

Chapter 8: RADIUS Accounting....81
About RADIUS accounting....82
About how APM handles binary values in RADIUS attributes....82
Configuring a RADIUS Accounting server in APM....83
Adding RADIUS accounting to an access policy....84
RADIUS authentication and accounting troubleshooting tips....84

Chapter 9: Kerberos Authentication with End-User Logons....87
About basic authentication and Kerberos end-user logon....88
How does end-user logon work?....88
About Kerberos authentication requirements....89
Task summary for configuring end-user login support....90
Access policy example for end-user login....92
Kerberos authentication troubleshooting tips....93

Chapter 10: NTLM Authentication for Microsoft Exchange Clients....95
Overview: Configuring APM for Exchange clients that use NTLM authentication....96
About using NTLM authentication....96
About configuration requirements for NTLM authentication....96
About reusing a machine account for different BIG-IP systems....96
About Outlook Anywhere and NTLM authentication....97
Task summary for Exchange clients that use NTLM authentication....97

Chapter 11: HTTP Basic Authentication for Microsoft Exchange Clients....103
Overview: Configuring APM for Exchange clients that use HTTP Basic....104
About Exchange profiles....104
Task summary for Exchange clients that use HTTP Basic authentication....104

Chapter 12: HTTP and HTTPS Authentication....109

About HTTP AAA server authentication....110
Task summary for HTTP authentication....110
Configuring an AAA server for HTTP Basic/NTLM authentication....110
Configuring an HTTP AAA server for form-based authentication....111
Configuring an HTTP AAA server for custom post authentication....111
Creating an access profile....112
Using HTTP authentication in an access policy....113
Creating a virtual server....114
Overview: Configuring HTTPS authentication....114
Creating a pool for HTTPS authentication....115
Creating a virtual server for HTTPS authentication....115
Creating an access profile....116
Using HTTP authentication in an access policy....117
Adding the access profile to the virtual server....117

Chapter 13: Local User Database....119
Overview: Configuring and administering a local user database....120
About backing up and restoring users....120
About local user database synchronization across devices....120
About writing to a local user database from an access policy....120
Task summary....121
Overview: Using a local user database to control authentication....123
About locking a user out of an AAA server using a local user database....123
About writing to a local user database from an access policy....124
Task summary....124
Overview: Branching in an access policy based on local user database groups....127
Creating an access policy to branch based on local DB group membership....127

Chapter 14: OCSP Authentication....131
About OCSP authentication....132
Task summary for OCSP authentication....132
Configuring an AAA OCSP responder....132
Creating an access profile....133
Configuring OCSP authentication....133
Configuring a client SSL profile for OCSP....134
Adding client-side SSL and access profiles to a virtual server....134
Policy example for OCSP authentication....135
OCSP session variables....135
OCSP authentication troubleshooting tips....136

Chapter 15: CRLDP Authentication....137
About CRLDP configuration....138
About AAA high availability....138
Task summary for CRLDP configuration....138

Creating an access profile 139
Configuring an access policy that uses CRLDP authentication....140
Configuring a client SSL profile for CRLDP 141
Adding client-side SSL and access profiles to a virtual server....142
Testing AAA high availability for supported authentication servers....142
Example access policy for CRLDP authentication....143
CRLDP session variables....143
CRLDP authentication troubleshooting tips 144

Chapter 16: On-Demand Certificate Authentication....145
Overview: Requesting and validating an SSL certificate on demand....146
Creating a custom Client SSL profile....146
Adding On-Demand certificate authentication to an access policy....147
Adding client-side SSL and access profiles to a virtual server....147

Chapter 17: Client Certificate Inspection....149
About client certificate inspection....150
Task summary for client certificate inspection....150
Creating a client SSL profile for certificate inspection....150
Configuring an access policy to confirm client certificate validity....151

Chapter 18: One-Time Password Authentication....153
Overview: Providing a one-time password using email....154
Creating an SMTP server configuration....154
Creating an access policy to send an OTP using email....154
Overview: Providing a one-time password using an external SMS....156
Configuring HTTP form-based authentication to deliver a one-time password....157
Creating an access policy to send an OTP using an SMS....158

Chapter 19: TACACS+ Authentication and Accounting....161
About TACACS+ authentication and accounting....162
About AAA high availability....162
Task summary for TACACS+ authentication and accounting....162
Configuring a TACACS+ AAA server for authentication and authorization 163
Using TACACS+ authentication in an access policy....163
Testing AAA high availability for supported authentication servers....164
Example access policy for TACACS+ authentication and accounting....164
TACACS+ session variables for access policy rules....165
TACACS+ authentication troubleshooting tips 165

Chapter 20: AAA High Availability and Upgrade....167

Chapter 21: Configuring Single Sign-On with Access Policy Manager.....169
What is Single Sign-On?.....170

Chapter 22: Single Sign-On Methods.....171
What are the supported SSO methods?.....172
About the Single Sign-On configuration object.....172
Creating an HTTP Basic SSO configuration.....173
HTTP Basic SSO configuration settings 173
Creating an HTTP forms-based SSO configuration.....174
HTTP Form SSO configuration settings.....174
Creating an NTLMV1 SSO configuration.....176
NTLMV1 SSO configuration settings 176
Creating an NTLMV2 SSO configuration.....177
NTLMV2 SSO configuration settings 177

Chapter 23: Form-Based Client-Initiated Single Sign-On Method.....179
About form-based client-initiated SSO authentication.....180
Basic configuration of form-based client-initiated SSO 180
How does form-based client-initiated SSO authentication work by default? 180
About advanced configuration options for form-based client-initiated SSO authentication.....181
Configuring form-based client-initiated SSO.....182
Forms-based client-initiated SSO configuration settings.....183
Form-based client-initiated SSO configuration examples.....186
DWA form-based client-initiated SSO example.....186
Bugzilla form-based client-initiated SSO example.....186
Ceridian form-based client-initiated SSO example.....187
Citrix 4.5 and 5 form-based client-initiated SSO example.....189
Devcentral form-based client-initiated SSO example.....189
Google form-based client-initiated SSO example.....190
Oracle Application Server form-based client-initiated SSO example.....191
OWA 2010 and 2007 form-based client-initiated SSO example.....191
OWA 2003 form-based client-initiated SSO example.....192
Perforce form-based client-initiated SSO example.....192
Reviewboard form-based client-initiated SSO example.....193
SAP form-based client-initiated SSO example.....193
Salesforce form-based client-initiated SSO example.....194
Sharepoint 2010 form-based client-initiated SSO example.....195
Weblogin form-based client-initiated SSO example.....196
Yahoo form-based client-initiated SSO example.....196

Chapter 24: Kerberos Single Sign-On Method.....199

About Kerberos SSO....200
How does Kerberos SSO work in Access Policy Manager?....200
Task summary for configuring Kerberos SSO....200
Setting up a delegation account to support Kerberos SSO....201
Creating a Kerberos SSO configuration in APM....201
Editing an access policy to support Kerberos SSO....202
Binding a Kerberos SSO object to an access profile....203
Attaching an access profile to a virtual server for Kerberos SSO....203
Kerberos SSO configuration settings 203
Kerberos SSO session variable list....206
Tips for successfully deploying Kerberos SSO....206

Chapter 25: Single Sign-On and Multi-Domain Support....209
About multi-domain support for SSO....210
How does multi-domain support work for SSO?....210
Task summary for configuring domain support for SSO....212
Configuring an access policy for SSO single domain support....212
Configuring an access policy for SSO multi-domain support....212
Creating a virtual server for SSO multi-domain support....213

Chapter 26: Common Deployment Examples for Single Sign-On....215
Common use cases for Single Sign-On deployment....216
Task summary for configuring web application over network access tunnel for SSO....216
Configuring network access for SSO with web applications....216
Configuring network access properties....217
Configuring and managing the access profile using SSO....217
Configuring an HTTP virtual server for the network access....217
Configuring a layered virtual server for your web service....218
Configuring portal access resources for SSO 218

Chapter 27: Introducing Access Policy Manager SAML Support....221
About SAML....222
About SAML metadata....222
About SAML single logout service....222
About the benefits of using APM for SAML support....222
When should I configure a BIG-IP system as a SAML IdP? 223
When should I configure a BIG-IP system as a SAML service provider? 224
Overview: Exchanging certificates among SAML entities....224
Importing an SSL certificate 224
Exporting an SSL certificate....225

Chapter 28: Using APM as a SAML IdP (SSO portal)....227
Overview: Configuring a BIG-IP system as IdP with an SSO portal....228

About local IdP service....229
About SP connectors....229
What are the available ways I can configure a SAML SP connector?....229
Task summary....230
Flowchart: Configuration to support a SAML SSO portal....230
Creating a virtual server for a BIG-IP (as SAML IdP) system....231
Configuring SAML SP connectors....232
Configuring a full webtop....236
Configuring an access policy for a SAML SSO portal....236
Adding the access profile to the virtual server....237
Adding IdP metadata from APM to external SAML SPs....238

Chapter 29: Using APM as a SAML IdP (no SSO portal)....239
Overview: Configuring a BIG-IP system as IdP for SP-initiated connections only....240
About local IdP service....240
About SP connectors....240
What are the available ways I can configure a SAML SP connector?....241
Task summary....241
Flowchart: Configuration to support SP-initiated connections only....241
Creating a virtual server for a BIG-IP (as SAML IdP) system....242
Configuring SAML SP connectors....243
Configuring a SAML IdP service....244
Binding a SAML IdP service to multiple SP connectors....245
Exporting SAML IdP metadata from APM 246
Creating an access profile associated with the SAML IdP service 246
Configuring an access policy to provide authentication from the local IdP....247
Adding the access profile to the virtual server....247
Adding IdP metadata from APM to external SAML SPs....248

Chapter 30: Using APM as a SAML Service Provider....249
About configuration requirements for APM as a SAML service provider....250
About local SP service....250
About SAML IdP discovery....250
About IdP connectors....252
About methods for configuring SAML IdP connectors in APM....252
Task summary....253
Flowchart: BIG-IP system as a SAML service provider configuration....253
Configuring a custom SAML IdP connector....254
Creating a virtual server for a BIG-IP (as SAML SP) system....255
Configuring a SAML SP service....255
Binding a SAML SP service to SAML IdP connectors....256
Exporting SAML SP metadata from APM 257
Configuring an access policy to authenticate with an external SAML IdP....258
Adding the access profile to the virtual server....259

Adding SAML SP metadata from APM to an external SAML IdP.....259

Chapter 31: Using BIG-IP® IdP Automation.....261
Overview: Automating SAML IdP connector creation.....262
When would I use SAML IdP automation? 262
Automating IdP connector creation for BIG-IP as SP 262

Chapter 32: BIG-IP System Federation for SP-Initiated Connections.....265
Overview: Federating BIG-IP systems for SAML SSO (without an SSO portal).....266
About SAML IdP discovery.....266
About local IdP service.....268
About local SP service.....268
Task summary.....269
Flowchart: BIG-IP system federation configuration.....269
Setting up a BIG-IP system as a SAML IdP.....270
Setting up a BIG-IP system as a SAML service provider system.....273
Setting up connectivity from the IdP system to the SP systems 277

Chapter 33: BIG-IP System Federation for SP- and IdP-Initiated Connections.....281
Overview: Federating BIG-IP systems for SAML SSO (with an SSO portal).....282
About local IdP service.....282
About local SP service.....282
Task summary.....283
Flowchart: BIG-IP system federation configuration with SSO portal.....284
Setting up a BIG-IP system as a SAML IdP.....284
Setting up a BIG-IP system as a SAML service provider system.....287
Setting up connectivity from the IdP system to the SP systems 291

Legal Notices

Publication Date

This document was published on January 27, 2014.

Publication Number

MAN-0506-00

Copyright

Copyright © 2013-2014, F5 Networks, Inc. All rights reserved.

F5 Networks, Inc. (F5) believes the information it furnishes to be accurate and reliable. However, F5 assumes no responsibility for the use of this information, nor any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent, copyright, or other intellectual property right of F5 except as specifically described by applicable user licenses. F5 reserves the right to change specifications at any time without notice.

Trademarks

AAM, Access Policy Manager, Advanced Client Authentication, Advanced Firewall Manager, Advanced Routing, AFM, APM, Application Acceleration Manager, Application Security Manager, ARX, AskF5, ASM, BIG-IP, BIG-IQ, Cloud Extender, CloudFucious, Cloud Manager, Clustered Multiprocessing, CMP, COHESION, Data Manager, DevCentral, DevCentral [DESIGN], DNS Express, DSC, DSI, Edge Client, Edge Gateway, Edge Portal, ELEVATE, EM, Enterprise Manager, ENGAGE, F5, F5 [DESIGN], F5 Certified [DESIGN], F5 Networks, F5 SalesXchange [DESIGN], F5 Synthesis, f5 Synthesis, F5 Synthesis [DESIGN], F5 TechXchange [DESIGN], Fast Application Proxy, Fast Cache, FirePass, Global Traffic Manager, GTM, GUARDIAN, iApps, IBR, Intelligent Browser Referencing, Intelligent Compression, IPv6 Gateway, iControl, iHealth, iQuery, iRules, iRules OnDemand, iSession, L7 Rate Shaping, LC, Link Controller, Local Traffic Manager, LTM, LineRate, LineRate Systems [DESIGN], LROS, LTM, Message Security Manager, MSM, OneConnect, Packet Velocity, PEM, Policy Enforcement Manager, Protocol Security Manager, PSM, Real Traffic Policy Builder, SalesXchange, ScaleN, Signalling Delivery Controller, SDC, SSL Acceleration, software designed applications services, SDAC (except in Japan), StrongBox, SuperVIP, SYN Check, TCP Express, TDR, TechXchange, TMOS, TotALL, Traffic Management Operating System, Traffix Systems, Traffix Systems (DESIGN), Transparent Data Reduction, UNITY, VAULT, vCMP, VE F5 [DESIGN], Versafe, Versafe [DESIGN], VIPRION, Virtual Clustered Multiprocessing, WebSafe, and ZoneRunner, are trademarks or service marks of F5 Networks, Inc., in the U.S. and other countries, and may not be used without F5's express written consent.

All other product and company names herein may be trademarks of their respective owners.

Patents

This product may be protected by one or more patents indicated at:
http://www.f5.com/about/guidelines-policies/patents

Export Regulation Notice

This product may include cryptographic software. Under the Export Administration Act, the United States government may consider it a criminal offense to export this product from the United States.

RF Interference Warning

This is a Class A product. In a domestic environment this product may cause radio interference, in which case the user may be required to take adequate measures.

FCC Compliance

This equipment has been tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This unit generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user, at his own expense, will be required to take whatever measures may be required to correct the interference.

Any modifications to this device, unless expressly approved by the manufacturer, can void the user's authority to operate this equipment under part 15 of the FCC rules.

Canadian Regulatory Compliance

This Class A digital apparatus complies with Canadian ICES-003.

Standards Compliance

This product conforms to the IEC, European Union, ANSI/UL and Canadian CSA standards applicable to Information Technology products at the time of manufacture.

Acknowledgments

This product includes software developed by Bill Paul.

This product includes software developed by Jonathan Stone.

This product includes software developed by Manuel Bouyer.

This product includes software developed by Paul Richards.

This product includes software developed by the NetBSD Foundation, Inc. and its contributors.

This product includes software developed by the Politecnico di Torino, and its contributors.

This product includes software developed by the Swedish Institute of Computer Science and its contributors.

This product includes software developed by the University of California, Berkeley and its contributors.

This product includes software developed by the Computer Systems Engineering Group at the Lawrence Berkeley Laboratory.

This product includes software developed by Christopher G. Demetriou for the NetBSD Project.

This product includes software developed by Adam Glass.

This product includes software developed by Christian E. Hopps.

This product includes software developed by Dean Huxley.

This product includes software developed by John Kohl.

This product includes software developed by Paul Kranenburg.

This product includes software developed by Terrence R. Lambert.

This product includes software developed by Philip A. Nelson.

This product includes software developed by Herb Peyerl.

This product includes software developed by Jochen Pohl for the NetBSD Project.

This product includes software developed by Chris Provenzano.

This product includes software developed by Theo de Raadt.

This product includes software developed by David Muir Sharnoff.

This product includes software developed by SigmaSoft, Th. Lockert.

This product includes software developed for the NetBSD Project by Jason R. Thorpe.

This product includes software developed by Jason R. Thorpe for And Communications, http://www.and.com.

This product includes software developed for the NetBSD Project by Frank Van der Linden.

This product includes software developed for the NetBSD Project by John M. Vinopal.

This product includes software developed by Christos Zoulas.

This product includes software developed by the University of Vermont and State Agricultural College and Garrett A. Wollman.

This product includes software developed by Balazs Scheidler (bazsi@balabit.hu), which is protected under the GNU Public License.

This product includes software developed by Niels Mueller (nisse@lysator.liu.se), which is protected under the GNU Public License.

In the following statement, *This software* refers to the Mitsumi CD-ROM driver: This software was developed by Holger Veit and Brian Moore for use with 386BSD and similar operating systems. *Similar operating systems* includes mainly non-profit oriented systems for research and education, including but not restricted to NetBSD, FreeBSD, Mach (by CMU).

This product includes software developed by the Apache Group for use in the Apache HTTP server project (http://www.apache.org/).

This product includes software licensed from Richard H. Porter under the GNU Library General Public License (© 1998, Red Hat Software), www.gnu.org/copyleft/lgpl.html.

This product includes the standard version of Perl software licensed under the Perl Artistic License (© 1997, 1998 Tom Christiansen and Nathan Torkington). All rights reserved. You may find the most current standard version of Perl at http://www.perl.com.

This product includes software developed by Jared Minch.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

This product contains software based on oprofile, which is protected under the GNU Public License.

This product includes RRDtool software developed by Tobi Oetiker (http://www.rrdtool.com/index.html) and licensed under the GNU General Public License.

This product contains software licensed from Dr. Brian Gladman under the GNU General Public License (GPL).

This product includes software developed by the Apache Software Foundation (http://www.apache.org/).

This product includes Hypersonic SQL.

This product contains software developed by the Regents of the University of California, Sun Microsystems, Inc., Scriptics Corporation, and others.

This product includes software developed by the Internet Software Consortium.

This product includes software developed by Nominum, Inc. (http://www.nominum.com).

This product contains software developed by Broadcom Corporation, which is protected under the GNU Public License.

This product contains software developed by MaxMind LLC, and is protected under the GNU Lesser General Public License, as published by the Free Software Foundation.

This product includes Intel QuickAssist kernel module, library, and headers software licensed under the GNU General Public License (GPL).

This product includes software licensed from Gerald Combs (gerald@wireshark.org) under the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or any later version. Copyright ©1998 Gerald Combs.

This product includes software developed by Daniel Stenberg. Copyright ©1996 - 2012, Daniel Stenberg, (daniel@haxx.se). All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

This product includes software developed by Thomas Williams and Colin Kelley. Copyright ©1986 - 1993, 1998, 2004, 2007

Permission to use, copy, and distribute this software and its documentation for any purpose with or without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. Permission to modify the software is granted, but not the right to distribute the complete modified source code. Modifications are to be distributed as patches to the released version. Permission to distribute binaries produced by compiling modified sources is granted, provided you

1. distribute the corresponding source modifications from the released version in the form of a patch file along with the binaries,
2. add special version identification to distinguish your version in addition to the base release version number,
3. provide your name and address as the primary contact for the support of your modified version, and
4. retain our contact information in regard to use of the base software.

Permission to distribute the released version of the source code along with corresponding source modifications in the form of a patch file is granted with same provisions 2 through 4 for binary distributions. This software is provided "as is" without express or implied warranty to the extent permitted by applicable law.

This product contains software developed by Google, Inc. Copyright ©2011 Google, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This product includes software developed by Jeremy Ashkenas and DocumentCloud, and distributed under the MIT license. Copyright © 2010-2013 Jeremy Ashkenas, DocumentCloud.

This product includes gson software, distributed under the Apache License version 2.0. Copyright © 2008-2011 Google Inc.

This product includes Boost libraries, which are distributed under the Boost license (http://www.boost.org/LICENSE_1_0.txt).

Chapter 1

Authentication Concepts

- *About AAA server support*
- *About AAA high availability support*
- *About AAA and load balancing*
- *About AAA traffic and route domains*
- *About APM support for multiple authentication types*
- *About APM certificate authentication support*
- *About SSL certificates on the BIG-IP system*
- *About local user database support*
- *About guest access (one-time password) support*
- *About authentication for Microsoft Exchange clients*
- *Documentation for Access Policy Manager authentication*

About AAA server support

Access Policy Manager®(APM®) interacts with authentication, authorization, and accounting (AAA) servers that contain user information. APM supports these AAA servers: RADIUS (authentication and accounting), Active Directory (authentication and query), LDAP (authentication and query), CRLDP, OCSP Responder, TACACS+ (authentication and accounting), SecurID, Kerberos, and HTTP.

A typical configuration includes:

- An APM AAA server configuration object that specifies information about the external AAA server.
- An access policy that includes a logon item to obtain credentials and an authentication item that uses the credentials to authenticate against a specific AAA server.

About AAA high availability support

Using AAA high availability with Access Policy Manager® (APM®), you can configure multiple authentication servers to process requests, so that if one authentication server goes down or loses connectivity, the others can resume authentication requests, and new sessions can be established. APM supports these AAA servers for high availability: RADIUS, Active Directory, LDAP, CRLDP, and TACACS+.

A typical configuration includes:

- An APM AAA server configuration object that specifies a pool of external AAA servers.
- An access policy that includes a logon item to obtain credentials and an authentication item that uses the credentials to authenticate against one of the servers in the pool.

About AAA and load balancing

When an AAA server supports high availability, you can configure a pool for it in the AAA configuration itself. An AAA server does not load balance over a pool that is attached to a virtual server.

About AAA traffic and route domains

To use route domains for AAA authentication traffic, you must use the pool option in the AAA server configuration. When **Use Pool** is the selected **Server Connection** option, the server address field can take an IP address with route domain (`IPAddress%`*`RouteDomain`*) format. The route domain value is ignored when the AAA server is configured to connect directly to a single server.

About APM support for multiple authentication types

You can add multiple authentication types to an access policy. For example, a user who fails Active Directory authentication might then attempt RADIUS authentication. Or, you might require authentication using a client certificate and then an AAA server.

You can add an authentication item anywhere in the access policy. Typically, you place authentication items somewhere after a logon item.

About APM certificate authentication support

SSL handshake verification and certificate revocation status

Access Policy Manager® (APM®) supports verifying the SSL handshake that occurs at the start of a session or renegotiating the SSL handshake and checking it on demand. A typical configuration includes:

- An access policy that includes a certificate-related access policy item, either Client Cert Inspection or On-Demand Cert Auth.
- A client SSL profile configured per the requirements of Client Cert Inspection or On-Demand Cert Auth.

Note: *If the client SSL profile specifies a certificate revocation list, the access policy item verifies against it.*

Certificate revocation status with OCSP or CRLDP

APM also supports verifying client certificate revocation status with an Online Certificate Status Protocol (OCSP) AAA server or with a Certificate Revocation List Distribution Point (CRLDP) AAA server.

A typical configuration includes:

- An AAA server configured to point to an external server (OCSP Responder or CRLDP).
- An access policy that includes either a Client Cert Inspection or an On-Demand Cert Auth access policy item and the appropriate authentication item (OCSP Auth or CRLDP Auth).
- A client SSL profile configured per the requirements of Client Cert Inspection or an On-Demand Cert Auth.

About SSL certificates on the BIG-IP system

Before systems on a network can authenticate one another using SSL, you must install one or more SSL certificates on the BIG-IP® system. An *SSL certificate* is a certificate that a BIG-IP system device presents to another device on the network, for authentication purposes. An SSL certificate can be either a self-signed certificate or a trusted CA certificate.

When you install BIG-IP® software, the application includes a self-signed SSL certificate named `Default`. A *self-signed certificate* is an authentication mechanism that is created and authenticated by the system on which it resides.

If your network includes one or more certificate authority (CA) servers, you can replace the self-signed certificate on each BIG-IP system with a *trusted CA certificate*, that is, a certificate that is signed by a third

party. Authenticating BIG-IP systems using trusted CA certificates is more secure than using self-signed certificates.

To ease the task of creating certificate requests and sending them to certificate authorities for signature, the BIG-IP system provides a set of certificate management screens within the BIG-IP Configuration utility.

About local user database support

Access Policy Manager® (APM®) supports authentication against a database that you create on the BIG-IP® system using the Configuration utility. You can employ a local user database for on-box authentication or to control access to external AAA servers.

A typical configuration includes:

- A local user database that you create and populate using the Configuration utility.
- An access policy that includes a local user database authentication item.

About guest access (one-time password) support

Access Policy Manager® (APM®) supports guest access with one-time password generation and verification. A typical configuration includes:

- An SMTP server for sending email or an HTTP AAA server for sending a text message.
- An access policy that includes items to generate a one-time password (OTP), send the generated password to a user, enable the user to log on, and verify the OTP that the user enters.

About authentication for Microsoft Exchange clients

Access Policy Manager® (APM®) supports NTLM and HTTP basic authentication for Microsoft Exchange clients and for this support requires an Exchange profile, created in the Configuration utility. Configuration requirements for NTLM and HTTP basic authentication for Microsoft Exchange clients are otherwise distinct.

Documentation for Access Policy Manager authentication

You can access all of the following APM® documentation from the AskF5™ Knowledge Base located at `http://support.f5.com/`.

Document	Description
BIG-IP® Access Policy Manager® Authentication and Single Sign-On Guide (this guide)	Use this guide to configure APM for authentication, using: • AAA servers • SSL certificates • Local user database • One-time password (guest authentication)

Document	Description
	• SSO configurations • Secure Assertion Markup Language (SAML) and to configure APM to authenticate Microsoft Exchange clients.
BIG-IP® Access Policy Manager®: Third-Party Integration Implementations	Use this document to configure APM for native integration with Oracle Access Manager.

Chapter 2

Active Directory Authentication

- *About Active Directory authentication*
- *About Active Directory password management*
- *About AAA high availability*
- *About how APM handles binary values in Active Directory attributes*
- *Task summary for Active Directory authentication*
- *Testing AAA high availability for supported authentication servers*
- *Example access policy using Active Directory authentication and query*
- *Active Directory authentication session variables*
- *Active Directory cross-domain support rules*
- *Active Directory authentication and query troubleshooting tips*
- *Overview: Using Active Directory Trusted Domains*

About Active Directory authentication

You can authenticate using Active Directory authentication with Access Policy Manager. We support using Kerberos-based authentication through Active Directory.

About Active Directory password management

Access Policy Manager® (APM®) supports password management for Active Directory authentication.

How APM supports password reset

The process works in this sequence:

- Access Policy Manager uses the client's user name and password to authenticate against the Active Directory server on behalf of the client.
- If the user password on the Active Directory server has expired, Access Policy Manager returns a new logon screen back to the user, requesting that the user change the password.
- After the user submits the new password, Access Policy Manager attempts to change the password on the Active Directory server. If this is successful, the user's authentication is validated.

If the password change fails, it is likely that the Active Directory server rejected it because the password did not meet the minimum requirements such as password length.

Number of attempts APM provides for password reset

In the AD Auth action, APM provides a **Max Password Reset Attempts Allowed** property.

Change password option

In the Logon page action, APM provides a Checkbox property in the visual policy editor. You can add the option on the APM logon screen to change the log on password.

About AAA high availability

Using AAA high availability with Access Policy Manager® (APM®), you can configure multiple authentication servers to process requests, so that if one authentication server goes down or loses connectivity, the others can resume authentication requests, and new sessions can be established, as usual.

Note: *Although new authentications fail if the BIG-IP® system loses connectivity to the server, existing sessions are unaffected provided that they do not attempt to re-authenticate.*

APM supports the following AAA servers for high availability: RADIUS, Active Directory, LDAP, CRLDP, and TACACS+. APM supports high availability by providing the option to create a pool of server connections when you configure the supported type of AAA server.

Note: *If you use AAA with pools, such as RADIUS pools or Active Directory pools, APM assigns each pool member with a different number for the pool member's priority group value. Since APM does not support*

AAA load balancing, APM must define each pool member with a different priority group. The priority group number increases automatically with each created pool member.

About how APM handles binary values in Active Directory attributes

For Active Directory, Access Policy Manager® (APM®) converts an attribute value to hex only if the value contains unprintable characters. If the session variable contains several values, and one or more of those values is unprintable, then APM converts only those particular values to hex.

Case 1:

Handling of attributes with single value:

```
7ecc84a2.session.ad.last.attr.objectSid 58 /
 0x0105000000000005150000001 3fe8e97c03cd5b5ad04e2e255040000
```

Case 2:

Handling of attributes with multiple values (mix of binary and non-binary values):

```
7ecc84a2.session.ad.last.attr.memberOf 460 |
CN=printable
group,OU=groups,OU=someco,DC=sherwood,DC=labt,DC=fp,DC=somelabnet,DC=com |
0x434e3d756e7072696e7461626c6520c2bdc2a12067726f75702c4f553d67726f7570732c4f553d66352
 | /
c44433d73686572776f6f642c44433d6c6162742c44433d66702c44433d66356e65742c44433d636f6d
 | /
CN=Domain Users,CN=Users,DC=smith,DC=labt,DC=fp,DC=somlabnet,DC=com | /
CN=CERTSVC_DCOM_ACCESS,CN=Users,DC=smith,DC=labt,DC=fp,DC=somelabnet,DC=com
| /
CN=Users,CN=Builtin,DC=smith,DC=labt,DC=fp,DC=somelabnet,DC=com |
```

Task summary for Active Directory authentication

This task list includes all steps required to set up this configuration. If you are adding Active Directory authentication to an existing access policy, you do not need to create another access profile, and the access policy might already include a logon page.

Task list

Configuring an Active Directory AAA server
Creating an access profile
Configuring Active Directory authentication
Creating a virtual server

Configuring an Active Directory AAA server

You configure an Active Directory AAA server in Access Policy Manager® (APM) to specify domain controllers and credentials for APM® to use for authenticating users.

1. On the Main tab, click **Access Policy** > **AAA Servers** > **Active Directory**.
   The Active Directory Servers list screen opens.
2. Click **Create**.
   The New Server properties screen opens.
3. In the **Name** field, type a unique name for the authentication server.
4. In the **Domain Name** field, type the name of the Windows domain.
5. For the **Server Connection** setting, select one of these options:
   - Select **Use Pool** to set up high availability for the AAA server.
   - Select **Direct** to set up the AAA server for standalone functionality.
6. If you selected **Direct**, type a name in the **Domain Controller** field.
7. If you selected **Use Pool**, configure the pool:
   a) Type a name in the **Domain Controller Pool Name** field.
   b) Specify the **Domain Controllers** in the pool by typing the IP address and host name for each, and clicking the **Add** button.
   c) To monitor the health of the AAA server, you have the option of selecting a health monitor: only the **gateway_icmp** monitor is appropriate in this case; you can select it from the **Server Pool Monitor** list.
8. In the **Admin Name** field, type a is case-sensitive name for an administrator who has Active Directory administrative permissions.
   APM uses the information in the **Admin Name** and **Admin Password** fields for AD Query. If Active Directory is configured for anonymous queries, you do not need to provide an Admin Name. Otherwise, APM needs an account with sufficient privilege to bind to an Active Directory server, fetch user group information, and fetch Active Directory password policies to support password-related functionality. (APM must fetch password policies, for example, if you select the Prompt user to change password before expiration option in an AD Query action.) If you do not provide Admin account information in this configuration, APM uses the user account to fetch information. This works if the user account has sufficient privilege.
9. In the **Admin Password** field, type the administrator password associated with the Domain Name.
10. In the **Verify Admin Password** field, retype the administrator password associated with the **Domain Name** setting.
11. In the **Group Cache Lifetime** field, type the number of days.
    The default lifetime is 30 days.
12. In the **Password Security Object Cache Lifetime** field, type the number of days.
    The default lifetime is 30 days.
13. From the **Kerberos Preauthentication Encryption Type** list, select an encryption type.
    The default is **None**. If you specify an encryption type, the BIG-IP® system includes Kerberos preauthentication data within the first authentication service request (AS-REQ) packet.
14. In the **Timeout** field, type a timeout interval (in seconds) for the AAA server. (This setting is optional.)
15. Click **Finished**.
    The new server displays on the list.

This adds the new Active Directory server to the Active Directory Servers list.

Creating an access profile

You create an access profile to provide the access policy configuration for a virtual server that establishes a secured session.

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. Click **Create**.
   The New Profile screen opens.
3. Type a name for the access profile.
4. From the **Profile Type** list, select one:
   - **APM-LTM** - Select for a web access management configuration.
   - **SSO** - Select only when you do not need to configure an access policy.
   - **SWG - Explicit** - Select to configure access using Secure Web Gateway explicit forward proxy.
   - **SWG - Transparent** - Select to configure access using Secure Web Gateway transparent forward proxy.
   - **SSL-VPN** - Select for other types of access, such as network access, portal access, application access. (Most access policy items are available for this type.)
   - **ALL** - Select for any type of access.

   Additional settings display.
5. In the Language Settings area, add and remove accepted languages, and set the default language.
   A browser uses the highest priority accepted language. If no browser language matches the accepted languages list, the browser uses the default language.
6. Click **Finished**.

This creates an access profile with a default access policy.

Configuring Active Directory authentication

Before you configure an access policy use Active Directory authentication, , you must have at least one Active Directory AAA server configured.

You create an access policy like this one to obtain user credentials and use them to authenticate the user against an external Active Directory server before granting access.

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. In the Access Policy column, click the **Edit** link for the access profile you want to configure.
   The visual policy editor opens the access policy in a separate screen.
3. Click the **(+)** icon anywhere in the access policy to add a new action item.
   A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
4. On the Logon tab, select **Logon Page** and click the **Add Item** button.
   The Logon Page Agent properties screen opens.
5. Make any changes that you require to the logon page properties and click **Save**.
   The properties screen closes and the visual policy editor displays.
6. Click the **(+)** icon anywhere in the access policy to add a new action item.

A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.

7. On the Authentication tab, select **AD Auth** and click **Add Item**.
   A Properties popup screen opens.
8. From the **Server** list, select the AAA Active Directory server to use for authentication, and click **Save**.
9. You can also set these options.

Option	Description
Cross Domain Support	Specifies whether AD cross domain authentication support is enabled for AD Auth agent.
Complexity check for Password Reset	Specifies whether Access Policy Manager performs a password policy check. ***Note:*** *Enabling this option increases overall authentication traffic significantly because Access Policy Manager must retrieve additional information. Because this option might require administrative privileges, if you enable it you should specify the administrator name and password on the AAA Active Directory server configuration page.*
Show Extended Error	When enabled, displays the comprehensive error messages generated by the authentication server to show on the user's Logon page. This setting is intended for use in testing only in a production or debugging environment. If you enable this setting in a live environment, your system might be vulnerable to malicious attacks
Max Logon Attempts Allowed	Specifies the number of user authentication logon attempts to allow. ***Note:*** *To use this access policy for Citrix Receiver client access, set the value to 1.*
Max Password Reset Attempts Allowed	Specifies the number of times that Access Policy Manager allows the user to try to change password.

10. Click **Apply Access Policy** to save your configuration.

This adds a logon page and Active Directory authentication to the access policy.

To put an access policy into effect, add it to a virtual server.

Creating a virtual server

When creating a virtual server for an access policy, specify that the virtual server is a host virtual server, and not a network virtual server.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, select **Host** and in the **Address** field, type the IP address for the virtual server.
5. In the **Service Port** field, type a port number or select a service name from the **Service Port** list.

6. From the **HTTP Profile** list, select **http**.
7. If you use server SSL for this connection, from the **SSL Profile (Server)** list, select a server SSL profile.
8. If you use client SSL for this profile, from the **SSL Profile (Client)** list, select a client SSL profile.
9. In the Access Policy area, from the **Access Profile** list, select the access profile.
10. From the **Connectivity Profile** list, select a connectivity profile.

    You can select the default connectivity profile, **connectivity** if you have not defined a specific profile for the traffic that is directed to this virtual server.
11. Click **Finished**.

You have configured a host virtual server and associated an access profile with it.

Testing AAA high availability for supported authentication servers

To effectively test that high availability works for your authentication servers, you should have two servers that are accessible, where you can remove one of them from the network.

Note: *High availability is supported for these authentication server types only: RADIUS, Active Directory, LDAP, CRLDP, and TACACS+.*

If you configured a supported authentication server type to use a pool of connection servers, you can test the configuration using these steps.

1. Begin a `tcpdump` on the Access Policy Manager®, using a protocol analyzer, and scanning for packets destined for the specific port for your authentication server.
2. Log in to the virtual server with both servers active.
3. Using the `tcpdump` records, verify that the requests are being sent to the higher priority server.
4. Log out of the virtual server.
5. Disable the higher-priority server.
6. Log in to the virtual server again.
7. Verify that the request is being sent to the other server.
8. Log out again, re-enabling the server, and try one more time to verify that the new requests are being sent to the high priority server.

Example access policy using Active Directory authentication and query

This is an example of an access policy with all the associated elements that are needed to authenticate and authorize your users with Active Directory authentication and Active Directory query.

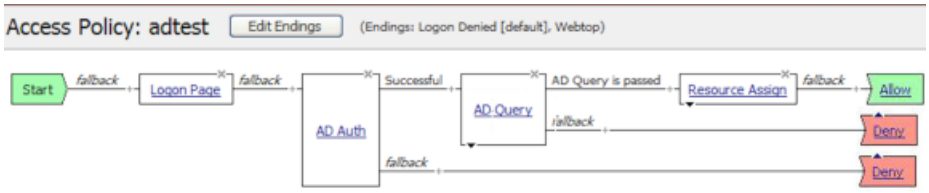


Figure 1: Example of an access policy for AD auth and query

Active Directory authentication session variables

When the AD Auth access policy item runs, it populates session variables which are then available for use in access policy rules. The tables list the session variables for the Active Directory access policy items and for a logon access policy item.

Session variables for Active Directory authentication

Session Variable	Description
`session.ad.last.actualdomain`	AD Auth agent sets this variable to the actual user domain used for successful Active Directory authentication, whether cross-domain support is enabled or disabled.
`session.ad.last.authresult`	Provides the result of the Active Directory authentication. The available values are: • 0: Failed • 1: Passed
`session.ad.last.errmsg`	Displays the error message for the last login. If `session.ad.last.authresult`is set to 0, then `session.ad.last.errmsg` might be useful for troubleshooting purposes.

Common session variables

Session Variable	Description
`session.logon.last.username`	Provides user credentials. The `username` string is stored after encrypting, using the system's client key.
`session.logon.last.password`	Provides user credentials. The `password` string is stored after encrypting, using the system's client key.

Active Directory cross-domain support rules

Rules	Explanation
Cross-domain support and **split domain from username** are both enabled.	If you enable **cross domain support**, and enable **split domain username** at the login page, and then the user enters his user name, such as `user@domain.com`, Access Policy Manager® uses the `user@domain.com` as the user principal name to authenticate the user against USERNAME.COM domain.
Cross-domain support is enabled but **split domain from username** is disabled	Access Policy Manager handles the user's input as a simple user name and escape "@" and "\" chars. In other words, Access Policy Manager uses `user\@userdomain.com@DEFAULTREALM.COM` to authenticate the user, where DEFAULTREALM.COM is the domain name that was configured on the AAA AD Server configuration page.

Rules	Explanation
If user does not specify a user's domain	Regardless of whether **split domain from username** option is enabled or disabled, Access Policy Manager uses `user@defaultrealm.com` to authenticate the user.

Active Directory authentication and query troubleshooting tips

You might run into problems with Active Directory authentication and query processes in some instances. Follow these tips to try to resolve any issues you might encounter.

Active Directory auth authentication and query troubleshooting

Possible error messages	Possible explanations and corrective actions
`Domain controller reply did not match expectations. (-1765328237)`	This error occurs when the principal/domain name does not match the domain controller server's database. For example, if the actual domain is `SALES.MYCOMPANY.COM`, and the administrator specifies `STRESS` as the domain, then the `krb5.conf` file displays the following: `default_realm = SALES SALES = { domain controller = (domain controller server) admin = (admin server)` So, when the administrator tries to authenticate with `useraccount@SALES`, the krb5 library notices that the principal name `SALES` differs from the actual one in the server database.

Additional troubleshooting tips for Active Directory authentication

You should	Steps to take
Check that your access policy is attempting to perform authentication	• Refer to the message boxes in your access policy to display information on what the access policy is attempting to do. • Refer to `/var/log/apm` to view authentication attempts by the access policy. ***Note:*** *Make sure that your log level is set to the appropriate level. The default log level is* `notice`*.*
Confirm network connectivity	• Access the Access Policy Manager® through the command line interface and check your connectivity by pinging the Active Directory server using the host entry in the AAA Server box. • Confirm that the Active Directory port (`88` or `389`) is not blocked between the Access Policy Manager, and the Active Directory server.
Check the Active Directory server configuration	• Confirm that the Active Directory server name can be resolved to the correct IP address, and that the reverse name resolution (IP address to name) is also possible. • Confirm that the Active Directory server and the Access Policy Manager have the correct time setting configured. ***Note:*** *Since Active Directory is sensitive to time settings, use NTP to set the correct time on the Access Policy Manager.*

You should	Steps to take
Capture a TCP dump	• Take a TCP dump from the Access Policy Manager when authentication attempts are made. For example, `%tcpdump-i 1.1 -s /tmp/dump`. You must first determine what interface the self IP address is on. These TCP dumps indicate activities between the Access Policy Manager and the authentication server. • Run the authentication test. After authentication fails, stop the TCP dump, and download the TCP dump to a client system, and use an analyzer to troubleshoot. ***Important:*** *If you decide to escalate the issue to customer support, you must provide a capture of the TCP dump when you encounter authentication issues that you cannot otherwise resolve on your own.*

Overview: Using Active Directory Trusted Domains

Active Directory Trusted Domains option in BIG-IP® Access Policy Manager® (APM) manages Active Directory AAA trusted domains. For enterprises that are service providers, their customers might have their own enterprise network infrastructure. Using APM®, the service provider provides access to their customers' networks. To avoid network traffic collisions between two customer networks, the service provider separates each customer using route domains. A *route domain* is a configuration object that isolates network traffic for a particular application on the network. The service provider uses Active Directory to authenticate their customer users. However, each customer's Active Directory service can contain multiple trusted domains or forests. The service provider can use the Active Directory Trusted Domains option to authenticate users across all trusted domains or forests for a customer.

Configuring an Active Directory Trusted Domain

You must create at least one Active Directory AAA server before you can configure an Active Directory Trusted Domain.

Configure an Active Directory Trusted Domain in Access Policy Manager ®(APM) to authenticate users in route domains with at least one trusted domain.

1. On the Main tab, click **Access Policy** > **AAA Servers** > **Active Directory Trusted Domains**.
   The Active Directory Trusted Domains list screen opens.
2. Click **Create**.
   The Create New Active Directory Trusted Domains screen opens.
3. In the **Name** field, type a name for the Active Directory Trusted Domain.
4. In the **Description** field, type a description for the Active Directory Trusted Domain.
5. For the **XXX** setting, in the **Available** list, select the Active Directory AAA server that you want to add to the Trusted Domain, and click << to move the Active Directory AAA server into the **Selected** list.
6. From the **Root** list, select a root domain.

   You use the root domain for an initial authentication request, such as an entry point to an Active Directory forest.
7. Click **OK**.

You have now added an Active Directory Trusted Domain to the Active Directory Trusted Domain list.

You can now add the Active Directory Trusted Domain option to either the AD Auth agent or the AD Query agent in the visual policy editor.

Note: *You can select a trusted domain only if you enable the Cross Domain support option.*

Chapter 3

Active Directory Query

- *About Active Directory queries*
- *About nested groups in Active Directory and LDAP queries*
- *About Active Directory password management*
- *About how APM handles binary values in Active Directory attributes*
- *Adding an Active Directory query to an access policy*
- *Using AD query with IPv6*
- *Active Directory query session variables*
- *Active Directory authentication and query troubleshooting tips*

About Active Directory queries

When running the AD Query access policy item, Access Policy Manager® (APM®) queries an external Active Directory server for additional information about the user. The AD Query item looks up the attribute memberOf to fetch the groups to which a user belongs and provides an additional option to fetch the primary group.

The AD Query item does not authenticate user credentials. To authenticate users, use another or an additional authentication item in the access policy.

About nested groups in Active Directory and LDAP queries

A *nested group* is a group that is a member of another group. For example, group1 is a member of group3 and group4. A user, user1, that belongs to group1 and group2 also belongs to group3 and group4 through nesting.

Whether AD Query and LDAP Query return nested groups in session variables

The AD Query and LDAP Query access policy items return and store the groups to which a user belongs in the `memberOf` session variable.

The contents of the `memberOf` session variable differ depending on whether the **Fetch Nested Group** setting is enabled or disabled in AD Query or LDAP Query properties:

- Enabled - The `memberOf` session variable contains all groups to which the user belongs. As in the example, this includes group1, group2, group3, and group4.
- Disabled - The `memberOf` session variable contains groups to which the user belongs directly. Based on the example, this would be group1 and group2.

About Active Directory password management

Access Policy Manager® (APM®) supports password management for Active Directory authentication.

How APM supports password reset

The process works in this sequence:

- Access Policy Manager uses the client's user name and password to authenticate against the Active Directory server on behalf of the client.
- If the user password on the Active Directory server has expired, Access Policy Manager returns a new logon screen back to the user, requesting that the user change the password.
- After the user submits the new password, Access Policy Manager attempts to change the password on the Active Directory server. If this is successful, the user's authentication is validated.

If the password change fails, it is likely that the Active Directory server rejected it because the password did not meet the minimum requirements such as password length.

Number of attempts APM provides for password reset

In the AD Auth action, APM provides a **Max Password Reset Attempts Allowed** property.

Change password option

In the Logon page action, APM provides a Checkbox property in the visual policy editor. You can add the option on the APM logon screen to change the log on password.

About how APM handles binary values in Active Directory attributes

For Active Directory, Access Policy Manager® (APM®) converts an attribute value to hex only if the value contains unprintable characters. If the session variable contains several values, and one or more of those values is unprintable, then APM converts only those particular values to hex.

Case 1:

Handling of attributes with single value:

```
7ecc84a2.session.ad.last.attr.objectSid 58 /
 0x0105000000000005150000000013fe8e97c03cd5b5ad04e2e255040000
```

Case 2:

Handling of attributes with multiple values (mix of binary and non-binary values):

```
7ecc84a2.session.ad.last.attr.memberOf 460 |
CN=printable
group,OU=groups,OU=someco,DC=sherwood,DC=labt,DC=fp,DC=somelabnet,DC=com |
0x434e3d756e7072696e7461626c6520c2bdc2a12067726f75702c4f553d67726f7570732c4f553d66352
 | /
c44433d73686572776f6f642c44433d6c6162742c44433d66702c44433d66356e65742c44433d636f6d
 | /
CN=Domain Users,CN=Users,DC=smith,DC=labt,DC=fp,DC=somlabnet,DC=com | /
CN=CERTSVC_DCOM_ACCESS,CN=Users,DC=smith,DC=labt,DC=fp,DC=somelabnet,DC=com
| /
CN=Users,CN=Builtin,DC=smith,DC=labt,DC=fp,DC=somelabnet,DC=com |
```

Adding an Active Directory query to an access policy

Before you add an AD query to an access policy, you must have at least one AD AAA server configured. You should also have an access profile that is configured with actions to authenticate the user.

You add an AD query to an access policy to get information about a user; for example, you might want to know whether a user is a member of a group before grancing access to particular particular resources. APM stores the attributes it retrieves in session variables.

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. In the Access Policy column, click the **Edit** link for the access profile you want to configure.
   The visual policy editor opens the access policy in a separate screen.

3. Click the **(+)** icon anywhere in the access policy to add a new action item.
   A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
4. On the Authentication tab, select **AD Query** and click **Add Item**.
   A Properties popup screen opens.
5. From the **Server** list, select the Active Directory AAA server to query.
6. You can also set these options.

Option	Description
SearchFilter	Type a search filter. (Otherwise if left empty, the policy uses the default filter, sAMAccountName=%{*session.logon.last.username*}. As a result, the `SearchFilter` parameter is populated with the Subject Alternative Name from the current Active Directory session.)
Fetch Primary Group	Enable this setting to populate the user's primary group in the session variables. This setting is optional.
Cross Domain Support	Specifies whether AD cross domain authentication support is enabled for AD Auth agent. This setting is optional.
Fetch Nested Groups	Enable to populate the *memberOf* session variable with user's membership in nested groups in addition to the groups to which the user belongs directly. ***Important:*** *Access Policy Manager does not query for the primary group and add it to the* `memberOf` *attribute. You must manually look up the attribute* `memberOf` *as well as the primary group.*
Complexity Check for Passwork Reset	Enable this setting so that APM performs the password policy checks it supports.
Max Password Reset Attempts Allowed	Select the number of times to allow a user to try to reset their password.
Prompt user to change password before expiration	Set (N days) to prompt user to change the password before it expires. The default is **none** (disabled). This setting is optional.

7. Click **Save**.
8. Click **Apply Access Policy** to save your configuration.

This adds an Active Directory query to the access policy.

Using AD query with IPv6

When you configure an AD AAA server with an IPv6 address in the Domain Controller setting, an AD query does not work. However, we tested AD query with an IPv6 address using this approach.

1. In the AD server configuration, use the host name of the DC in the Domain Controller setting.

```
apm aaa active-directory /Common/AD-IPv6 {
admin-encrypted-password ".(.5(lEhJfN\\<^FaLGC0Bt8CG0KMfR\\9;coEKdIm=5@32II"

admin-name Administrator
```

```
domain enterprise.lab.fp.mynet.com
domain-controller win2008.enterprise.lab.fp.mynet.com
```

The host name is win2008.enterprise.lab.fp.mynet.com in the example.

2. Update the system's global setting to include a remote host entry for the DC host name that was used in step 1 and map it to an IPv4 address as shown in this example.

```
sys global-settings {
gui-setup disabled
hostname bigip2mgmt.lab.fp.mynet.com
mgmt-dhcp disabled
remote-host {
/Common/abc { addr 172.31.54.99
hostname win2008.enterprise.lab.fp.mynet.com
}
}
}
```

3. Create a pool with the DC IPv6 address as a member as shown in this example.

```
ltm pool /Common/AD-IPv6-Pool {
members {
/Common/fd00:ffff:ffff:fff1:912e:cdfe:c884:2607.any {
address fd00:ffff:ffff:fff1:912e:cdfe:c884:2607
}
}
}
```

4. Create a wildcard TCP virtual server with these settings:
   a) Set the **Destination IP** settings to the IPv4 address that was used in step 2.
      That address is 172.31.54.99 in the example.
   b) For the **Service Port** setting, select *** All ports**.
   c) In the Configuration area, leave the **Protocol** setting at the default, **TCP**.
   d) Scroll down to the **Source Address Translation** setting and select **Auto Map**.
   e) Scroll down to the Resources area and select the pool that you configured previously from the **Default Pool** list.

```
ltm virtual /Common/bigip2.lab.fp.mynet.com-tcp {
destination /Common/172.31.54.99:any
ip-protocol tcp
mask 255.255.255.255
pool /Common/AD-IPv6-Pool
profiles {
/Common/tcp { }
}
source-address-translation automap
translate-port disabled
vlans-disabled
}
```

5. Create another similar virtual server, but for UDP traffic. (Set the Protocol setting in the virtual server configuration to UDP).

```
ltm virtual /Common/bigip2.lab.fp.mynet.com-udp {
destination /Common/172.31.54.99:any
```

```
ip-protocol udp
mask 255.255.255.255
pool /Common/AD-IPv6-Pool
profiles {
/Common/udp { }
}
source-address-translation automap
translate-port disabled
vlans-disabled
}
```

Active Directory query session variables

When the AD Query access policy item runs, it populates session variables which are then available for use in access policy rules. The tables list the session variables for the Active Directory access policy items and for a logon access policy item.

Session variables for Active Directory query

Session Variable	Description
`session.ad.last.queryresult`	Provides the result of the Active Directory query. The available values are: • 0: Failed • 1: Passed
`session.ad.last.errmsg`	Displays the error message for the last login. If `session.ad.last.queryresult` is set to 0, then `session.ad.last.errmsg` might be useful for troubleshooting purposes.
`session.ad.last.attr.$attr_name`	*`$attr_name`* is a value that represents the user's attributes received from the Active Directory. Each attribute is converted to separate session variables.
`session.ad.last.attr.primarygroup.$attr_name`	*`primarygroup.$attr_name`* is a value that represents the user's group attributes received from the Active Directory. Each attribute is converted to separate session variables.

Common session variables

Session Variable	Description
`session.logon.last.username`	Provides user credentials. The `username` string is stored after encrypting, using the system's client key.
`session.logon.last.password`	Provides user credentials. The `password` string is stored after encrypting, using the system's client key.

Active Directory authentication and query troubleshooting tips

You might run into problems with Active Directory authentication and query processes in some instances. Follow these tips to try to resolve any issues you might encounter.

Active Directory auth authentication and query troubleshooting

Possible error messages	Possible explanations and corrective actions
`Domain controller reply did not match expectations.(-1765328237)`	This error occurs when the principal/domain name does not match the domain controller server's database. For example, if the actual domain is `SALES.MYCOMPANY.COM`, and the administrator specifies `STRESS` as the domain, then the `krb5.conf` file displays the following: `default_realm = SALES SALES = { domain controller = (domain controller server) admin = (admin server)` So, when the administrator tries to authenticate with `useraccount@SALES`, the krb5 library notices that the principal name `SALES` differs from the actual one in the server database.

Additional troubleshooting tips for Active Directory authentication

You should	Steps to take
Check that your access policy is attempting to perform authentication	• Refer to the message boxes in your access policy to display information on what the access policy is attempting to do. • Refer to `/var/log/apm` to view authentication attempts by the access policy. ***Note:*** *Make sure that your log level is set to the appropriate level. The default log level is* `notice`.
Confirm network connectivity	• Access the Access Policy Manager® through the command line interface and check your connectivity by pinging the Active Directory server using the host entry in the AAA Server box. • Confirm that the Active Directory port (`88` or `389`) is not blocked between the Access Policy Manager, and the Active Directory server.
Check the Active Directory server configuration	• Confirm that the Active Directory server name can be resolved to the correct IP address, and that the reverse name resolution (IP address to name) is also possible. • Confirm that the Active Directory server and the Access Policy Manager have the correct time setting configured. ***Note:*** *Since Active Directory is sensitive to time settings, use NTP to set the correct time on the Access Policy Manager.*
Capture a TCP dump	• Take a TCP dump from the Access Policy Manager when authentication attempts are made. For example, `%tcpdump-i 1.1 -s /tmp/dump`. You must first determine what interface the self IP address is on. These TCP dumps indicate activities between the Access Policy Manager and the authentication server.

You should	Steps to take
	• Run the authentication test. After authentication fails, stop the TCP dump, and download the TCP dump to a client system, and use an analyzer to troubleshoot. ***Important:*** *If you decide to escalate the issue to customer support, you must provide a capture of the TCP dump when you encounter authentication issues that you cannot otherwise resolve on your own.*

Chapter 4

LDAP and LDAPS Authentication

- *About LDAP and LDAPS authentication*
- *About how APM handles binary values in LDAP attributes*
- *About AAA high availability*
- *Task summary for configuring for LDAPS authentication*
- *Testing AAA high availability for supported authentication servers*
- *Example of LDAP auth and query default rules*
- *LDAP authentication session variables*
- *UserDN settings in LDAP*
- *LDAP authentication and query troubleshooting tips*

About LDAP and LDAPS authentication

You can use LDAPS in place of LDAP when the authentication messages between the Access Policy Manager® and the LDAP server must be secured with encryption. However, there are instances where you will not need LDAPS and the security it provides. For example, authentication traffic happens on the internal side of Access Policy Manager, and might not be subject to observation by unauthorized users. Another example of when not to use LDAPS is when authentication is used on separate VLANs to ensure that the traffic cannot be observed by unauthorized users.

Figure 2: How LDAP works

LDAPS is achieved by directing LDAP traffic over a virtual server that uses server side SSL to communicate with the LDAP server. Essentially, the system creates an LDAP AAA object that has the address of the virtual server. That virtual server (with server SSL) directs its traffic to a pool, which has as a member that has the address of the LDAP server.

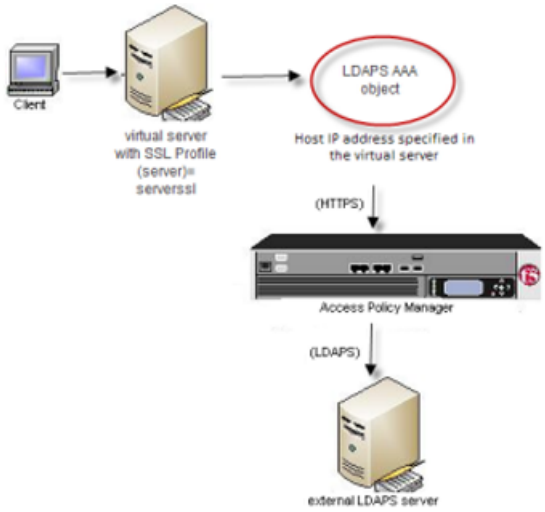


Figure 3: How LDAPS works

About how APM handles binary values in LDAP attributes

For LDAP, Access Policy Manager® (APM®) converts an attribute value to hex only if the value contains unprintable characters. If the session variable contains several values, and one or more of those values is unprintable, then APM converts only those particular values to hex.

Case 1:
Handling of attributes with single value:

```
9302eb80.session.ldap.last.attr.objectGUID 34 /
0xfef232d3039be9409a72bfc60bf2a6d0
```

Case 2:
Handling of attributes with multiple values (mix of binary and non-binary values):

```
29302eb80.session.ldap.last.attr.memberOf 251 | /
CN=printable group,OU=groups,OU=someco,DC=smith, /
DC=labt,DC=fp,DC=somelabnet,DC=com | /
0x434e3d756e7072696e7461626c6520c2bdc2a12067726f75702c4f553d67726f7570732c4f553d66352c
 /
44433d73686572776f6f642c44433d6c6162742c44433d66702c44433d66356e65742c44433d636f6d
 |
```

About AAA high availability

Using AAA high availability with Access Policy Manager® (APM®), you can configure multiple authentication servers to process requests, so that if one authentication server goes down or loses connectivity, the others can resume authentication requests, and new sessions can be established, as usual.

Note: *Although new authentications fail if the BIG-IP® system loses connectivity to the server, existing sessions are unaffected provided that they do not attempt to re-authenticate.*

APM supports the following AAA servers for high availability: RADIUS, Active Directory, LDAP, CRLDP, and TACACS+. APM supports high availability by providing the option to create a pool of server connections when you configure the supported type of AAA server.

Note: *If you use AAA with pools, such as RADIUS pools or Active Directory pools, APM assigns each pool member with a different number for the pool member's priority group value. Since APM does not support AAA load balancing, APM must define each pool member with a different priority group. The priority group number increases automatically with each created pool member.*

Task summary for configuring for LDAPS authentication

This task list includes all steps required to set up this configuration. If you are adding LDAPS authentication to an existing access policy, you do not need to create another access profile and the access policy might already include a logon page.

Task list

Configuring an LDAPS AAA server in APM
Creating an access profile
Configuring LDAPS authentication
Creating a virtual server for LDAPS
Testing LDAPS authentication

Configuring an LDAPS AAA server in APM

You create an LDAPS AAA server when you need to encrypt authentication messages between Access Policy Manager® (APM®) and the LDAP server.

1. Select **Access Policy** > **AAA Servers** > **LDAP**.
   The LDAP Servers screen displays.
2. Click **Create**.
   The New Server properties screen opens.
3. In the **Name** field, type a unique name for the authentication server.
4. For the **Server Connection** setting, select **Use Pool** even if you have only one LDAP server.
5. In the **Server Pool Name** field, type a name for the AAA server pool.
6. Populate the **Server Addresses** field by typing the IP address of a pool member and clicking **Add**.
   Type the IP address of an external LDAP server. If you have more than one pool member, repeat this step.
7. For the **Mode** setting, select `LDAPS`.
8. In the **Service Port** field, retain the default port number for LDAPS, `636`, or type the port number for the SSL service on the server.
9. In the **Admin DN** field, type the distinguished name (DN) of the user with administrator rights.
   Type the value in this format: `CN=administrator,CN=users,DC=sales,DC=mycompany,DC=com`.
10. In the **Admin Password** field, type the administrative password for the server.
11. In the **Verify Admin Password** field, re-type the administrative password for the server.
12. From the **SSL Profile (Server)** list, select an SSL server profile.
    You can select the default profile, serverssl, if you do not need a custom SSL profile.
    LDAPS is achieved by directing LDAP traffic over a virtual server that uses server-side SSL to communicate with the LDAP server.
13. Click **Finished**.
    The new server displays on the list.

The new LDAPS server displays on the LDAP Server list.

Creating an access profile

You create an access profile to provide the access policy configuration for a virtual server that establishes a secured session.

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. Click **Create**.
   The New Profile screen opens.
3. Type a name for the access profile.
4. From the **Profile Type** list, select one:
   - **APM-LTM** - Select for a web access management configuration.
   - **SSO** - Select only when you do not need to configure an access policy.
   - **SWG - Explicit** - Select to configure access using Secure Web Gateway explicit forward proxy.

- **SWG - Transparent** - Select to configure access using Secure Web Gateway transparent forward proxy.
- **SSL-VPN** - Select for other types of access, such as network access, portal access, application access. (Most access policy items are available for this type.)
- **ALL** - Select for any type of access.

Additional settings display.

5. In the Language Settings area, add and remove accepted languages, and set the default language.

   A browser uses the highest priority accepted language. If no browser language matches the accepted languages list, the browser uses the default language.

6. Click **Finished**.

This creates an access profile with a default access policy.

Configuring LDAPS authentication

You configure an access policy with an LDAP Auth action to provide LDAP authentication for users.

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. In the Access Policy column, click the **Edit** link for the access profile you want to configure.
   The visual policy editor opens the access policy in a separate screen.
3. Click the **(+)** icon anywhere in the access policy to add a new action item.
   A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
4. On the Logon tab, select **Logon Page** and click the **Add Item** button.
   The Logon Page Agent properties screen opens.
5. Make any changes that you require to the logon page properties and click **Save**.
   The properties screen closes and the visual policy editor displays.
6. Click the **(+)** icon anywhere in the access policy to add a new action item.
   A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
7. On the Authentication tab, select **LDAP Auth** and click **Add Item**.
8. From the **Server** list, select an AAA LDAP server.

   The LDAP Auth action uses SSL connections if you select an LDAP AAA server that is configured for LDAPS.

9. Specify the **SearchDN**, and **SearchFilter** settings.

   SearchDN is the base DN from which the search is done.

10. Click **Save**.
    The properties screen closes and the visual policy editor displays.
11. Click **Apply Access Policy** to save your configuration.

This creates a basic access policy that collects credentials and uses them to authenticate with an LDAP server over SSL. In practice, an access policy might include additional types of authentication and might also assign ACLS and resources

---

Important: *If you use LDAP Query, Access Policy Manager® does not query for the primary group and add it to the* `memberOf` *attribute. You must manually look up the attribute* `memberOf` *as well as the primary group.*

Creating a virtual server for LDAPS

You should have an Access Policy Manager® LDAP AAA server configured in LDAPS mode.

You create a virtual server to handle LDAP traffic and to encrypt authentication messages between Access Policy Manager® and the LDAP server.

Note: *An AAA server does not load-balance. Do not select a local traffic pool for this virtual server.*

---

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. From the **Configuration** list, select **Advanced**.
5. For the **Destination** setting in the **Address** field, type the IP address for the external LDAP server.
   This IP address must match a server address configured in the LDAP AAA server.
6. In the **Service Port** field, type the port number for the LDAP server.
   The server port (389) is the virtual port used as the external LDAP server's service port.

   Note: *The LDAP AAA server uses the external LDAP server's SSL service port.*

   ---
7. From the **SSL Profile (Server)** list, select `serverssl`.
   This ensures the SSL connection between the virtual server and the external LDAP server is in place.
8. From the **Source Address Translation** list, select **Auto Map**.
9. Click **Finished**.

Testing LDAPS authentication

Before starting this procedure, make sure that all the appropriate steps were performed to create an LDAPS authentication.

1. Ensure that LDAP authentication works in your environment.
   An intermediate virtual server should not exist for this verification step.
2. Create an access policy that uses a AAA object that points directly to the LDAP server.
3. Add an intermediate virtual server without a server-side SSL profile.
   Using the same access policy that you just created, modify the AAA object to point to a virtual server.
4. Implement LDAPS by enabling server side SSL, and change the pool member to use port `636`.
5. Review the log messages in Access Policy Manager® reports.
6. Make sure to set the Access Policy log level to **Debug**.
   To set log levels, see **System > Logs > Configurations > Options** > **.**
7. Review the log for LDAP messages and locate and confirm that the bind and search operation succeeds.

Testing AAA high availability for supported authentication servers

To effectively test that high availability works for your authentication servers, you should have two servers that are accessible, where you can remove one of them from the network.

Note: *High availability is supported for these authentication server types only: RADIUS, Active Directory, LDAP, CRLDP, and TACACS+.*

If you configured a supported authentication server type to use a pool of connection servers, you can test the configuration using these steps.

1. Begin a `tcpdump` on the Access Policy Manager®, using a protocol analyzer, and scanning for packets destined for the specific port for your authentication server.
2. Log in to the virtual server with both servers active.
3. Using the `tcpdump` records, verify that the requests are being sent to the higher priority server.
4. Log out of the virtual server.
5. Disable the higher-priority server.
6. Log in to the virtual server again.
7. Verify that the request is being sent to the other server.
8. Log out again, re-enabling the server, and try one more time to verify that the new requests are being sent to the high priority server.

Example of LDAP auth and query default rules

In this example, after successful authentication, the system retrieves a user group using an LDAP query. Resources are assigned to users and users are directed to a webtop if the user group has access to the network access resources.

In this figure, the default branch rule for LDAP query was changed to check for a specific user group attribute.

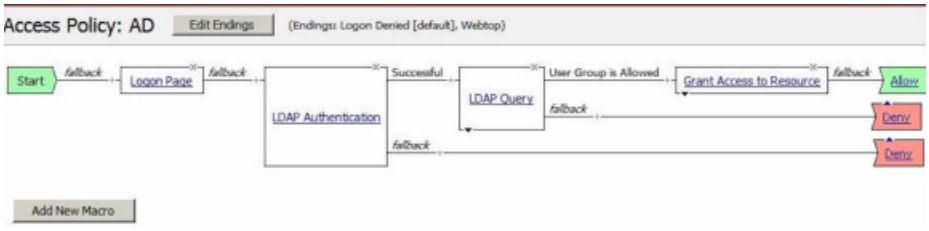


Figure 4: Example of an access policy for LDAP auth query

LDAP authentication session variables

When the LDAP Auth access policy item runs, it populates session variables which are then available for use in access policy rules. The tables list the session variables for the LDAP Auth access policy items and for a logon access policy item.

Session variables for LDAP authentication

Session Variable	Description
`session.ldap.last.authresult`	Provides the result of the LDAP authentication. The available values are: • 0: Failed • 1: Passed
`session.ldap.last.errmsg`	Useful for troubleshooting, and contains the last error message generated for LDAP, for example `aad2a221.ldap.last.errmsg`.

Common session variables

Session Variable	Description
`session.logon.last.username`	Provides user credentials. The `username` string is stored after encrypting, using the system's client key.
`session.logon.last.password`	Provides user credentials. The `password` string is stored after encrypting, using the system's client key.

UserDN settings in LDAP

The following is an example of a typical UserDN usage for LDAP.

Access Policy Manager® attempts to bind with the LDAP server using the supplied DN and user-entered password. If the bind succeeds, that is, authentication succeeds, the user is validated. If the bind fails, the authentication fails. This value is a fully qualified DN of the user with rights to run the query. Specify this value in lowercase and without spaces to ensure compatibility with some specific LDAP servers. The specific content of this string depends on your directory layout.

For example, in an LDAP structure, a typical UserDN for query would be similar to the following string: *`cn=%{session.logon.last.username}, cn=users, dc=sales, dc=com.`*

Access Policy Manager supports using session variables in the **SearchFilter**, **SearchDN**, and **UserDN** settings.

For example, if you want to use the user's CN from the user's SSL certificate as input in one of these fields, you can use the session variable *`session.ssl.cert.last.cn`* in place of *`session.logon.last.username`*.

LDAP authentication and query troubleshooting tips

You might run into problems with LDAP authentication and query in some instances. Follow these tips to try to resolve any issues you might encounter.

LDAP auth and query troubleshooting

Possible error messages	Possible explanations and corrective actions
`LDAP auth failed`	• User name or password does not match records. • No LDAP server is associated with the LDAP Auth agent. • The target LDAP server host/port information associated with the LDAP Auth agent might be invalid. • The target LDAP service might be not accessible.
`LDAP query failed`	• The specified administrative credential is incorrect. • If no administrative credential is specified, then the user name or password does not match. • No LDAP server is associated with the LDAP query agent. • The target LDAP server host/port information associated with the LDAP query agent might be invalid. • The target LDAP service might be not accessible. • If the LDAP query is successfully, then check whether the LDAP query Rules are properly configured.

Additional troubleshooting tips for LDAP authentication

You should	Steps to take
Check that your access policy is attempting to perform authentication	• Refer to the message boxes in your access policy to display information on what the access policy is attempting to do. • Refer to `/var/log/apm` to view authentication attempts by the access policy. ***Note:*** *Make sure that your log level is set to the appropriate level. The default log level is* `notice`
Confirm network connectivity	• Access the Access Policy Manager® through the command line interface and check your connectivity by pinging the LDAP server using the host entry in the AAA Server box. • Confirm that the LDAP port `389` is not blocked between the Access Policy Manager and the LDAP server.
Confirm network connectivity	• Access the Access Policy Manager through the command line interface and check your connectivity by pinging the LDAP server using the host entry in the AAA Server box. • Confirm that the LDAP port `389` is not blocked between the Access Policy Manager and the LDAP server.
Check the LDAP server configuration	• Verify that the administrative credentials are correct on the LDAP server, and that they match the credentials used by the AAA entry. ***Note:*** *A good test is to use full administrative credentials with all rights. If that works, you can use less powerful credentials for verification.*
Capture a TCP dump	• Take a TCP dump from the Access Policy Manager when authentication attempts are made. For example, %tcpdump-i 1.1 -s /tmp/dump. You must first determine what interface the self-IP is on. These TCP dumps indicate activities between the Access Policy Manager and the authentication server.

You should	Steps to take
	• Run the authentication test. After authentication fails, stop the TCP dump, and download the TCP dump to a client system, and use an analyzer to troubleshoot. ***Important:*** *If you decide to escalate the issue to customer support, you must provide a capture of the TCP dump when you encounter authentication issues that you cannot otherwise resolve on your own.*

Chapter 5

LDAP Query

- *About LDAP queries*
- *About nested groups in Active Directory and LDAP queries*
- *About how APM handles binary values in LDAP attributes*
- *Adding an LDAP query to an access policy*
- *Example of LDAP auth and query default rules*
- *Session variables in LDAP query properties*
- *LDAP query session variables*
- *LDAP authentication and query troubleshooting tips*

About LDAP queries

When running the LDAP Query access policy item, Access Policy Manager® (APM®) queries an external LDAP server for additional information about the user.

Important: *If you use LDAP query, Access Policy Manager does not query for the primary group and add it to the* `memberOf` *attribute. You must look up the attribute* `memberOf`*, as well as the primary group, manually.*

The LDAP Query item does not authenticate user credentials. To authenticate users, use another or an additional authentication item in the access policy.

About nested groups in Active Directory and LDAP queries

A *nested group* is a group that is a member of another group. For example, group1 is a member of group3 and group4. A user, user1, that belongs to group1 and group2 also belongs to group3 and group4 through nesting.

Whether AD Query and LDAP Query return nested groups in session variables

The AD Query and LDAP Query access policy items return and store the groups to which a user belongs in the `memberOf` session variable.

The contents of the `memberOf` session variable differ depending on whether the **Fetch Nested Group** setting is enabled or disabled in AD Query or LDAP Query properties:

- Enabled - The `memberOf` session variable contains all groups to which the user belongs. As in the example, this includes group1, group2, group3, and group4.
- Disabled - The `memberOf` session variable contains groups to which the user belongs directly. Based on the example, this would be group1 and group2.

About how APM handles binary values in LDAP attributes

For LDAP, Access Policy Manager® (APM®) converts an attribute value to hex only if the value contains unprintable characters. If the session variable contains several values, and one or more of those values is unprintable, then APM converts only those particular values to hex.

Case 1:
Handling of attributes with single value:

```
9302eb80.session.ldap.last.attr.objectGUID 34 /
0xfef232d3039be9409a72bfc60bf2a6d0
```

Case 2:
Handling of attributes with multiple values (mix of binary and non-binary values):

```
29302eb80.session.ldap.last.attr.memberOf 251 | /
CN=printable group,OU=groups,OU=someco,DC=smith, /
DC=labt,DC=fp,DC=somelabnet,DC=com | /
0x434e3d756e7072696e7461626c6520c2bdc2a12067726f75702c4f553d67726f7570732c4f553d66352c
 /
44433d73686572776f6f642c44433d6c6162742c44433d66702c44433d66356e65742c44433d636f6d
 |
```

Adding an LDAP query to an access policy

Before you add an LDAP query to an access policy, you must have at least one LDAP AAA server configured. You should also have an access profile that is configured with actions to authenticate the user.

You add an LDAP query to an access policy to get information about a user. APM stores the attributes it retrieves in session variables.

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. In the Access Policy column, click the **Edit** link for the access profile you want to configure.
   The visual policy editor opens the access policy in a separate screen.
3. Click the **(+)** icon anywhere in the access policy to add a new action item.
   A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
4. On the Authentication tab, select **LDAP Query** and click **Add Item**.
5. From the **Server** list, select an AAA LDAP server.

   An LDAP Query uses SSL connections when you select an LDAP AAA server that is configured for LDAPS.
6. Specify the **SearchDN**, and **SearchFilter** settings.

   SearchDN is the base DN from which the search is done.
7. Click **Save**.
   The properties screen closes and the visual policy editor displays.
8. Click **Apply Access Policy** to save your configuration.

This adds an LDAP Query to an existing access policy.

Example of LDAP auth and query default rules

In this example, after successful authentication, the system retrieves a user group using an LDAP query. Resources are assigned to users and users are directed to a webtop if the user group has access to the network access resources.

In this figure, the default branch rule for LDAP query was changed to check for a specific user group attribute.

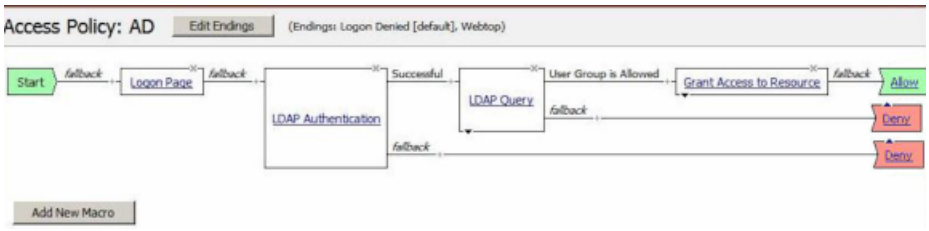


Figure 5: Example of an access policy for LDAP auth query

Session variables in LDAP query properties

You can use session variables to configure properties for the LDAP query access policy item. The properties are listed in the table.

Property	Example value	Description
SearchFilter	`(sAMAccountName=%{session.logon.last.username})`	Populates the `SearchFilter` parameter with the username from the current session.
UserDN	*`cn=%{session.logon.last.username}, cn=users, dc=sales, dc=com.`*	A typical UserDN for query in an LDAP structure.
SearchDN	*`session.ssl.cert.last.cn`*	Uses the user CN from the SSL certificate. Useful as a value for any property in this table.

LDAP query session variables

When the LDAP Query access policy item runs, it populates session variables which are then available for use in access policy rules. The tables list the session variables for the LDAP query access policy item and for a logon access policy item.

Session variables for LDAP query

Session Variable	Description
`session.ldap.last.queryresult`	Provides the result of the LDAP query. The available values are: • 0: Failed • 1: Passed
`session.ldap.last.attr.$attr_name`	*`$attr_name`* is a value that represents the user's attributes received during LDAP/query. Each attribute is converted to separate session variables.
`session.ldap.last.errmsg`	Useful for troubleshooting, and contains the last error message generated for LDAP, for example `aad2a221.ldap.last.errmsg`.

Common session variables

Session Variable	Description
`session.logon.last.username`	Provides user credentials. The `username` string is stored after encrypting, using the system's client key.
`session.logon.last.password`	Provides user credentials. The `password` string is stored after encrypting, using the system's client key.

LDAP authentication and query troubleshooting tips

You might run into problems with LDAP authentication and query in some instances. Follow these tips to try to resolve any issues you might encounter.

LDAP auth and query troubleshooting

Possible error messages	Possible explanations and corrective actions
`LDAP auth failed`	• User name or password does not match records. • No LDAP server is associated with the LDAP Auth agent. • The target LDAP server host/port information associated with the LDAP Auth agent might be invalid. • The target LDAP service might be not accessible.
`LDAP query failed`	• The specified administrative credential is incorrect. • If no administrative credential is specified, then the user name or password does not match. • No LDAP server is associated with the LDAP query agent. • The target LDAP server host/port information associated with the LDAP query agent might be invalid. • The target LDAP service might be not accessible. • If the LDAP query is successfully, then check whether the LDAP query Rules are properly configured.

Additional troubleshooting tips for LDAP authentication

You should	Steps to take
Check that your access policy is attempting to perform authentication	• Refer to the message boxes in your access policy to display information on what the access policy is attempting to do. • Refer to `/var/log/apm` to view authentication attempts by the access policy. ***Note:*** *Make sure that your log level is set to the appropriate level. The default log level is* `notice`
Confirm network connectivity	• Access the Access Policy Manager® through the command line interface and check your connectivity by pinging the LDAP server using the host entry in the AAA Server box. • Confirm that the LDAP port `389` is not blocked between the Access Policy Manager and the LDAP server.

You should	Steps to take
Confirm network connectivity	• Access the Access Policy Manager through the command line interface and check your connectivity by pinging the LDAP server using the host entry in the AAA Server box. • Confirm that the LDAP port `389` is not blocked between the Access Policy Manager and the LDAP server.
Check the LDAP server configuration	• Verify that the administrative credentials are correct on the LDAP server, and that they match the credentials used by the AAA entry. ***Note:*** *A good test is to use full administrative credentials with all rights. If that works, you can use less powerful credentials for verification.*
Capture a TCP dump	• Take a TCP dump from the Access Policy Manager when authentication attempts are made. For example, %tcpdump-i 1.1 -s /tmp/dump. You must first determine what interface the self-IP is on. These TCP dumps indicate activities between the Access Policy Manager and the authentication server. • Run the authentication test. After authentication fails, stop the TCP dump, and download the TCP dump to a client system, and use an analyzer to troubleshoot. ***Important:*** *If you decide to escalate the issue to customer support, you must provide a capture of the TCP dump when you encounter authentication issues that you cannot otherwise resolve on your own.*

Chapter 6

RSA SecurID Authentication

- *About RSA SecurID authentication*
- *About RSA SecurID configuration requirements for APM AAA*
- *Task summary for configuring for RSA SecurID authentication*
- *Access policy example for RSA and AD authentication*
- *RSA SecurID session variables for access policy rules*
- *RSA SecurID on Windows using RADIUS configuration troubleshooting tips*

About RSA SecurID authentication

RSA SecurID is a two-factor authentication mechanism based on a user PIN or password and code that an authenticator generates and provides to the user.

A *token* is an authentication code generated every 60 seconds by an authenticator (hardware or software) assigned to the user.

Figure 6: How Access Policy Manager works with RSA SecurID

1. The client submits the user name and PIN code to Access Policy Manager®.
2. Access Policy Manager sends the user-specified inputs to the RSA authentication server.
3. Based on the authentication results, Access Policy Manager grants or denies access to the client.

About RSA SecurID configuration requirements for APM AAA

Before you can use a SecurID AAA server in Access Policy Manager® (APM®), you need to meet specific requirements for configuration elements and settings on RSA SecurID, as described here.

Authentication agent

To provide RSA SecurID authentication for APM, the RSA Authentication Manager requires an authentication agent for APM in its database.

To create an authentication agent from the RSA Security Console, you need:

- Hostname
- IP addresses for all network interfaces
- Agent Type (set to Standard Agent)

RADIUS client

To provide RSA SecurID authentication for APM, RSA Authentication Manager requires a RADIUS client that corresponds to the authentication agent for APM.

To create a RADIUS client from the RSA Security Console, you need:

- Hostname
- IP addresses for all network interface
- RADIUS secret (this RADIUS secret must match the corresponding RADIUS secret on the APM system).

Character requirements setting in a SecurID token policy

To avoid a problem in the RSA SDK with alphabetic-only PIN policies, do not use them. When you set up a SecurID token policy, set the character requirements to one of these values:

- Require numeric PINs
- Allow alpha-numeric PINs

Task summary for configuring for RSA SecurID authentication

This task list includes all steps required to set up this configuration and provides an example access policy that uses both RSA SecurID and Active Directory authentication. It is only an example. If you are adding RSA SecurID authentication to an existing access policy, you do not need to create another access profile.

Task list

Configuring a SecurID AAA server in APM
Creating an access profile
Configuring RSA SecurID authentication in an access policy
Creating a virtual server

Configuring a SecurID AAA server in APM

Configure a SecurID AAA server for Access Policy Manager® (APM®) to request RSA SecurID authentication from an RSA Manager authentication server.

1. On the Main tab, click **Access Policy** > **AAA Servers**.
   The AAA Servers list screen opens.
2. On the menu bar, click **AAA Servers By Type**, and select **SecurID**.
   The SecurID screen opens and displays the servers list.
3. Click **Create**.
   The New Server properties screen opens.
4. In the **Name** field, type a unique name for the authentication server.
5. In the Configuration area, for the **Agent Host IP Address (must match the IP address in SecurID Configuration File)** setting, select an option as appropriate:
   - **Select from Self IP List**: Choose this when there is no NAT device between APM and the RSA Authentication Manager. Select an IP from the list of those configured on the BIG-IP® system (in the Network area of the Configuration utility).
   - **Other**: Choose this when there is a NAT device in the network path between Access Policy Manager and the RSA Authentication Manager server. If selected, type the address as translated by the NAT device.
6. For the **SecurID Configuration File** setting, browse to upload the `sdconf.rec` file.
   Consult your RSA Authentication Manager administrator to generate this file for you.
7. Click **Finished**.
   The new server displays on the list.

This adds a new RSA SecurID server to the AAA Servers list.

Creating an access profile

You create an access profile to provide the access policy configuration for a virtual server that establishes a secured session.

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. Click **Create**.
   The New Profile screen opens.
3. Type a name for the access profile.
4. From the **Profile Type** list, select one:
   - **APM-LTM** - Select for a web access management configuration.
   - **SSO** - Select only when you do not need to configure an access policy.
   - **SWG - Explicit** - Select to configure access using Secure Web Gateway explicit forward proxy.
   - **SWG - Transparent** - Select to configure access using Secure Web Gateway transparent forward proxy.
   - **SSL-VPN** - Select for other types of access, such as network access, portal access, application access. (Most access policy items are available for this type.)
   - **ALL** - Select for any type of access.

   Additional settings display.
5. In the Language Settings area, add and remove accepted languages, and set the default language.
   A browser uses the highest priority accepted language. If no browser language matches the accepted languages list, the browser uses the default language.
6. Click **Finished**.

This creates an access profile with a default access policy.

Configuring RSA SecurID authentication in an access policy

Before you add RSA SecurID authentication to an access policy, you must have at least one AAA SecurID server configured in Access Policy Manager® (APM®). You might need an AAA server configured for another type of authentication, depending on the number of authentication actions that you plan to add to this access policy. This access policy uses Active Directory authentication in addition to SecurID; in this case, an Active Directory AAA server is required.

You add RSA SecurID authentication to an access policy so that APM can request RSA SecurID authentication using the AAA SecurID server that you specify.

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. In the Access Policy column, click the **Edit** link for the access profile you want to configure.
   The visual policy editor opens the access policy in a separate screen.
3. Click the **(+)** icon anywhere in the access policy to add a new action item.
   A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
4. On the Logon tab, select **Logon Page** and click the **Add Item** button.
   The Logon Page Agent properties screen opens.
5. To customize the Logon Page to prompt for a token code in addition to a password, perform these substeps:

Add a second password field to the logon page and supply the appropriate prompts for both password fields.

a) From the **Type** list in row 3, select **password**.
b) In the **Post Variable Name** field in row 3, type `password1`.

   The name password1 is an example.

c) From the **Session Variable Name** field in row 3, type `password1`.

   The name password1 is an example. If you type `password1`, the name password1 becomes part of the session variable name, *`session.logon.last.password1`*. APM stores user input for the field in this session variable.

   You now have two fields that accept passwords on this Logon Page. Next you must set the prompts that display for each password field. This access policy runs RSA SecurID authentication first and another type of authentication afterward.

d) In the Customization area in **Logon Page Input Field #2**, in place of the text `Password` type `RSA Token` or the wording of your choice,
e) In **Logon Page Input Field #3**, type a prompt for the other type of authentication, for example `Password`.
f) Click **Save**.
   The properties screen closes and the visual policy editor is displayed.

6. Click the **(+)** icon anywhere in the access policy to add a new action item.
   A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
7. On the Authentication tab, select **RSA SecurID** and click **Add Item**.
   A properties popup screen opens.
8. From the **AAA Server** list in the properties popup screen, select the SecurID AAA server that you want to associate to the agent.
9. Set **Max Logon Attempts** to a value from from 1 to 5.

Note: *To use this access policy for Citrix Receiver client access, you must set* ***Max Logon Attempts*** *to 1.*

---

10. Click **Save**.
    The properties screen closes and the visual policy editor displays.
11. Add a Variable Assign action after the Logon Page action.

    Authentication actions use the password in the *`session.last.logon.password`* session variable. When the access policy runs and reaches this point, the RSA token code is stored in that session variable.

    After you add the Variable Assign action, a Properties popup screen displays.
12. On the Properties screen, add an entry to replace the contents of the *`session.last.logon.password`* session variable with the password stored in the *`session.last.logon.password1`* session variable:

    a) Click **Add new entry**.
       An **empty** entry appears in the Assignment table.
    b) Click the **change** link in the new entry.
       A popup screen opens.
    c) From the left-side list, select **Custom Variable** (the default), and type *`session.logon.last.password`*.
    d) From the right-side list, select **Custom Expression** (the default), and type `expr { "[mcget -secure session.logon.last.password1] }`.
    e) Click **Finished**.

       The popup screen closes.

f) Click **Save**.
The properties screen closes and the visual policy editor is displayed.

This example adds an AD Auth access policy item as a second type of authentication. You can add an authentication access policy item other than AD Auth.

The *session.logon.last.password* session variable now contains the user-entered password.

13. On the fallback branch after the previous action, click the **(+)** icon to add an item to the access policy.
A popup screen opens.
14. On the Authentication tab, select **AD Auth**.
A properties screen displays.
15. From the **Server** list, select a server.
16. To support Citrix Receiver clients, you must set **Max Logon Attempts** to 1.
17. Click **Save**.
The properties screen closes and the visual policy editor displays.
18. Add another authentication action and any other actions you require.
19. Click **Apply Access Policy** to save your configuration.

This adds RSA SecurID AAA authentication to the access policy and a second type of authentication.

Creating a virtual server

When creating a virtual server for an access policy, specify that the virtual server is a host virtual server, and not a network virtual server.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, select **Host** and in the **Address** field, type the IP address for the virtual server.
5. In the **Service Port** field, type a port number or select a service name from the **Service Port** list.
6. From the **HTTP Profile** list, select **http**.
7. If you use server SSL for this connection, from the **SSL Profile (Server)** list, select a server SSL profile.
8. If you use client SSL for this profile, from the **SSL Profile (Client)** list, select a client SSL profile.
9. In the Access Policy area, from the **Access Profile** list, select the access profile.
10. From the **Connectivity Profile** list, select a connectivity profile.
You can select the default connectivity profile, **connectivity** if you have not defined a specific profile for the traffic that is directed to this virtual server.
11. Click **Finished**.

You have configured a host virtual server and associated an access profile with it.

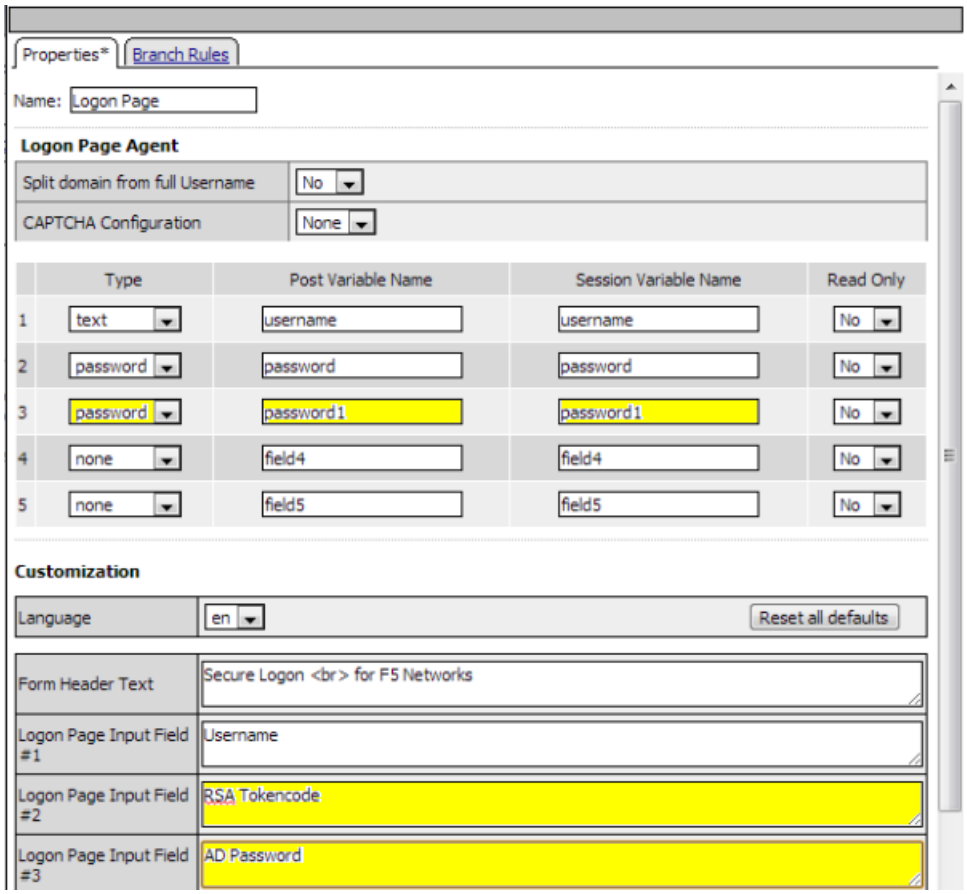
Access policy example for RSA and AD authentication

Typically, when you configure an authentication action, you precede it with a Logon Page action to collect credentials. This example describes how to include more than one authentication item (RSA and AD authentication) in an access policy and present a Logon Page only once.

Access policy with RSA SecurID and AD Auth actions

In this example, if the Logon Page action is not customized, the access policy passes the same credentials to both the RSA SecurID and AD Auth authentication agents. But RSA SecurID accepts a user name and a token at logon, while Active Directory accepts a user name and password. To accommodate these differences, customize the Logon Page item.

Logon Page customization: how to collect a token and a password



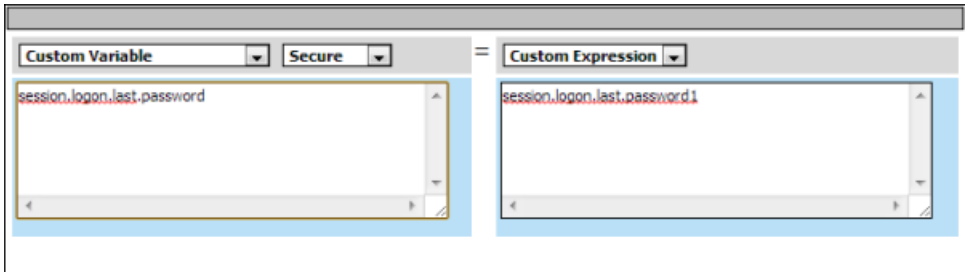
The first highlighted entry defines a second password field. The second password is stored in the `session.variable.last.password1` variable.

Note: *Although the second password is stored in a session variable, it is not the session variable,* `session.variable.last.password`*, from which an authentication agent accepts the password.*

The highlighted entries in the Customization area change the labels that the Logon Page displays, from Password to `RSA Token Code` for the first password and to `AD Password` for the second password.

Variable Assign action: How to pass the AD Password to the AD Auth action

Use the Variable Assign action to provide the appropriate password before the AD Auth action occurs.



The Variable Assign action moves the AD Auth password, stored in *`session.variable.last.password1`*, to the *`session.variable.last.password`* variable.

RSA SecurID session variables for access policy rules

When the RSA SecurID access policy item runs, it populates session variables which are then available for use in access policy rules. The tables list the session variables for the RSA SecurID access policy item and a logon access policy item.

Session variables for RSA SecurID

Session Variable	Description
`session.securid.last.result`	Provides the result of the RSA SecurID authentication. The available values are: • 0: Failed • 1: Passed

Common session variables

Session Variable	Description
`session.logon.last.username`	Provides user credentials. The `username` string is stored after encrypting, using the system's client key.
`session.logon.last.password`	Provides user credentials. The `password` string is stored after encrypting, using the system's client key.

RSA SecurID on Windows using RADIUS configuration troubleshooting tips

You might run into problems with RSA SecurID on Windows using RADIUS configuration. Follow these tips to try to resolve any issues that you encounter.

RSA SecurID on Windows using RADIUS configuration troubleshooting

Possible error messages	Possible explanations and corrective actions
`The RADIUS server is inactive`	Even if the RADIUS server was started from the SecurID options window on the Windows SecurID server, the server might not be active. In Windows Services Manager, make sure that the server is set to start each time the server boots, and is currently running. RSA SecurID authentication using RADIUS takes place on a different port than the native securid ID.
`The SecurID is configured incorrectly for RADIUS authentication`	While using RSA SecurID over RADIUS, the SecurID server is a client of itself. The RADIUS service functions as a standalone process, and if the SecurID server is not set up as a client of itself, it rejects the Access Policy Manager ®authentication request and does not store anything in the logs.
`No response from the RSA SecurID server`	Check that RSA Authentication Manager is configured properly. To facilitate communication between Access Policy Manager and the RSA Authentication Manager, you must add an Authentication Agent record to the RSA Authentication Manager database. The Authentication Agent record identifies the Access Policy Manager within its database, and contains information about communication and encryption. To create the Authentication Agent record, you need this information. • Host name • IP addresses for all network interfaces When adding the Authentication Agent record, you should configure the Access Policy Manager as a Standard Agent. The RSA Authentication Manager uses this setting to determine how to communicate with Access Policy Manager. You must also add a RADIUS client that corresponds to the Authentication Agent. To create the RADIUS client, you need this information. • Host name • IP addresses for all network interfaces • RADIUS secret (This RADIUS secret must match the corresponding RADIUS secret on the Access Policy Manager.)

Chapter 7

RADIUS Authentication

- *About RADIUS authentication*
- *About AAA high availability*
- *Guidelines for setting up RADIUS authentication for AAA high availability*
- *About how APM handles binary values in RADIUS attributes*
- *Task summary for RADIUS authentication*
- *Testing AAA high availability for supported authentication servers*
- *RADIUS attributes*
- *RADIUS session variables for access policy rules*
- *RADIUS authentication and accounting troubleshooting tips*

About RADIUS authentication

Access Policy Manager® supports authenticating and authorizing the client against external RADIUS servers. When a client connects with the user name and password, Access Policy Manager authenticates against the external server on behalf of the client, and authorizes the client to access resources if the credentials are valid.

Figure 7: How RADIUS works

- The client requests access to network resources through Access Policy Manager.
- Access Policy Manager then issues a `RADIUS Access Request` message to the RADIUS server, requesting authorization to grant access.
- The RADIUS server then processes the request, and issues one of three responses to Access Policy Manager: `Access Accept`, `Access Challenge`, or `Access Reject`.

About AAA high availability

Using AAA high availability with Access Policy Manager® (APM®), you can configure multiple authentication servers to process requests, so that if one authentication server goes down or loses connectivity, the others can resume authentication requests, and new sessions can be established, as usual.

Note: *Although new authentications fail if the BIG-IP® system loses connectivity to the server, existing sessions are unaffected provided that they do not attempt to re-authenticate.*

APM supports the following AAA servers for high availability: RADIUS, Active Directory, LDAP, CRLDP, and TACACS+. APM supports high availability by providing the option to create a pool of server connections when you configure the supported type of AAA server.

Note: *If you use AAA with pools, such as RADIUS pools or Active Directory pools, APM assigns each pool member with a different number for the pool member's priority group value. Since APM does not support AAA load balancing, APM must define each pool member with a different priority group. The priority group number increases automatically with each created pool member.*

Guidelines for setting up RADIUS authentication for AAA high availability

When you use RADIUS as the authentication method for AAA high availability, there are general guidelines that you must follow when you set up your server connections.

- In a non-high availability environment, both the **Direct** and **Use Pool** options use the self IP address as a source IP address of the packet reaching the RADIUS server. For this scenario, you just need to add one IP address to the RADIUS allowed IP list to achieve this.
- In a high availability environment where the **Use Pool** option is used, the floating self IP address is used as a source IP of the RADIUS packet reaching the back-end. For this scenario, you need to add one self IP address (which is floating self IP address) to the RADIUS allowed IP list because the IP address is used even after a failover occurs.
- In a high availability environment where the **Direct** option is used, the self IP address is used as a source IP address of the RADIUS packet reaching the back-end. In this scenario, you need to add the self IP address from both active and standby devices to the RADIUS allowed IP list so that when failover occurs, the self IP address from the second device is accepted by the RADIUS server.

About how APM handles binary values in RADIUS attributes

For RADIUS authentication, Access Policy Manager® (APM®) converts an attribute value to hex if it contains unprintable characters, or if it is the `class` attribute. APM converts the class attribute to hex even if it contains only printable values (by attribute type). No other attributes are encoded to hex if they do not contain unprintable characters.

Case 1:

Handling of attributes with single value:

```
1bf80e04.session.radius.last.attr.class 62 /
0x5423061600001370001ac1d423301caa87483dadf740000000000000007
```

Case 2:

Handling of attributes with multiple values (mix of binary and non-binary values):

```
243be90d.session.radius.last.attr.class 119 0x6162636465666768696 /
a6b6c6d6e6f7071727374757677787 97a | 0x5422061500000137 0001ac1d423301caa87483
 /
dadf740000000000000006
```

If the attribute type does not require hex encoding, and some of the values are unprintable, then only those value(s) are encoded to hex.

```
3888eb70.session.radius.last.attr.login-lat-group 37 /
    0x6d7920bda12067726f757032 | mygroup1
```

Task summary for RADIUS authentication

This task list includes all steps required to set up this configuration. If you add RADIUS authentication to an existing access policy, you already have an access profile configured and the access policy might already include a logon access policy item.

Task list

Configuring a RADIUS AAA server in APM
Creating an access profile
Using RADIUS authentication in an access policy
Creating a virtual server

Configuring a RADIUS AAA server in APM

The Access Policy Manager ® (APM®) is a network access server (NAS) that operates as a client of the server configured here.

1. On the Main tab, click **Access Policy** > **AAA Servers**.
   The AAA Servers list screen opens.
2. On the Main tab, click **Access Policy** > **AAA Servers** > **RADIUS**.
   The RADIUS Servers screen displays.
3. Click **Create**.
   The New Server properties screen opens.
4. In the **Name** field, type a unique name for the authentication server.
5. For the **Mode** setting, select **Authentication**.
6. For the **Server Connection** setting, select one of these options:
   - Select **Use Pool** to set up high availability for the AAA server.
   - Select **Direct** to set up the AAA server for standalone functionality.
7. If you selected **Use Pool**, type a name in the **Server Pool Name field**.
   You create a pool of servers on this screen.
8. Provide the addresses required for your server connection:
   - If you selected **Direct**, type an IP address in the **Server Address** field.
   - If you selected **Use Pool**, for each pool member you want to add, type an IP address and click **Add**.

   Note: *When you configure a pool, you have the option to type the server address in route domain format:* `IPAddress%RouteDomain`.
9. In the **Authentication Service Port** field, type the authentication port number of your server. The default is `1812`.
10. In the **Secret** field, type the shared secret password of the server.
11. In the **Confirm Secret** field, re-type the shared secret password of the server.
12. Click **Finished**.
    The new server displays on the list.

The new AAA server displays on the RADIUS Servers list.

Creating an access profile

You create an access profile to provide the access policy configuration for a virtual server that establishes a secured session.

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.

2. Click **Create**.
   The New Profile screen opens.
3. Type a name for the access profile.
4. From the **Profile Type** list, select one:
   - **APM-LTM** - Select for a web access management configuration.
   - **SSO** - Select only when you do not need to configure an access policy.
   - **SWG - Explicit** - Select to configure access using Secure Web Gateway explicit forward proxy.
   - **SWG - Transparent** - Select to configure access using Secure Web Gateway transparent forward proxy.
   - **SSL-VPN** - Select for other types of access, such as network access, portal access, application access. (Most access policy items are available for this type.)
   - **ALL** - Select for any type of access.

   Additional settings display.
5. In the Language Settings area, add and remove accepted languages, and set the default language.
   A browser uses the highest priority accepted language. If no browser language matches the accepted languages list, the browser uses the default language.
6. Click **Finished**.

This creates an access profile with a default access policy.

Using RADIUS authentication in an access policy

You configure an access policy with a RADIUS Auth action to provide RADIUS authentication as one of authentication options for users trying to gain accesss.

Note: *You can use RADIUS authentication in addition to other authentication types. You can require that users pass at least one type of authentication or that they pass multiple types of authentication.*

---

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. In the Access Policy column, click the **Edit** link for the access profile you want to configure.
   The visual policy editor opens the access policy in a separate screen.
3. Click the **(+)** icon anywhere in the access policy to add a new action item.
   A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
4. On the Logon tab, select **Logon Page** and click the **Add Item** button.
   The Logon Page Agent properties screen opens.
5. Make any changes that you require to the logon page properties and click **Save**.
   The properties screen closes and the visual policy editor displays.
6. Click the **(+)** icon anywhere in the access policy to add a new action item.
   A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
7. From the Authentication tab, select **RADIUS Auth** and click **Add Item**.
   The popup screen closes. A Properties popup screen opens.
8. On the Properties popup screen from the **AAA Server** listselect the AAA RADIUS server you configured previously and click **Save**.
   The popup screen closes and the visual policy editor displays.
9. Complete the access policy:

a) Add any additional access policy items you require.
b) Change the ending from **Deny** to **Allow** on any access policy branch on which you want to grant access.

10. Click **Apply Access Policy** to save your configuration.

This creates an access policy that collects user credentials and uses them to authenticat with a RADIUS server..

For an access policy to go into effect on network traffic, you must add the access profile to a virtual server.

Creating a virtual server

When creating a virtual server for an access policy, specify that the virtual server is a host virtual server, and not a network virtual server.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, select **Host** and in the **Address** field, type the IP address for the virtual server.
5. In the **Service Port** field, type a port number or select a service name from the **Service Port** list.
6. From the **HTTP Profile** list, select **http**.
7. If you use server SSL for this connection, from the **SSL Profile (Server)** list, select a server SSL profile.
8. If you use client SSL for this profile, from the **SSL Profile (Client)** list, select a client SSL profile.
9. In the Access Policy area, from the **Access Profile** list, select the access profile.
10. From the **Connectivity Profile** list, select a connectivity profile.
You can select the default connectivity profile, **connectivity** if you have not defined a specific profile for the traffic that is directed to this virtual server.
11. Click **Finished**.

You have configured a host virtual server and associated an access profile with it.

Testing AAA high availability for supported authentication servers

To effectively test that high availability works for your authentication servers, you should have two servers that are accessible, where you can remove one of them from the network.

Note: *High availability is supported for these authentication server types only: RADIUS, Active Directory, LDAP, CRLDP, and TACACS+.*

If you configured a supported authentication server type to use a pool of connection servers, you can test the configuration using these steps.

1. Begin a `tcpdump` on the Access Policy Manager®, using a protocol analyzer, and scanning for packets destined for the specific port for your authentication server.

2. Log in to the virtual server with both servers active.
3. Using the `tcpdump` records, verify that the requests are being sent to the higher priority server.
4. Log out of the virtual server.
5. Disable the higher-priority server.
6. Log in to the virtual server again.
7. Verify that the request is being sent to the other server.
8. Log out again, re-enabling the server, and try one more time to verify that the new requests are being sent to the high priority server.

RADIUS attributes

The following table lists the specific RADIUS attributes that Access Policy Manager® sends with RADIUS requests.

Attribute	Purpose
`User-Name`	Indicates the name of the authenticated user.
`User-Password`	Indicates the password of the authenticated user.
`NAS-IP-Address`	Indicates the identifying IP Address of the NAS.
`NAS-IPv6-Address`	Indicates the identifying IPv6 Address of the NAS.
`NAS-Identifier`	Indicates the identifying name of the NAS .
`Service-Type`	Indicates the type of service the user has requested.
`NAS-Port`	Indicates the physical port number of the NAS that is authenticating the user.

RADIUS session variables for access policy rules

When the RADIUS Auth access policy item runs, it populates session variables which are then available for use in access policy rules. The tables list the session variables for the RADIUS authentication access policy item and for a logon access policy item.

Session variables for RADIUS

Session Variable	Description
`session.RADIUS.last.result`	Provides the result of the RADIUS authentication. The available values are: • 0: Failed • 1: Passed
`session.RADIUS.last.attr.$attr_name`	*`$attr_name`* is a value that represents the user's attributes received during RADIUS authentication. Each attribute is converted to separate session variables.
`session.RADIUS.last.errmsg`	Displays the error message for the last login. If `session.RADIUS.last.result` is set to 0, then

Session Variable	Description
	`session.RADIUS.last.errmsg` might be useful for troubleshooting purposes. Example: `c76a50c0.session.RADIUS.last.errmsg 13 Access-Reject`

Common session variables

Session Variable	Description
`session.logon.last.username`	Provides user credentials. The `username` string is stored after encrypting, using the system's client key.
`session.logon.last.password`	Provides user credentials. The `password` string is stored after encrypting, using the system's client key.

RADIUS authentication and accounting troubleshooting tips

You might run into problems with RADIUS authentication and accounting in some instances. Follow these tips to try to resolve any issues you might encounter.

RADIUS authentication and accounting access policy action troubleshooting

Possible error messages	Possible explanations and actions
`Authentication failed due to timeout`	• Verify that Access Policy Manager® is configured as a client on the RADIUS server. • You might have encountered a general network connection problem.
`Authentication failed due to RADIUS access reject`	• Verify that the shared secret on the RADIUS server is valid. • Verify that user credentials are entered correctly.

Additional troubleshooting tips for RADIUS authentication and accounting

Action	Steps
Check to see if your access policy is attempting to perform authentication	• Add message boxes to your access policy to display information about what the access policy is attempting to do. • Refer to `/var/log/apm` to view authentication and accounting attempts by the access policy. ***Note:*** *Make sure that your log level is set to the appropriate level. The default log level is* `notice`*.*
Check the RADIUS Server configuration	• Confirm that the Access Policy Manager is registered as a RADIUS client. Since the Access Policy Manager makes requests from the self IP address to the RADIUS server for authentication requests, the address of the self-IP address should be registered as a RADIUS client. • Check the RADIUS logs and check for any errors.

Action	Steps
Confirm network connectivity	• Access the BIG-IP® system through the command line interface and check your connectivity by pinging the RADIUS server using the host entry in the AAA Server box. • Confirm that the RADIUS port 1812 is not blocked between the Access Policy Manager and the RADIUS server.
Capture a TCP dump	• Take a TCP dump from the Access Policy Manager when authentication attempts are made. For example, %TCP dump-i 1.1 -s /tmp/dump. You must first determine what interface the self IP address is on. These TCP dumps indicate activities between the Access Policy Manager and the authentication server. • Run the authentication test. After authentication fails, stop the TCP dump, download the TCP dump records to a client system, and use an analyzer to troubleshoot. ***Important:*** *If you decide to escalate the issue to customer support, you must provide a capture of the TCP dump when you encounter authentication issues that you cannot otherwise resolve on your own.*

Chapter 8

RADIUS Accounting

- *About RADIUS accounting*
- *About how APM handles binary values in RADIUS attributes*
- *Configuring a RADIUS Accounting server in APM*
- *Adding RADIUS accounting to an access policy*
- *RADIUS authentication and accounting troubleshooting tips*

About RADIUS accounting

You can report user session information to an external RADIUS accounting server. If you select this mode only, the system assumes that you have set up another type of authentication method to authenticate and authorize your users to access their resources.

1. After RADIUS accounting runs successfully in an access policy, Access Policy Manager® sends an accounting start request message to the external RADIUS server. The `start` message typically contains the user's ID, networks address, point of attachment, and a unique session identifier.
2. When the session is destroyed, Access Policy Manager issues an accounting `stop` message to the external RADIUS server, providing information on the final usage in terms of time, packets transferred, data transferred, and reason for disconnect, as well as other information related to the user's access.

This accounting data is used primarily for billing, statistical, and general network monitoring purposes.

Note: *You can perform both RADIUS authentication and accounting actions. Keep in mind that if you select this mode, the RADIUS server and the RADIUS accounting server must run on different service ports.*

About how APM handles binary values in RADIUS attributes

For RADIUS authentication, Access Policy Manager® (APM®) converts an attribute value to hex if it contains unprintable characters, or if it is the `class` attribute. APM converts the class attribute to hex even if it contains only printable values (by attribute type). No other attributes are encoded to hex if they do not contain unprintable characters.

Case 1:

Handling of attributes with single value:

```
1bf80e04.session.radius.last.attr.class 62 /
0x5423061600001370001ac1d423301caa87483dadf740000000000000007
```

Case 2:

Handling of attributes with multiple values (mix of binary and non-binary values):

```
243be90d.session.radius.last.attr.class 119 0x6162636465666768696 /
a6b6c6d6e6f7071727374757677787 97a | 0x5422061500001370001ac1d423301caa87483
```

```
 /
dadf740000000000000006
```

If the attribute type does not require hex encoding, and some of the values are unprintable, then only those value(s) are encoded to hex.

```
3888eb70.session.radius.last.attr.login-lat-group 37 /
    0x6d7920bda12067726f757032 | mygroup1
```

Configuring a RADIUS Accounting server in APM

1. On the Main tab, click **Access Policy** > **AAA Servers** > **RADIUS**.
   The RADIUS Servers screen displays.
2. Click **Create**.
   The New Server properties screen opens.
3. In the **Name** field, type a unique name for the authentication server.
4. From the **Mode** list, select **Accounting**.
5. For the **Server Connection** setting, select one of these options:
   - Select **Use Pool** to set up high availability for the AAA server.
   - Select **Direct** to set up the AAA server for standalone functionality.
6. If you selected **Use Pool**, type a name in the **Server Pool Name field**.
   You create a pool of servers on this screen.
7. Provide the addresses required for your server connection:
   - If you selected **Direct**, type an IP address in the **Server Address** field.
   - If you selected **Use Pool**, for each pool member you want to add, type an IP address and click **Add**.

   Note: *When you configure a pool, you have the option to type the server address in route domain format:* `IPAddress%RouteDomain`.
8. If you selected **Use Pool**, you have the option to select a **Server Pool Monitor** to track the health of the server pool.
9. In the **Accounting Service Port** field, type the service port for your accounting server if the default value is not appropriate.
   The default is `1813`.
10. In the **Secret** field, type the shared secret password of the server.
11. In the **Confirm Secret** field, re-type the shared secret password of the server.
12. In the **Timeout** field, type a timeout interval (in seconds) for the AAA server.
    This setting is optional.
    If you use the **Timeout** setting, you can also use the **Retries** setting. If these settings are enabled, the Access Policy Manager attempts to reach the AAA server within the specified time frame, in seconds. If the server does not respond, the Access Policy Manager retries the authentication attempt, depending on how many retries you specify.
13. In the **Retries** field, type the number of times the BIG-IP system should try to make a connection to the server after the first attempt fails.

This setting is optional.

14. Click **Finished**.
    The new server displays on the list.

Adding RADIUS accounting to an access policy

Before you set up an access policy to use RADIUS accounting, you must have at least one RADIUS AAA server configured. You should also have an access profile that is configured with actions that authenticate the user.

You add a RADIUS accounting action to an access policy to send RADIUS start and stop messages to a RADIUS server. RADIUS accounting does not authenticate a user.

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. In the Access Policy column, click the **Edit** link for the access profile you want to configure.
   The visual policy editor opens the access policy in a separate screen.
3. Click the **(+)** icon anywhere in the access policy to add a new action item.
   A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
4. From the Authentication tab, select **RADIUS Acct** and click **Add Item**.
   The popup screen closes. A properties popup screen opens.
5. From the **AAA Server** list, select a RADIUS accounting server and click **Save**.
   The properties popup screen closes and the visual policy editor displays.
6. Click **Apply Access Policy** to save your configuration.

This adds the RADIUS accounting action to the access policy.

To put the access policy into effect, you must add it to a virtual server.

RADIUS authentication and accounting troubleshooting tips

You might run into problems with RADIUS authentication and accounting in some instances. Follow these tips to try to resolve any issues you might encounter.

RADIUS authentication and accounting access policy action troubleshooting

Possible error messages	Possible explanations and actions
`Authentication failed due to timeout`	• Verify that Access Policy Manager® is configured as a client on the RADIUS server. • You might have encountered a general network connection problem.
`Authentication failed due to RADIUS access reject`	• Verify that the shared secret on the RADIUS server is valid. • Verify that user credentials are entered correctly.

Additional troubleshooting tips for RADIUS authentication and accounting

Action	Steps
Check to see if your access policy is attempting to perform authentication	• Add message boxes to your access policy to display information about what the access policy is attempting to do. • Refer to `/var/log/apm` to view authentication and accounting attempts by the access policy. ***Note:*** *Make sure that your log level is set to the appropriate level. The default log level is* `notice`.
Check the RADIUS Server configuration	• Confirm that the Access Policy Manager is registered as a RADIUS client. Since the Access Policy Manager makes requests from the self IP address to the RADIUS server for authentication requests, the address of the self-IP address should be registered as a RADIUS client. • Check the RADIUS logs and check for any errors.
Confirm network connectivity	• Access the BIG-IP® system through the command line interface and check your connectivity by pinging the RADIUS server using the host entry in the AAA Server box. • Confirm that the RADIUS port 1812 is not blocked between the Access Policy Manager and the RADIUS server.
Capture a TCP dump	• Take a TCP dump from the Access Policy Manager when authentication attempts are made. For example, %TCP dump-i 1.1 -s /tmp/dump. You must first determine what interface the self IP address is on. These TCP dumps indicate activities between the Access Policy Manager and the authentication server. • Run the authentication test. After authentication fails, stop the TCP dump, download the TCP dump records to a client system, and use an analyzer to troubleshoot. ***Important:*** *If you decide to escalate the issue to customer support, you must provide a capture of the TCP dump when you encounter authentication issues that you cannot otherwise resolve on your own.*

Chapter 9

Kerberos Authentication with End-User Logons

- *About basic authentication and Kerberos end-user logon*
- *How does end-user logon work?*
- *About Kerberos authentication requirements*
- *Task summary for configuring end-user login support*
- *Access policy example for end-user login*
- *Kerberos authentication troubleshooting tips*

About basic authentication and Kerberos end-user logon

Access Policy Manager® (APM®) provides an alternative to a form-based login authentication method. This alternative method uses a browser login box that is triggered by an HTTP 401 response to collect credentials. A SPNEGO/Kerberos or basic authentication challenge can generate a HTTP 401 response.

This option is useful when a user is already logged in to the local domain and you want to avoid submitting an APM HTTP form for collecting user credentials. The browser automatically submits credentials to the server and bypasses the login box to collect the credentials again.

Note: *Because SPNEGO/Kerberos is a request-based authentication feature, the authentication process is different from other authentication methods, which run at session creation time. SPNEGO/Kerberos authentication can occur at any time during the session.*

The benefits of this feature include:

- Provides flexible login mechanism instead of restricting you to use only the form-based login method.
- Eliminates the need for domain users to explicitly type login information again to log in to Access Policy Manager.
- Eliminates the need for user password transmission with Kerberos method.

Important: *Administrators should not turn off the* ***KeepAlive*** *setting on the web server because turning that setting off might interfere with Kerberos authentication.*

How does end-user logon work?

To retrieve user credentials for end-user logon, you can use basic authentication or SPEGNO/Kerberos methods or both.

Basic authentication

Use this method to retrieve user credentials (user name and password) from a browser. You can think of this method as a replacement for form-based authentication used by the standard login screen. If you use basic authentication, the system populates the user name and password session variables, which can then be used by any other authentication actions, such as Active Directory or RADIUS.

SPNEGO/Kerberos

Use this method to retrieve user credentials through SPNEGO/Kerberos authentication header. With the Kerberos method, the client system must first join a domain and a Kerberos action must follow. The Kerberos action does not run immediately; it runs only when clients request SPNEGO/Kerberos authentication. By default, Kerberos authentication runs not only on the first request, but also on subsequent requests where authentication is needed, such as for new connections. Access Policy Manager® (APM®) validates the request by confirming that a valid ticket is present.

Note: *You can disable Kerberos per request-based authentication in the Kerberos authentication access policy item configuration in APM. If you disable it, authentication occurs while the access policy runs and subsequent authentications do not occur. In that case, end-user logon does not occur.*

Note: *You can achieve multi-domain support for Kerberos authentication through multiple virtual servers. Each virtual server must have its own access policy and its own Kerberos configuration.*

Both methods require that an HTTP 401 Response action item be configured in the access policy and that the authentication method be specified in the action item. In cases where both methods are selected, the browser determines which method to perform based on whether the system has joined a domain. The HTTP 401 Response action has two default branches to indicate whether basic authentication or Kerberos method is performed.

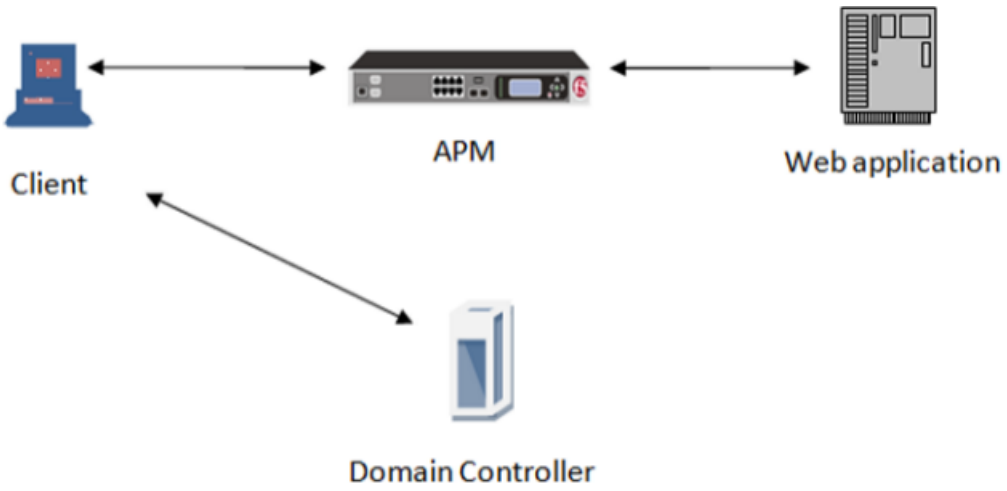


Figure 8: How SPNEGO/Kerberos end-user logon works

The end-user logon works with events happening in this order:

- The client becomes a member and connects to the domain.
- The client connects to a virtual server on the BIG-IP® system.
- The access policy runs and issues a 401 HTTP request action.
- If Kerberos is present, the browser forwards the Kerberos ticket along with the request when it receives the 401 HTTP request.
- Access Policy Manager validates the Kerberos ticket after the request is received and determines whether or not to permit the request.

About Kerberos authentication requirements

To configure Kerberos authentication, you must meet specific configuration requirements as described here.

Virtual server

The virtual server IP address and host name are necessary to configure DNS.

DNS configuration

Make sure you have the zone file and PTR record for the virtual server IP address. For example:

```
testbed.lab.companynet 10.10.4.100
```

Browser configuration

Configure the browser to use Kerberos. Typically, Internet Explorer is already configured for Kerberos; however, you might need to configure it for trusted sites. To use Firefox, you must configure it for negotiate authentication.

Task summary for configuring end-user login support

To set up this configuration, perform the procedures in the task list.

Task list

Joining a Kerberos user account to a domain
Configuring an AAA server for Kerberos authentication
Creating an access profile
Configuring an access policy for end-user logon support

Joining a Kerberos user account to a domain

To use Kerberos authentication, you need the client joined and connected to a domain and you need a keytab file.

1. Create a surrogate user in the domain.

   In this example, the hostname of the virtual server on the BIG-IP system is testbed.lab.companynet and the user name is john.

   ```
   setspn -U -A HTTP/testbed.lab.companynet john
   ```

2. Map the user account to the service account and generate a keytab file for the service.

   You can use the ktpass utility to do this. In this example, LAB.COMPANYNET specifies the Kerberos authentication realm.

   ```
   c:>ktpass -princ HTTP/testbed.lab.companynet.com@LAB.COMPANYNET -mapuser
   john@LAB.COMPANYNET -crypto rc4-hmac-nt -ptype KRB5_NT_SRV_HST -pass password
   -out c:\temp\john.keytab
   ```

Configuring an AAA server for Kerberos authentication

Configure a Kerberos AAA server so that you can add it to a Kerberos authentication action in an access policy.

1. On the Main tab, click **Access Policy** > **AAA Servers** > **Kerberos**.
   The Kerberos Servers list screen opens.
2. Click **Create**.
   The New Server properties screen opens.
3. In the **Name** field, type a unique name for the authentication server.
4. In the **Auth Realm** field, type a Kerberos authentication realm name (administrative name), such as `LAB.COMANYNET`.

   Type the realm name all uppercase; it is case-sensitive.
5. In the **Service Name** field, type a service name; for example, `HTTP`.
6. In the **Keytab File** area, click **Choose File** to locate and upload the keytab file.

   A keytab file contains Kerberos encryption keys (these are derived from the Kerberos password).
7. Click **Finished**.
   The new server displays on the list.

Creating an access profile

You create an access profile to provide the access policy configuration for a virtual server that establishes a secured session.

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. Click **Create**.
   The New Profile screen opens.
3. Type a name for the access profile.
4. From the **Profile Type** list, select one:
   - **APM-LTM** - Select for a web access management configuration.
   - **SSO** - Select only when you do not need to configure an access policy.
   - **SWG - Explicit** - Select to configure access using Secure Web Gateway explicit forward proxy.
   - **SWG - Transparent** - Select to configure access using Secure Web Gateway transparent forward proxy.
   - **SSL-VPN** - Select for other types of access, such as network access, portal access, application access. (Most access policy items are available for this type.)
   - **ALL** - Select for any type of access.

   Additional settings display.
5. In the Language Settings area, add and remove accepted languages, and set the default language.
   A browser uses the highest priority accepted language. If no browser language matches the accepted languages list, the browser uses the default language.
6. Click **Finished**.

This creates an access profile with a default access policy.

Configuring an access policy for end-user logon support

To use basic authentication in addition to Kerberos authentication, you need an AAA server configured for the authentication agent that you plan to use.

Configure an access policy like this one to handle basic and SPEGNO/Kerberos authentication challenges without submitting an Access Policy Manager® HTTP form to collect user credentials.

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. In the Access Policy column, click the **Edit** link for the access profile you want to configure.
   The visual policy editor opens the access policy in a separate screen.
3. Click the **(+)** icon anywhere in the access policy to add a new action item.
   A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
4. Under General Purpose, select **HTTP 401 Response**, and click **Add item**.
   A properties screen opens.
5. In the 401 Response Setting area from the **HTTP Auth Level** list, select **basic+negotiate**, and click **Save**.
   The properties screen closes. The visual policy editor displays the HTTP 401 Response item with 3 branches: Basic, Negotiate, and fallback.
6. To perform basic authentication, add an authentication server agent on the **Basic** branch.

7. To use the Kerberos authentication method:
   a) Add the **Kerberos Auth** agent on the **Negotiate** branch.
      After you add the Kerberos Auth item, a properties popup screen displays.
   b) On the properties screen for the **AAA Server** setting, select the Kerberos AAA server.
   c) Click **Save**.
      The properties screen closes and the visual policy editor is displayed.

8. Complete the access policy:
   a) Add any additional access policy items you require.
   b) Change the ending from **Deny** to **Allow** on any access policy branch on which you want to grant access.

9. Click **Apply Access Policy**.

For an access policy to go into effect, you must add the corresponding access profile to the virtual server.

Access policy example for end-user login

This is an example of an access policy with all the associated elements needed to successfully support the end-user login feature. Notice that separate branches are created automatically to support using either basic authentication or Kerberos method to retrieve user credentials.

Note: *For basic authentication, the user name and password validation occurs at the session creation time. After the access policy completes, the session cookie is used to validate the session.*

Note: *By default, Kerberos runs not only at the access policy run time but also at any time in the session.*

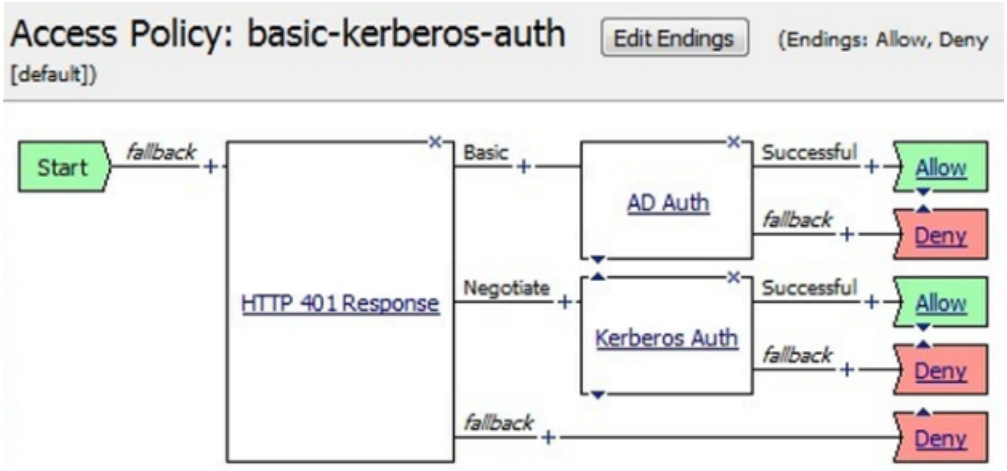


Figure 9: Example access policy for end-user login

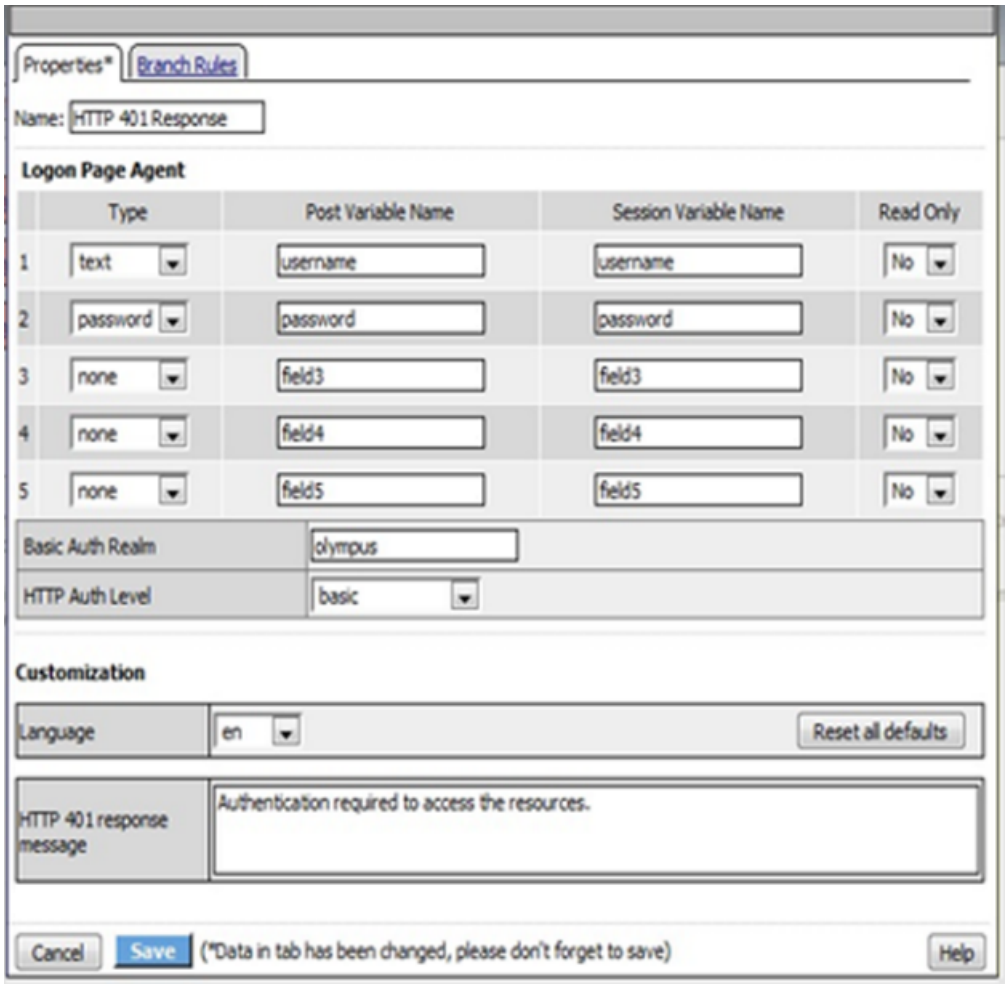


Figure 10: Example properties for an HTTP 401 response action

Figure 11: Example properties for a Kerberos Auth action on the Negotiate branch

Kerberos authentication troubleshooting tips

You might choose to verify Kerberos authentication configurations in some instances. Use these troubleshooting tips to help resolve any issues you might encounter.

Verify the keytab file

From the command line, use the `klist` command as shown in this example.

Important: *The command must be typed on one line.*

```
klist -ke
WRFILE:/config/filestore/files_d/Common_d/kerberos_keytab_file_d/\:Common\:SUN-SPNEGO-APM106_key_file_2
```

The output for the example contains information like this.

```
Keytab name:
FILE:/config/filestore/files_d/Common_d/kerberos_keytab_file_d/:Common:SUN-SPNEGO-APM106_key_file_2
KVNO Principal
3    HTTP/apm106.labt.companynet.com@labt.companynet.com(arcfour-hmac)
```

Verify Kerberos delegation

From the command line, use the `kinit` command, as shown in this example.

```
kinit HTTP/apm106.labt.companynet.com@labt.companynet.com
```

You are prompted for a password and should receive a ticket (no output, no error).

Verify ticket

From the command line, type `klist`. Here is sample output: `/etc/krb5.conf`

Capture a TCP dump

Make sure the client sends the ticket to the BIG-IP® system; this verifies that the client setup is successful.

Chapter
10

NTLM Authentication for Microsoft Exchange Clients

- *Overview: Configuring APM for Exchange clients that use NTLM authentication*

Overview: Configuring APM for Exchange clients that use NTLM authentication

Access Policy Manager® (APM®) supports Microsoft Exchange clients that are configured to use NTLM, by checking NTLM outside of the APM session as needed. APM requires a machine account and an NTLM Auth configuration to perform these checks. APM requires an Exchange profile to support Microsoft Exchange clients, regardless of the authentication they are configured to use.

About using NTLM authentication

Microsoft software systems use NTLM as an integrated single sign-on (SSO) mechanism. However, in an Active Directory-based SSO scheme, Kerberos replaces NTLM as the default authentication protocol. NTLM is still used when a domain controller is not available or is unreachable, such as when the client is not Kerberos-capable, the server is not joined to a domain, or the user authenticates remotely over the web.

About configuration requirements for NTLM authentication

In Access Policy Manager®, you need to configure these elements:

- Machine account
- NTLM authentication configuration
- Kerberos SSO configuration
- Exchange profile that specifies the NTLM authentication configuration and specifies Kerberos SSO configurations for the specific Microsoft Exchange services supported
- Access profile that specifies the Exchange profile
- Access policy
- Pool of servers for the Exchange service to support Outlook Anywhere, supply a pool of Outlook Anywhere servers
- Virtual server that specifies the access profile and the pool

You also need to configure a special account in Active Directory for Kerberos constrained delegation (KDC).

About reusing a machine account for different BIG-IP systems

You can use the same machine account for two BIG-IP® systems when they are in an active-standby configuration. Otherwise, F5® recommends that you create a new NTLM machine account using the Access Policy Manager® user interface on each BIG-IP system.

Creating a new NTLM machine account on each BIG-IP system is helpful, for example, when two systems independently update their configurations without propagating them, or when you replicate the configuration into different BIG-IP systems using any configuration replication method. If you export a configuration and import it on another system, the machine account is included; however, after the import completes, you still need a new machine account and an NTLM authentication configuration that uses the new machine account on the target system.

About Outlook Anywhere and NTLM authentication

Access Policy Manager® supports Outlook Anywhere clients that are configured to use NTLM and HTTP Basic protocols independently. Typically, mobile devices use HTTP Basic authentication, while Outlook Anywhere clients can use both NTLM and HTTP Basic authentication. APM determines whether a client uses NTLM or HTTP Basic authentication and enforces the use of one or the other. After a client authenticates with NTLM or HTTP Basic, APM supports single sign-on with the back-end application or server using Kerberos constrained delegation (KCD).

Task summary for Exchange clients that use NTLM authentication

Task list

Configuring a machine account
Creating an NTLM Auth configuration
Setting up a delegation account to support Kerberos SSO
Creating a Kerberos SSO configuration in APM
Configuring an Exchange profile
Creating an access profile for Exchange clients
Configuring an access policy for NTLM authentication
Adding the access profile to the virtual server
Maintaining a machine account

Configuring a machine account

You need to configure a machine account so that Access Policy Manager® (APM®) can establish a secure channel to a domain controller.

1. On the Main tab, click **Access Policy** > **Access Profiles** > **NTLM** > **Machine Account**.
   A new Machine Account screen opens.
2. In the Configuration area, in the **Machine Account Name** field, type a name.
3. In the **Domain FQDN** field, type the fully qualified domain name (FQDN) for the domain that you want the machine account to join.
4. (Optional) In the **Domain Controller FQDN** field, type the FQDN for a domain controller.
5. In the **Admin User** field, type the name of a user who has administrator privilege.
6. In the **Admin Password** field, type the password for the admin user.
   APM uses these credentials to create the machine account on the domain controller. However, APM does not store the credentials and you do not need them to update an existing machine account configuration later.
7. Click **Join**.

This creates a machine account and joins it to the specified domain.

Creating an NTLM Auth configuration

Create an NTLM Auth configuration to specify the domain controllers that a machine account can use to log in.

1. On the Main tab, click **Access Policy** > **Access Profiles** > **NTLM** > **NTLM Auth Configuration**.
   A new NTLM Auth Configuration screen opens.
2. In the **Name** field, type a name.
3. From the **Machine Account Name** list, select the machine account configuration to which this NTLM Auth configuration applies.
   You can assign the same machine account to multiple NTLM authentication configurations.
4. For each domain controller, type a fully qualified domain name (FQDN) and click **Add**.

   Note: *You should add only domain controllers that belong to one domain.*

   By specifying more than one domain controller, you enable high availability. If the first domain controller on the list is not available, Access Policy Manager® tries the next domain controller on the list, successively.
5. Click **Finished**.

This specifies the domain controllers that a machine account can use to log in.

Setting up a delegation account to support Kerberos SSO

Before you can configure Kerberos SSO in Access Policy Manager®, you must create a delegation account in Active Directory. Note that for every server realm, you must create a delegation account in that realm.

1. Open the Active Directory Users and Computers administrative tool and create a new user account.
   The user account should be dedicated for delegation and the **Password never expires** setting enabled.
2. Run the `setspn` command-line tool for the user account from an elevated command prompt:
   The setspn command-line tool is available in Windows 2000 and Windows Server 2003 from Support Tools; it needs to be installed. The `setspn` tool is built into Windows Server 2008; it is available if you have the Active Directory Domain Services (AD DS) server role installed.
   a) Click the Windows Start button, right-click **Command Prompt**, and then click **Run as administrator**.
      An elevated command prompt opens.
   b) Type an `setspn` command.
      Example of command usage on Windows Server 2003: `setspn -A HTTP/bigcompany.lab.appnet.com domainname\userfordelegation`

      Note: *In Windows Server 2008, use the newly introduced* `-L` *option instead of* `-A`*. It validates the unicity of the service principal name (SPN) in the domain.*
3. Return to the Active Directory Users and Computers screen to open your account again.
   A Delegation tab should appear.
4. Click the Delegation tab.
5. Select **Trust this user for delegation to specified services only**.
6. Select **Use any authentication protocol**, and add all your services to the list under **Services to which this account can present delegated credentials**.
   Every service should have Service Type HTTP (or http) and host name of the pool member or web application resource host that you will use in your configuration.
7. Click **OK**.
   This creates the new delegation account.

Creating a Kerberos SSO configuration in APM

Before you create a Kerberos SSO configuration in Access Policy Manager®, create a delegation account in Active Directory.

To support Kerberos single sign-on authentication from APM®, you must create a Kerberos SSO configuration.

1. On the Main tab, click **Access Policy** > **SSO Configurations** > **Kerberos**.
   The SSO Configurations screen opens for Kerberos type.
2. Click **Create**.
   The New SSO Configuration screen opens.
3. In the **Name** field, type a name for the SSO configuration.
4. In the Credentials Source area, specify the credentials that you want cached for Single Sign-On.
5. In the **Kerberos Realm field,** type the name of the realm in uppercase.
   For example, `MY.HOST.LAB.MYNET.COM`
6. In the **Account Name** field, type the name of the Active Directory account configured for delegation.
7. In the **Account Password** and **Confirm Account Password** fields, type the delegation account password.
8. Click **Finished**.

Configuring an Exchange profile

If any of the Microsoft Exchange clients you support authenticate using NTLM, you must first create these objects:

- A machine account
- An NTLM Auth configuration
- At least one Kerberos SSO configuration

Note: *For Access Policy Manager® (APM®) to support Kerberos SSO, a delegation account is required on Active Directory.*

You create an Exchange profile to specify how to handle traffic from Microsoft Exchange clients.

1. On the Main tab, click **Access Policy** > **Application Access/Microsoft Exchange**.
   A list of Exchange profiles displays.
2. Click **Create**.
   A Create New Exchange Profile popup screen displays general settings.
3. In the **Exchange Name** field, type a name for the Exchange profile.
4. From the **Parent Profile** list, select a profile.
   The Exchange profile inherits settings from the parent profile that you select.

   Note: *APM supplies a default Exchange profile named exchange.*

   ---

5. Repeat these steps for one or more Microsoft Exchange services:
   a) From Service Settings on the left, select an Exchange service.
   Settings for the service are displayed in the right pane.
   b) In the **URL** field, retain any default settings that are displayed or type a path to use to match the Exchange client.

Default settings for this field are supplied in the default exchange profile.

c) From the **Front End Authentication** list, select the type of authentication to use: **Basic**, **Basic-NTLM**, or **NTLM**.

Only the applicable authentication types for the particular the Exchange service are included on the list.

Note: *If you select **NTLM** or **Basic-NTLM**, you must also select a configuration from **NTLM Configuration** list on the General Settings screen.*

d) From the **SSO Configuration** list, select an SSO configuration, if needed, for use after initial login.

For **Basic-NTLM**and **NTLM** authentication types, only Kerberos SSO is supported.

You configured settings for one or more Microsoft Exchange services.

6. Click **OK**.
The screen closes.

The Exchange profile is displayed on the list.

Apply this Exchange profile by adding it to an access profile.

Creating an access profile for Exchange clients

You create an access profile to provide the access policy configuration for a virtual server that establishes a secured session. You add an Exchange profile to the access policy to specify how to handle traffic from Microsoft Exchange clients.

1. On the Main tab, click **Access Policy** > **Access Profiles**.
The Access Profiles List screen opens.
2. Click **Create**.
The New Profile screen opens.
3. In the **Name** field, type a name for the access profile.
4. (Optional) In the Configurations area from the **Exchange** list, select an Exchange profile.

   Exchange profiles specify any SSO configurations for Microsoft Exchange services, such as Autodiscover, Outlook Anywhere, and so on. The configuration in the Exchange profile is used for Microsoft Exchange clients regardless of any SSO configuration you select from the **SSO Configuration** list in this access profile.
5. In the Language Settings area, add and remove accepted languages, and set the default language.

   A browser uses the highest priority accepted language. If no browser language matches the accepted languages list, the browser uses the default language.
6. Click **Finished**.

The access profile appears in the Access Profiles List.

Configuring an access policy for NTLM authentication

You configure an access policy for NTLM authentication to support Outlook Anywhere clients that log in using NTLM to also gain SSO access to a backend server that is protected by Kerberos KCD.

Note: *NTLM authentication occurs before an access policy runs. If NTLM authentication fails, an error displays and the access policy does not run.*

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. In the Access Policy column, click the **Edit** link for the access profile you want to configure.
   The visual policy editor opens the access policy in a separate screen.
3. Click the **(+)** icon anywhere in the access policy to add a new action item.
   A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
4. On the Endpoint Security (Server-Side) tab, select **Client for MS Exchange** and click **Add Item** to add the action to the access policy.

   A Client for MS Exchange action determines whether the client is using Microsoft Exchange or ActiveSync protocols. You must add this action before an NTLM Auth Result action.

   The Client for MS Exchange action popup screen opens.
5. Click **Save**.
   The properties screen closes and the visual policy editor displays.
6. Check whether the Outlook Anywhere client authenticated using NTLM.
   a) Click the **[+]** sign on the successful branch after the Client for MS Exchange action.
      An Add Item window opens.
   b) On the **Authentication** tab, select **NTLM Auth Result**.
   c) Click **Add Item**.
      A popup screen opens.
   d) Click **Save**.
      The properties screen closes and the visual policy editor is displayed.
7. Configure a branch in the access policy for an Outlook Anywhere client that has authenticated using NTLM.
   a) Click the **[+]** sign on the successful branch after the NTLM Auth Result action.
      An Add Item window opens.
   b) On the Assignment tab, select **SSO Credential Mapping** and click **Add Item**.
      The SSO Credential Mapping screen opens.
   c) Click **Save**.
      The properties screen closes and the visual policy editor is displayed.
   d) On the fallback branch after the SSO Credential Mapping action, click the **Deny** ending.
      A popup screen opens.
   e) Select **Allow** and click **Save**.
      You have completed a branch in the access policy for an Outlook Anywhere client that, having previously authenticated with NTLM, has SSO (Kerberos KCD) access on the back end.
8. Configure a branch in the access policy for an Outlook Anywhere client that uses HTTP Basic authentication.
   a) Click the **[+]** sign on the fallback branch after the NTLM Auth Result action.
      An Add Item window opens.
   b) On the Logon tab, select **Logon Page** and click the **Add Item** button.
      The Logon Page Agent properties screen opens.
   c) Make any changes that you require to logon page properties and click **Save**.
      The properties screen closes and the visual policy editor is displayed.
   d) On the Successful branch after the Logon Page action, add an authentication action.
   e) On the Successful branch after the authentication action, add an SSO Credential Mapping action.
   f) On the fallback branch after SSO Credential Mapping, change the ending from Deny to Allow.

   You have completed a branch in the access policy to authenticate an Outlook Anywhere client that uses HTTP Basic authentication and provides SSO (Kerberos KCD) access for the client on the back end.

9. (Optional) On the fallback branch after the MS Exchange Client action, configure a branch for a client that is not an Outlook Anywhere client.

   You could add Logon Page, authentication, and SSO Credential Mapping actions or other actions here.

10. Click the **Apply Access Policy** link to apply and activate the changes to the access policy.

You have created an access policy that checks whether the client is an Outlook Anywhere client and whether such a client has authenticated using NTLM. If so, the policy provides SSO (Kerberos KCD) access on the backend server.

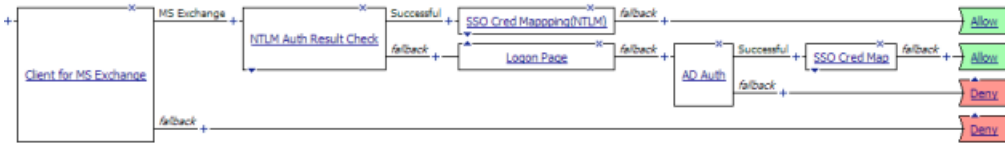


Figure 12: Example access policy with actions based on whether NTLM authentication occurred

Adding the access profile to the virtual server

You associate the access profile with the virtual server so that Access Policy Manager® can apply the profile to incoming traffic.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the name of the virtual server you want to modify.
3. In the Access Policy area, from the **Access Profile** list, select the access profile.
4. Click **Update** to save the changes.

Your access policy is now associated with the virtual server.

Maintaining a machine account

In some networks, administrators run scripts to find and delete outdated machine accounts on the domain controllers. To keep the machine account up-to-date, you can renew the password periodically.

1. On the Main tab, click **Access Policy** > **Access Profiles** > **NTLM** > **Machine Account**.
   The Machine Account screen opens.
2. Click the name of a machine account.
   The properties screen opens and displays the date and time of the last update to the machine account password.
3. Click the **Renew Machine Password** button.
   The screen refreshes and displays the updated date and time.

This changes the machine account last modified time.

Chapter
11

HTTP Basic Authentication for Microsoft Exchange Clients

- *Overview: Configuring APM for Exchange clients that use HTTP Basic*

Overview: Configuring APM for Exchange clients that use HTTP Basic

Access Policy Manager® (APM®) requires an Exchange profile to support Microsoft Exchange clients. An Exchange profile is specified in the access profile attached to the virtual server that handles the traffic from Exchange clients.

About Exchange profiles

An Exchange profile specifies service settings for Microsoft Exchange clients. Based on the settings, Access Policy Manager® (APM®) identifies the client, authenticates the client and, when an SSO configuration is specified, provides SSO.

In an Exchange profile, you can specify settings for one or more of these Microsoft Exchange services:

- ActiveSync
- Autodiscover
- Exchange Web Service
- Offline Address Book
- Outlook Anywhere

For Microsoft Exchange clients that are configured to use NTLM, you must include an NTLM authentication configuration in the Exchange profile.

Note: *With an NTLM authentication configuration, APM supports only Kerberos SSO on the back end.*

An Exchange profile is specified in an access profile.

Task summary for Exchange clients that use HTTP Basic authentication

Task list

Configuring an Exchange profile
Creating an access profile for Exchange clients
Configuring an access policy for Microsoft Exchange clients
Adding the access profile to the virtual server

Configuring an Exchange profile

If any of the Microsoft Exchange clients you support authenticate using NTLM, you must first create these objects:

- A machine account
- An NTLM Auth configuration
- At least one Kerberos SSO configuration

Note: *For Access Policy Manager® (APM®) to support Kerberos SSO, a delegation account is required on Active Directory.*

You create an Exchange profile to specify how to handle traffic from Microsoft Exchange clients.

1. On the Main tab, click **Access Policy** > **Application Access/Microsoft Exchange**.
   A list of Exchange profiles displays.
2. Click **Create**.
   A Create New Exchange Profile popup screen displays general settings.
3. In the **Exchange Name** field, type a name for the Exchange profile.
4. From the **Parent Profile** list, select a profile.

   The Exchange profile inherits settings from the parent profile that you select.

   Note: *APM supplies a default Exchange profile named exchange.*

5. Repeat these steps for one or more Microsoft Exchange services:
   a) From Service Settings on the left, select an Exchange service.
   Settings for the service are displayed in the right pane.
   b) In the **URL** field, retain any default settings that are displayed or type a path to use to match the Exchange client.

   Default settings for this field are supplied in the default exchange profile.
   c) From the **Front End Authentication** list, select the type of authentication to use: **Basic**, **Basic-NTLM**, or **NTLM**.

   Only the applicable authentication types for the particular the Exchange service are included on the list.

   Note: *If you select **NTLM** or **Basic-NTLM**, you must also select a configuration from **NTLM Configuration** list on the General Settings screen.*

   d) From the **SSO Configuration** list, select an SSO configuration, if needed, for use after initial login.
   For **Basic-NTLM**and **NTLM** authentication types, only Kerberos SSO is supported.

   You configured settings for one or more Microsoft Exchange services.
6. Click **OK**.
   The screen closes.

The Exchange profile is displayed on the list.

Apply this Exchange profile by adding it to an access profile.

Creating an access profile for Exchange clients

You create an access profile to provide the access policy configuration for a virtual server that establishes a secured session. You add an Exchange profile to the access policy to specify how to handle traffic from Microsoft Exchange clients.

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. Click **Create**.
   The New Profile screen opens.
3. In the **Name** field, type a name for the access profile.
4. (Optional) In the Configurations area from the **Exchange** list, select an Exchange profile.

   Exchange profiles specify any SSO configurations for Microsoft Exchange services, such as Autodiscover, Outlook Anywhere, and so on. The configuration in the Exchange profile is used for Microsoft Exchange clients regardless of any SSO configuration you select from the **SSO Configuration** list in this access profile.

5. In the Language Settings area, add and remove accepted languages, and set the default language.

   A browser uses the highest priority accepted language. If no browser language matches the accepted languages list, the browser uses the default language.

6. Click **Finished**.

The access profile appears in the Access Profiles List.

Configuring an access policy for Microsoft Exchange clients

Before you configure this access policy, you must have an AAA Active Directory server configured in Access Policy Manager®.

You configure an access policy to support Microsoft Exchange clients with login, HTTP basic authentication, and SSO.

Note: *This access policy does not support Microsoft Exchange clients that are configured to authenticate using NTLM.*

---

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. In the Access Policy column, click the **Edit** link for the access profile you want to configure.
   The visual policy editor opens the access policy in a separate screen.
3. On an access policy branch, click the **(+)** icon to add an item to the access policy.
   A popup screen displays actions on tabs, such as General Purpose and Authentication, and provides a search field.
4. On the Logon tab, select **Logon Page** and click the **Add Item** button.
   The Logon Page Agent properties screen opens.
5. Make any changes that you require to the properties and click **Save**.
   The properties screen closes and the visual policy editor displays.
6. On the fallback branch after the previous action, click the **(+)** icon to add an item to the access policy.
   A popup screen opens.
7. On the Authentication tab, select **AD Auth**.
   A properties screen displays.
8. From the **Server** list, select a server.
9. Click **Save**.
   The properties screen closes and the visual policy editor displays.
10. On the Successful branch after the previous action, click the **(+)** icon.
    A popup screen opens.
11. On the Assignment tab, select **SSO Credential Mapping** and click **Add Item**.
    A properties screen opens.
12. Click **Save**.
    The properties screen closes and the visual policy editor displays.
13. Click the **Apply Access Policy** link to apply and activate the changes to the access policy.

Adding the access profile to the virtual server

You associate the access profile with the virtual server so that Access Policy Manager® can apply the profile to incoming traffic.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.

The Virtual Server List screen opens.

2. Click the name of the virtual server you want to modify.
3. In the Access Policy area, from the **Access Profile** list, select the access profile.
4. Click **Update** to save the changes.

Your access policy is now associated with the virtual server.

Chapter 12

HTTP and HTTPS Authentication

- *About HTTP AAA server authentication*
- *Task summary for HTTP authentication*
- *Overview: Configuring HTTPS authentication*

About HTTP AAA server authentication

An HTTP AAA server directs users to an external web-based server to validate credentials. Access Policy Manager® (APM®) supports these HTTP authentication types:

- HTTP basic authentication - Directs users to a URI
- HTTP NTLM authentication - Directs users to a URI
- HTTP form-based authentication - Directs users to a form action URL and provides the specified form parameters
- HTTP custom post - Directs users to a POST URL, a submit URL, or a relative URL and provides the specified content

Tip: *Use HTTPS instead of HTTP authentication for improved security, because HTTP authentication passes user credentials as clear text.*

Task summary for HTTP authentication

To set up this configuration, you must first configure one HTTP AAA server that supports the type of authentication that you want: HTTP Basic/NTLM, form-based, or custom post. After you configure an HTTP AAA server, you must add an HTTP Auth action to an access policy and specify the HTTP AAA server that supports the authentication type that you want to use.

Task list

Configuring an AAA server for HTTP Basic/NTLM authentication
Configuring an HTTP AAA server for form-based authentication
Configuring an HTTP AAA server for custom post authentication
Creating an access profile
Using HTTP authentication in an access policy
Creating a virtual server

Configuring an AAA server for HTTP Basic/NTLM authentication

You configure an HTTP AAA server when you want to use Basic/NTLM authentication.

1. On the Main tab, click **Access Policy** > **AAA Servers** > **HTTP**.
   The HTTP Servers screen displays.
2. Click **Create**.
   The New Server properties screen opens.
3. In the **Name** field, type a unique name for the authentication server.
4. For **Authentication Type**, select `Basic/NTLM`.
5. In the **Start URI** field, type the complete URI that returns the logon form.
   The URI resource must respond with a challenge to a non-authenticated request.
6. Click **Finished**.
   The new server displays on the list.

Configuring an HTTP AAA server for form-based authentication

You create a form-based HTTP AAA configuration to use HTTP form-based authentication from an access policy.

1. On the Main tab, click **Access Policy** > **AAA Servers** > **HTTP**.
   The HTTP Servers screen displays.
2. Click **Create**.
   The New Profile screen opens.
3. In the **Name** field, type a unique name for the authentication server.
4. For **Authentication Type**, select **Form Based**.
5. (Optional) In the **Start URI** field, type a URL resource, for example, `http://plum.tree.lab2.sp.companynet.com/`.

   This resource must respond with a challenge to a non-authenticated request.

   Note: *Typing a URL resource is optional, because the form action field specifies either an absolute URL or a relative URL resource. However, if you choose to specify both the* ***Start URI*** *and* ***Form Action****, then Access Policy Manager® uses both start URI and form action parameters as the final URL for HTTP POST. If you do not specify a start URI, Access Policy Manager is likely to detect that the absolute URI based on the form action parameter should be used for HTTP POST.*

   ---

6. From the **Form Method** list, select either **GET** or **POST**.

   If you specify **GET** , the authentication request converts as HTTP GET.
7. In the **Form Action** field, type the complete destination URL to process the form.

   This is used to specify the form action URL which is used for doing HTTP form-based authentication. This is required. If you do not specify a form action, then Access Policy Manager uses the URI from the request to perform HTTP form-based authentication.
8. In the **Form Parameter For User Name** and **Form Parameter For Password** fields, type the parameter name and password used by the form to which you are sending the POST request.
9. In the **Hidden Form Parameters/Values** field, type the hidden form parameters required by the authentication server logon form at your location.

   You must provide hidden form parameters and values if there are any. When present, these values are required by the authentication server logon form at your location.
10. In the **Number Of Redirects To Follow** field, type how far from the landing page, in pages, the request should travel before failing.
11. For the **Successful Logon Detection Match Type** setting, select the method your authenticating server uses, and type the option definition in the **Successful Logon Detection Match Value** field.
12. Click **Finished**.
    The new server displays on the list.

Configuring an HTTP AAA server for custom post authentication

You create a custom post configuration when there is no form and when body encoding is different from form encoding. (This can happen when POST is generated by JavaScript or ActiveX.) Using a custom post, you can specify the entire post body and any non-default HTTP headers.

1. On the Main tab, click **Access Policy** > **AAA Servers** > **HTTP**.
   The HTTP Servers screen displays.

2. Click **Create**.
   The New Profile screen opens.
3. In the **Name** field, type a unique name for the authentication server.
4. For the **Authentication Type** setting, select **Custom Post**.
5. In the **Start URI** field, type in a URL resource, for example, `http://plum.tree.lab2.sp.companynet.com/`.

   If you do not specify a Start URI, Access Policy Manager will likely detect that the absolute URI based on the Form Action parameter should be used for HTTP POST. If you specify a Start URI, Access Policy Manager® uses both the Start URI and the Form Action parameters as the final URL for HTTP POST.
6. In the **Form Action** field, type the POST URL, the submit URL, or a relative URL.
7. For the **Successful Logon Detection Match Type** setting, select the method that the authenticating server uses.
8. For the **Successful Logon Detection Match Value**, type a value depending on the **Successful Logon Detection Match Type** that you selected:
   - **By Resulting Direct URL** - Specify a URL if you selected this type.
   - **By Presence of Specific String in Cookie** - Specify a single string if you selected this type.

     Note: *With this option, when APM® receives a duplicate cookie, it adds it to the existing cookie list. As a result, multiple cookies with the same name, domain, and path can exist and can be searched.*
   - **By Presence of Cookie That Exactly Matches** - Specify the exact key fields (name, path, and domain) that are present in the HTTP response cookie if you select this type. Failure to supply the exact number of keys and the exact values for the HTTP response cookie results in a `No matching cookie found` error.

     Note: *This option supports cookie merge functionality. When APM receives a cookie that has the same name, domain, and path as an existing cookie, it merges it into the existing cookie.*
   - **By Specific String in Response** - Specify a string if you select this option.
9. In the **Number Of Redirects To Follow** field, type how far from the landing page, in pages, the request should travel before failing.
10. From the **Content Type** list, select an encoding for the HTTP custom post.

    The default setting is **XML UTF-8**.

    Note: *If you select **None**, you must add a header in the **Custom Headers** setting and you must apply your own encoding through an iRule.*
11. In the **Custom Body** field, specify the body for the HTTP custom post.
12. For **Custom Headers**, specify names and values for header content to insert in the HTTP custom post.
13. Click **Finished**.
    The new server displays on the list.

This creates an HTTP AAA server that provides a custom post for authentication.

To put this authentication into effect, add this AAA server to an HTTP Auth action in an access policy.

Creating an access profile

You create an access profile to provide the access policy configuration for a virtual server that establishes a secured session.

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. Click **Create**.
   The New Profile screen opens.
3. Type a name for the access profile.
4. From the **Profile Type** list, select one:
   - **APM-LTM** - Select for a web access management configuration.
   - **SSO** - Select only when you do not need to configure an access policy.
   - **SWG - Explicit** - Select to configure access using Secure Web Gateway explicit forward proxy.
   - **SWG - Transparent** - Select to configure access using Secure Web Gateway transparent forward proxy.
   - **SSL-VPN** - Select for other types of access, such as network access, portal access, application access. (Most access policy items are available for this type.)
   - **ALL** - Select for any type of access.

   Additional settings display.
5. In the Language Settings area, add and remove accepted languages, and set the default language.

   A browser uses the highest priority accepted language. If no browser language matches the accepted languages list, the browser uses the default language.
6. Click **Finished**.

This creates an access profile with a default access policy.

Using HTTP authentication in an access policy

Before you can set up an access policy to use HTTP authentication, you must have at least one HTTP AAA server configured.

You configure an access policy with an HTTP Auth action when you want users to authenticate using one of the HTTP authentication types that Access Policy Manager® (APM®) supports: Basic, NTLM, form-based, or custom.

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. In the Access Policy column, click the **Edit** link for the access profile you want to configure.
   The visual policy editor opens the access policy in a separate screen.
3. Click the **(+)** icon anywhere in the access policy to add a new action item.
   A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
4. On the Logon tab, select **Logon Page** and click the **Add Item** button.
   The Logon Page Agent properties screen opens.
5. Make any changes that you require to the logon page properties and click **Save**.
   The properties screen closes and the visual policy editor displays.
6. Click the **(+)** icon anywhere in the access policy to add a new action item.
   A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
7. On the Authentication tab, select **HTTP Auth** and click **Add item**.
   A properties popup screen opens.
8. From the **AAA Server** list, select the AAA HTTP server you want to use for authentication.
9. (Optional) Add any other branches and actions that you need to complete the access policy.

10. Click **Save**.
    The properties screen closes and the visual policy editor displays.
11. Click **Apply Access Policy** to save your configuration.

This adds an HTTP AAA authentication server to the access policy.

To put the access policy into effect, add it to a virtual server.

Creating a virtual server

When creating a virtual server for an access policy, specify that the virtual server is a host virtual server, and not a network virtual server.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, select **Host** and in the **Address** field, type the IP address for the virtual server.
5. In the **Service Port** field, type a port number or select a service name from the **Service Port** list.
6. From the **HTTP Profile** list, select **http**.
7. If you use server SSL for this connection, from the **SSL Profile (Server)** list, select a server SSL profile.
8. If you use client SSL for this profile, from the **SSL Profile (Client)** list, select a client SSL profile.
9. In the Access Policy area, from the **Access Profile** list, select the access profile.
10. From the **Connectivity Profile** list, select a connectivity profile.
    You can select the default connectivity profile, **connectivity** if you have not defined a specific profile for the traffic that is directed to this virtual server.
11. Click **Finished**.

You have configured a host virtual server and associated an access profile with it.

Overview: Configuring HTTPS authentication

You can configure HTTP AAA authentication to use server-side SSL (HTTPS). To set up this configuration, you must first configure one HTTP AAA server that supports the type of authentication that you want to use: HTTP Basic/NTLM, form-based, or custom post.

HTTP AAA server configuration notes

Configure the HTTP AAA server so that in the **Start URI** or **Form Action** field you use:

- The http scheme (not https)
- The host name of the external HTTP server (rather than the IP address)

For example: `http://plumtree.lab2.sp.companynet.com`.

Virtual server configuration notes

Configure the virtual server to use the host name of the external HTTP server; this is the same host name as used in the HTTP AAA server configuration.

Important: *Set the* ***Destination*** *field to use the host name of the external HTTP server. For example:* `companynet.com` *(and set the* ***Service Port*** *to HTTP).*

To ensure that SSL is used between the HTTP AAA server and the external HTTP server, the virtual server configuration includes a server SSL profile and a pool with a member that uses SSL.

DNS configuration notes

The DNS configuration on the BIG-IP® system must send traffic to the virtual server instead of the external HTTP server.

Note: *This implementation does not explain how to configure DNS.*

Task summary

Before you start these tasks, configure an HTTP AAA server.

Creating a pool for HTTPS authentication
Creating a virtual server for HTTPS authentication
Creating an access profile
Using HTTP authentication in an access policy
Adding the access profile to the virtual server

Creating a pool for HTTPS authentication

You create a pool (HTTPS) so that you can assign it to a virtual server (HTTP) that accepts HTTP traffic and provides server-side SSL using this pool.

1. On the Main tab, click **Local Traffic** > **Pools**.
   The Pool List screen opens.
2. Click **Create**.
   The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. Scroll down to the Resources area.
5. In the **New Members Address** field, type an IP address.
6. From the **Service Port** list, select `HTTPS`.
7. Click **Add**.
8. Click **Finished**.

Creating a virtual server for HTTPS authentication

You create a virtual server that accepts HTTP traffic, encrypts it (using a server SSL profile), and passes it to an HTTPS server to provide secure communication between the BIG-IP® system and an external HTTP authentication server.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.

The Virtual Server List screen opens.

2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. In the **Destination Address** field, type the host name of the external HTTP server.
5. From the **Service Port** list, select **HTTP**.
6. From the **SSL Profile (Server)** list, select a profile.

   This ensures that there is an SSL connection between the HTTP virtual server and the external HTTPS server.
7. From the **VLAN and Tunnel Traffic** list, select **Enabled on...**
8. From the **Source Address Translation** list, select **Auto Map**.
9. Scroll all the way down to the Resources area and from the **Default Pool** list, select the pool you configured previously.

   The pool must contain a member configured for HTTPS.
10. Click **Finished**.

Creating an access profile

You create an access profile to provide the access policy configuration for a virtual server that establishes a secured session.

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. Click **Create**.
   The New Profile screen opens.
3. Type a name for the access profile.
4. From the **Profile Type** list, select one:

   - **APM-LTM** - Select for a web access management configuration.
   - **SSO** - Select only when you do not need to configure an access policy.
   - **SWG - Explicit** - Select to configure access using Secure Web Gateway explicit forward proxy.
   - **SWG - Transparent** - Select to configure access using Secure Web Gateway transparent forward proxy.
   - **SSL-VPN** - Select for other types of access, such as network access, portal access, application access. (Most access policy items are available for this type.)
   - **ALL** - Select for any type of access.

   Additional settings display.
5. In the Language Settings area, add and remove accepted languages, and set the default language.

   A browser uses the highest priority accepted language. If no browser language matches the accepted languages list, the browser uses the default language.
6. Click **Finished**.

This creates an access profile with a default access policy.

Using HTTP authentication in an access policy

Before you can set up an access policy to use HTTP authentication, you must have at least one HTTP AAA server configured.

You configure an access policy with an HTTP Auth action when you want users to authenticate using one of the HTTP authentication types that Access Policy Manager® (APM®) supports: Basic, NTLM, form-based, or custom.

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. In the Access Policy column, click the **Edit** link for the access profile you want to configure.
   The visual policy editor opens the access policy in a separate screen.
3. Click the **(+)** icon anywhere in the access policy to add a new action item.
   A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
4. On the Logon tab, select **Logon Page** and click the **Add Item** button.
   The Logon Page Agent properties screen opens.
5. Make any changes that you require to the logon page properties and click **Save**.
   The properties screen closes and the visual policy editor displays.
6. Click the **(+)** icon anywhere in the access policy to add a new action item.
   A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
7. On the Authentication tab, select **HTTP Auth** and click **Add item**.
   A properties popup screen opens.
8. From the **AAA Server** list, select the AAA HTTP server you want to use for authentication.
9. (Optional) Add any other branches and actions that you need to complete the access policy.
10. Click **Save**.
    The properties screen closes and the visual policy editor displays.
11. Click **Apply Access Policy** to save your configuration.

This adds an HTTP AAA authentication server to the access policy.

To put the access policy into effect, add it to a virtual server.

Adding the access profile to the virtual server

You associate the access profile with the virtual server so that Access Policy Manager® can apply the profile to incoming traffic.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the name of the virtual server you want to modify.
3. In the Access Policy area, from the **Access Profile** list, select the access profile.
4. Click **Update** to save the changes.

Your access policy is now associated with the virtual server.

Chapter 13

Local User Database

- *Overview: Configuring and administering a local user database*
- *Overview: Using a local user database to control authentication*
- *Overview: Branching in an access policy based on local user database groups*

Overview: Configuring and administering a local user database

You can create multiple local user databases to provide on-box authentication, to control user access, to segment your users, and to store user information.

During access policy operation, you can read from and write to a local user database.

- You can read from a local user database to:
  - Determine whether a user is locked out of a local user database instance.
  - Check the number of failed login attempts for a user.
  - Check group membership for the user to determine which access policy branch to take.

  Note: *Groups are text strings. You create them from the Configuration utility.*

- You can write to a local user database primarily to increment or reset the number of login failures for a user. You can also update the locked out status for the user; although this option provides flexibility, use it sparingly. Normally, locked out status is set programmatically.

About backing up and restoring users

You can export user data from a local user database instance to a comma-separated values (CSV) file. The purpose is to provide you with a way to back up user data, which you can import from the CSV file.

Note: *Dynamically created users are not included in the CSV file that you export to back up user data. (You can configure the* ***Local Database*** *action to dynamically create users from an access policy.)*

About local user database synchronization across devices

When BIG-IP® systems are included in a Sync-only device group, configuration data is synchronized automatically or manually using ConfigSync. ConfigSync has no effect on local user databases, however. In a Sync-only device group, the active node provides the local user database data to the other nodes initially, and then provides updated data every five minutes.

Synchronization status

The date and time of the last high availability (HA) synchronization displays in the Configuration utility on the Local User List and Local Database Instance screens. If errors occur, they are logged and then available in Access Policy Manager® reports.

About writing to a local user database from an access policy

Administrators using the Configuration utility to update user data need to know that user data can also be updated by access policies. When you include a Local Database action in an access policy and configure a write action in it, it can make these changes in a local user database instance.

Updates to local user databases in the Common partition

You can use the Local Database action to write to local user databases in the Common partition, in addition to the partition in which you create the access policy.

Note: *Usually, you have read-only access to the Common partition from any other partition. The Local Database write action is an exception to this general rule.*

Changes to individual user data

The Local Database action can change these types of information for a user:

- Number of login failures
- Locked out status - If you change the locked out status for a user, you could change the result of other access policy actions, such as a local database read action, or a local database authentication action.

Note: *When you create a Local Database write action, although groups is on the list of destination DB properties, you cannot write to it. The groups that are entered using the Configuration utility are preserved.*

Addition and deletion of dynamic users

The Allow User Creation option of the Local Database action creates a user when one does not already exist in a local user database. You can use this option to track login attempts by users who are not found in the local user database and to lock them out for a period of time. Dynamically created users are removed from the database after 20 minutes.

Task summary

Configuring a local user database instance
Adding a user to a local user database instance
Forcing change of password for a local user database instance

Configuring a local user database instance

Configure a local user database instance so you can add users and user data to it.

1. On the Main tab, select **Access Policy** > **Local User DB** > **Manage Instances**.
   The Manage Instances screen displays.
2. Click **Create New Instance**.
   The Create New Local User DB Instance popup screen displays.
3. In the **Name** field, type a unique name for the database instance.
4. In the **Lockout Interval (in seconds)** field, type the number of seconds to keep a user account locked.
   The default setting is 600 seconds.
5. In the **Lockout Threshold** field, type the maximum number of login failures to allow.
   The default setting is 3.
6. Click **OK**.

You have created a local user database instance.

Adding a user to a local user database instance

Before you start this procedure, a local user database instance must already exist.

Add a user to a local user database instance for authentication or for determining a branching strategy in an access policy that is based on user group membership.

Note: *The data in a local user database is not validated against external sources.*

---

1. On the Main tab, select **Access Policy** > **Local User DB** > **Manage Users**.
   The Manage Users screen displays.
2. Click **Create New User**.
   The Create New Local User screen opens and displays User Information settings.
3. In the **User Name** field, type the user name.
4. In the **Password** and **Confirm Password** fields, type the user's password.
5. Select the **Force Password Change** check box to force the user to change password the next time they log in.

   After the user successfully changes password, this check box is cleared. You can select this check box at any time to force the user to change password at their next log in.
6. From the **Instance** list, select a local user database instance.
   You have completed the mandatory settings.
7. (Optional) Select **Personal Information** to specify **First Name**, **Last Name**, and **Email**.

   This information is not accessible to an access policy.
8. (Optional) Select **User Groups** to specify **Group Memberships** for the user.

   A group membership is a text string.

   A **Local Database** action can read groups from the database to determine branching strategy in an access policy.
9. Click **OK**.

You have created a local user in a local user database instance.

To add this user to another local user database instance, repeat this procedure and select the other instance.

Forcing change of password for a local user database instance

You can force a user to change password for the local user database instance when you need to do so.

1. On the Main tab, select **Access Policy** > **Local User DB** > **Manage Users**.
   The Manage Users screen displays.
2. Select a user and click **Edit**.
   The Edit Local User screen opens and displays **User Information** settings.
3. Select the **Force Password Change** check box.

   You can select this check box at any time to force the user to change a password at their next log in.
4. Click **OK**.
   The screen closes.

The user is prompted to change a password the next time they log in. After the user successfully changes a password, the **Force Password Change** check box is cleared.

Overview: Using a local user database to control authentication

You can authenticate users directly against a local user database using the Local DB Auth action from an access policy.

Also, you can use the local user database to track login failures and lock a user out for a period of time. (This is possible whether authentication occurs against an external AAA server, or a local user database.) Furthermore, you can use the local database to count login failures by users that are not found in a user database, and locking them out for a period of time. You can accomplish these tasks by using Local Database actions to read and write information from an access policy.

About locking a user out of an AAA server using a local user database

A macro, AD auth and LocalDB lockout, is available in the visual policy editor that provides a good example of using the Local Database action to lock users out of an external AAA server.

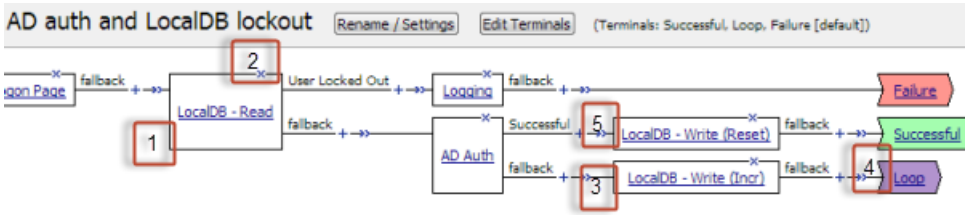


Figure 13: AD auth and LocalDB lockout macro

1. A local database action (LocalDB - Read) reads the **locked_out database** property and determines whether the user is locked out.
2. Exiting the LocalDB - Read action on the User Locked Out branch leads to a logging action and a Failure terminal.
3. If the login (AD Auth) fails, a local database write action (LocalDB - Write (Incr)) increments the number of login failures by 1.

Note: *To keep actual logon attempts aligned with the number recorded by the LocalDB - Write (Incr) action (incrementing by 1), the* ***Max Logon Attempts Allowed*** *property in the AD Auth action is set to 1. If it was set to another number, for example 2, you would need to configure the LocalDB - Write (Incr) action to increment login failures by the same number, 2.*

---

4. A Loop terminal in the macro causes the macro to loop through the AD Auth and LocalDB - Write (Incr) actions until authentication succeeds or until the maximum number of logon attempts is surpassed and the macro exits through the Loop terminal.
5. If the login, AD Auth, succeeds, a Local Database write action, LocalDB - Write (Reset), resets the user's login failures to 0 (zero).

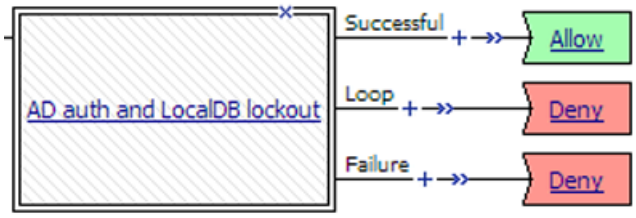


Figure 14: AD auth and LocalDB lockout macrocall in an access policy

In an access policy, three branches follow the macro:

- Successful: The user logged in successfully.
- Loop: The user failed to log in the maximum number of times and is locked out now.
- Failure: The user is locked out and does not get another chance to try to log in.

About writing to a local user database from an access policy

Administrators using the Configuration utility to update user data need to know that user data can also be updated by access policies. When you include a Local Database action in an access policy and configure a write action in it, it can make these changes in a local user database instance.

Updates to local user databases in the Common partition

You can use the Local Database action to write to local user databases in the Common partition, in addition to the partition in which you create the access policy.

Note: *Usually, you have read-only access to the Common partition from any other partition. The Local Database write action is an exception to this general rule.*

Changes to individual user data

The Local Database action can change these types of information for a user:

- Number of login failures
- Locked out status - If you change the locked out status for a user, you could change the result of other access policy actions, such as a local database read action, or a local database authentication action.

Note: *When you create a Local Database write action, although groups is on the list of destination DB properties, you cannot write to it. The groups that are entered using the Configuration utility are preserved.*

Addition and deletion of dynamic users

The Allow User Creation option of the Local Database action creates a user when one does not already exist in a local user database. You can use this option to track login attempts by users who are not found in the local user database and to lock them out for a period of time. Dynamically created users are removed from the database after 20 minutes.

Task summary

Authenticating users and locking them out with a local database
Unlocking a user who is locked out of a local user database instance

Authenticating users and locking them out with a local database

Before you start this task: create a local user database instance in Access Policy Manager®, add users to the local database instance, and then create an access profile.

Authenticate a user against a local user database when an external AAA server is not available. Read and write to a local user database when you want to track failed login attempts and lock out users that repeatedly attempt and fail to log in.

Note: *For enhanced security, F5® recommends that you place Local Database actions before and after a LocalDB Auth action to read and write user information. This enables you to track and block login attempts by any user. This process is demonstrated in the example access policy described here. (You can use this same process to lock users out of an AAA server by substituting another authentication action for the LocalDB Auth action.)*

---

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. In the Access Policy column, click the **Edit** link for the access profile you want to configure.
   The visual policy editor opens the access policy in a separate screen.
3. On an access policy branch, click the **(+)** icon to add an item to the access policy.

   Repeat this action from the visual policy editor whenever you want to add an item to the access policy.

   A popup screen displays actions on tabs, such as General Purpose and Authentication, and provides a search field.
4. On the selection screen, type `logon` in the search field, select **Logon Page** from the results, and select **Add Item**.

   A logon page should precede other actions in the access policy.

   Note: *A locked out user is not presented with a logon page, regardless of how many authentication attempts are allowed.*

   ---

5. Make any changes that you require to the logon page properties and click **Save**.
   The properties screen closes and the visual policy editor displays.
6. On the fallback branch after the previous action, click the **(+)** icon to add an item to the access policy.
   A popup screen opens.
7. Add a Local Database action and configure it to read the user's locked out status, and create a branch for a locked out user.
   a) Type `local` in the search field.

      Search is not case-sensitive.

      A list of matching actions displays.
   b) Select **Local Database** and click **Add Item**.
      A properties screen opens.
   c) From the **LocalDB Instance** list, select a local user database.
   d) In the **User Name** field, retain the default session variable or type another variable name or a user name.
   e) Click **Add new entry**
      A new line is added to the list of entries with the Action set to **Read**, and with a **Session Variable** field and default value in the Destination column.
   f) In the Destination column **Session Variable** field, type `session.localdb.locked_out` (or type the name of another variable).
   g) In the Source column from the **DB Property** list, select **locked_out**.
      The entry is complete. A read action reads a value from the database into a session variable.

h) Click the Branch Rules tab.
i) Click the **Add Branch Rule** button.
New **Name** and **Expression** settings are displayed.
j) In the **Name** field, replace the default name by typing a new name.
The default name is Branch Rule *n* where *n* is a number. The name appears on the branch in the access policy, so it should be descriptive.
k) Click the **change** link next to the Expression setting.
A popup screen opens.
l) Click the **Add Expression** button.
Settings are displayed.
m) From the **Agent Sel.** list select **LocalDB**.
The **Condition** list displays **LocalDB Auth Passed**. The **LocalDB Auth Passed** list displays **Passed**.
n) From the **LocalDB Auth Passed** list, select **Locked User Out**.
The branch rule is complete.
o) Click **Finished**.
The popup screen closes.
p) Click **Save**.
The properties screen closes and the visual policy editor is displayed.

8. On the fallback branch after the previous action , add a **LocalDB Auth** action and configure properties for it.

Valid values for the properties, **LocalDB Instance** and **Max Logon Attempts Allowed**, are available from lists.

Note: *A user that accumulates the maximum number of logon failures specified in the* ***LocalDB Auth*** *action is locked out of the local user database instance and exits the action on a Locked User Out branch. A user that is not found in the local user database exits the action to the fallback branch.*

---

9. On the fallback branch after the previous action, click the **(+)** icon to add an item to the access policy.
A popup screen opens.

10. Add a Local Database action to to increment and write login failures. Configure it to allow user creation.
a) Set the **LocalDB Instance** and **User Name** fields to the same values you selected previously.
b) From the **Allow User Creation** list, select **Yes**.
At each subsequent login attempt, login failures increase until the user is eventually locked out.
c) Add a new entry and configure it to read the login_failures DB property into a session variable.
d) Add a new entry and configure it to write the value of an expression that increments the number of failures into the login_failures DB property.

Here is an example of the expression:

```
expr { [mcget {session.localdb.login_failures}] + 1 }.
```

11. On the successful branch after the Local DB Auth action, add a **Local Database** action and configure it to reset login failures to 0 (zero).
a) Set the **LocalDB Instance** and **User Name** fields to the same values you selected previously.
b) Add a new entry and configure it to write the value of an expression that evaluates to zero into the login_failures DB property.

Here is an example of the expression:

```
expr { "0" }.
```

12. Click **Save**.
The properties screen closes and the visual policy editor displays.

13. Click the **Apply Access Policy** link to apply and activate the changes to the access policy.

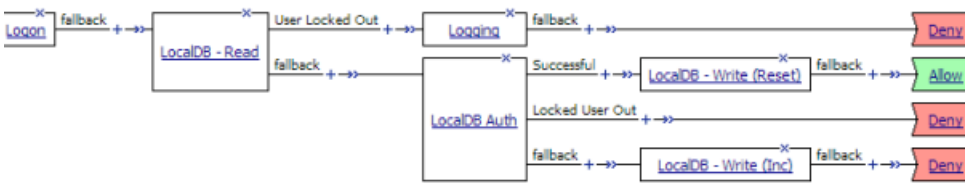


Figure 15: Sample access policy (with logging action on the User Locked Out branch)

You must add the access profile to a virtual server in order for this access policy to take effect.

Unlocking a user who is locked out of a local user database instance

You can unlock a user who is locked out of a local user database instance if you do not want to wait the for lockout interval to elapse.

Note: *The lockout interval is configurable and can be different for each local user database instance.*

1. On the Main tab, select **Access Policy** > **Local User DB** > **Manage Users**.
   The Manage Users screen displays.
2. Select a user for whom the Locked Out column specifies **yes**.
3. Click **Unlock User**.

The account is unlocked in the local user database instance.

Overview: Branching in an access policy based on local user database groups

You can store user group membership strings in a local user database instance. You can add one or more strings for a user to the database. The strings can reflect any grouping strategy that you want to apply. You can make user group-based branching decisions in an access policy by reading the group information for the user from the database, and creating rules for branching based on it.

Before you can perform this task, you need users and user group membership strings configured in a local user database instance. You also need an access profile.

Task summary

Creating an access policy to branch based on local DB group membership

Creating an access policy to branch based on local DB group membership

You can use an access policy to retrieve user group membership from a local user database instance and configure branch rules to provide different actions for users in different groups.

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. In the Access Policy column, click the **Edit** link for the access profile you want to configure.

The visual policy editor opens the access policy in a separate screen.

3. On an access policy branch, click the **(+)** icon to add an item to the access policy.

   Repeat this action from the visual policy editor whenever you want to add an item to the access policy.

   A popup screen displays actions on tabs, such as General Purpose and Authentication, and provides a search field.

4. Type `local` in the search field.

   Search is not case-sensitive.

   A list of matching actions is displayed.

5. Select **Local Database** and click **Add Item**.
   A properties screen displays.

6. From the **LocalDB Instance** list, select a local user database.

7. In the **User Name** field, retain the default session variable or type another variable name or a user name.

8. From the **Allow User Creation** list, retain the default value (**No**).

9. Click **Add new entry**.
   A new line is added to the list of entries.

10. Configure the entry to read the groups from the database and store them in a variable:
    a) From the Action list, select **Read**.
    b) In the **Source** column from the **DB Property** list, select **groups**.
    c) In the **Destination** column **Session Variable** field, retain the default value, session.localdb.groups or type the name of a variable.
    d) In the **Source** column from the **DB Property** list, select **groups**.
    You have configured an action that reads the user's groups into a variable.

11. Click the Branch Rules tab to edit a branch rule.

12. Click the **Add Branch Rule** button.
    New **Name** and **Expression** settings display.

13. In the **Name** field, replace the default name by typing a new name over it.

    The default name is Branch Rule *n* where *n* is a number. The name appears on the branch in the access policy and so should be descriptive.

14. Click the **change** link in the Expression section.
    A popup screen opens.

15. Click the **Advanced** tab.

    Use this tab to enter Tcl expressions.

    A text input field displays.

16. Type an expression into the text input field.
    If you expect groups to include only one entry, you can type an expression similar to this one.

    ```
    expr { [mcget {session.localdb.groups}] eq "eng" }
    ```

    If you expect groups to include multiple entries, you can type an expression similar to this one

    ```
    expr { [mcget {session.localdb.groups}] contains "sales" }
    ```

17. Click **Finished**.
    The popup screen closes.

18. Add more branch rules to provide branches for different user groups.

19. Click **Save**.
    The properties screen closes and the visual policy editor displays.

20. (Optional) Add any other branches and actions that you need to complete the access policy.

When the access policy runs and takes the branch with the Local Database read action, additional branching is done based on group membership.

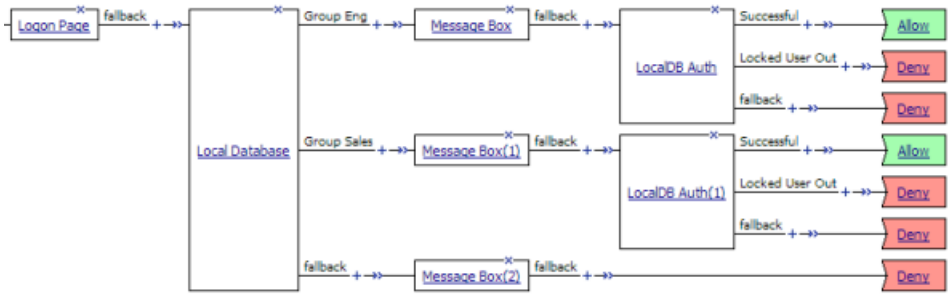


Figure 16: Sample access policy that uses local user DB groups in a branching strategy

Overview: Branching in an access policy based on local user database groups

Chapter 14

OCSP Authentication

- *About OCSP authentication*
- *Task summary for OCSP authentication*
- *Policy example for OCSP authentication*
- *OCSP session variables*
- *OCSP authentication troubleshooting tips*

About OCSP authentication

Access Policy Manager® supports authenticating and authorizing the client against Online Certificate Status Protocol (OCSP). *OCSP* is a mechanism used to retrieve the revocation status of an X.509 certificate by sending the certificate information to a remote OCSP responder. This responder maintains up-to-date information about the certificate's revocation status. OCSP ensures that Access Policy Manager always obtains real-time revocation status during the certificate verification process.

Important: *Access Policy Manager must include an OCSP responder configuration for every OCSP responder that exists.*

Task summary for OCSP authentication

This task list includes all steps required to set up this configuration. If you are adding OCSP authentication to an existing access policy, you do not need to create another access profile.

Task list

Configuring an AAA OCSP responder
Creating an access profile
Configuring OCSP authentication
Configuring a client SSL profile for OCSP
Adding client-side SSL and access profiles to a virtual server

Configuring an AAA OCSP responder

Before you can specify a certificate authority file for an OCSP responder, the file must be imported into the system SSL certificate list.

You create an OCSP responder in Access Policy Manager® when you want to use OCSP authentication for user access.

1. On the Main tab, click **Access Policy** > **AAA Servers** > **OCSP Responders**.
   The OCSP Responder Servers list screen opens.
2. Click **Create**.
   The New Server properties screen opens.
3. In the **Name** field, type a unique name for the authentication server.
4. In the **URL** field, type the URL used to contact the OCSP service on the responder.

   You can skip this step if you did not select the **Ignore AIA** check box and all users have certificates with the correct AIA structure.
5. (Optional) From the **Certificate Authority File** list, select an SSL certificate.
6. Click **Finished**.
   The new server displays on the list.

You can select this OCSP Responder from an OCSP Auth access policy item.

Creating an access profile

You create an access profile to provide the access policy configuration for a virtual server that establishes a secured session.

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. Click **Create**.
   The New Profile screen opens.
3. Type a name for the access profile.
4. From the **Profile Type** list, select one:
   - **APM-LTM** - Select for a web access management configuration.
   - **SSO** - Select only when you do not need to configure an access policy.
   - **SWG - Explicit** - Select to configure access using Secure Web Gateway explicit forward proxy.
   - **SWG - Transparent** - Select to configure access using Secure Web Gateway transparent forward proxy.
   - **SSL-VPN** - Select for other types of access, such as network access, portal access, application access. (Most access policy items are available for this type.)
   - **ALL** - Select for any type of access.

   Additional settings display.
5. In the Language Settings area, add and remove accepted languages, and set the default language.
   A browser uses the highest priority accepted language. If no browser language matches the accepted languages list, the browser uses the default language.
6. Click **Finished**.

This creates an access profile with a default access policy.

Configuring OCSP authentication

Add an OCSP authentication item to an access policy when you want to authenticate using OCSP.

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. In the Access Policy column, click the **Edit** link for the access profile you want to configure.
   The visual policy editor opens the access policy in a separate screen.
3. Click the **(+)** icon anywhere in the access policy to add a new action item.
   A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
4. From the Authentication tab, select either `Client Cert Inspection` or `On-Demand Cert Auth`, and click **Add item**.
   Client Cert Inspection checks the result of an SSL handshake request that occurs at the start of an SSL session. On Demand Cert Auth performs an SSL re-handshake and checks the result. The CRLDP and OCSP Auth actions require certificate information made available by one of these access policy items.
5. Click the **(+)** icon anywhere in the access policy to add a new action item.
   A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
6. Select `OSCP Auth`, and click **Add item.**
   A properties popup screen opens.

7. From the **OCSP Responder** list, select an OCSP responder.
8. Click **Save**.
   The properties screen closes and the visual policy editor displays.
9. Click **Apply Access Policy** to save your configuration.

This creates an access policy that uses OCSP authentication.

To put an access policy into effect, add it to a virtual server.

Configuring a client SSL profile for OCSP

You need a clientssl profile to use OCSP authentication from an access policy.

1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Client**.
   The Client profile list screen opens.
2. Click **Create**.
   The New Client SSL Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. Select **clientssl** in the **Parent Profile** list.
5. Scroll down to the Client Authentication area.
6. Select the **Custom** check box for **Client Authentication**.
   The settings become available.
7. From the **Client Certificate** list, select the option that is applicable to the item you selected when you edited the access policy.
   - Select **request** if the Client Cert Inspection agent is used in the access policy.
   - Select **ignore** if the On-Demand Cert Auth agent is used.
8. From the **Trusted Certificate Authorities** list, select the Certificate Authority that issues the user certificates.
9. From the **Advertised Certificate Authorities** list, select the advertised Certificate Authority file for client certificate authentication.
10. Click **Finished**.

To put a client SSL profile into effect, you must add it to a virtual server.

Adding client-side SSL and access profiles to a virtual server

You associate the client SSL and access profiles with the virtual server so that the BIG-IP® system handles client-side SSL traffic as specified, and so that Access Policy Manager®can apply the access profile to incoming traffic.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the name of the virtual server you want to modify.
3. For the **SSL Profile (Client)** setting, from the **Available** list, select the name of the Client SSL profile you previously created, and using the Move button, move the name to the **Selected** list.
4. In the Access Policy area, from the **Access Profile** list, select the access profile.
5. Click **Update** to save the changes.

The access policy and client-side SSL profiles are now associated with the virtual server.

Policy example for OCSP authentication

This is an example of an access policy with all the associated elements needed to authenticate and authorize users with OCSP authentication. Notice that you must add either the Client Cert Inspection agent or the On-Demand Cert Auth agent before the OCSP Auth object in your access policy. One of those agents is required in order to receive the X.509 certificate from the user. This is also important since both agents store the user information as well as the issuer certificates in the session variables. This allows the OCSP Auth agent to check the revocation status of the user's certificate.

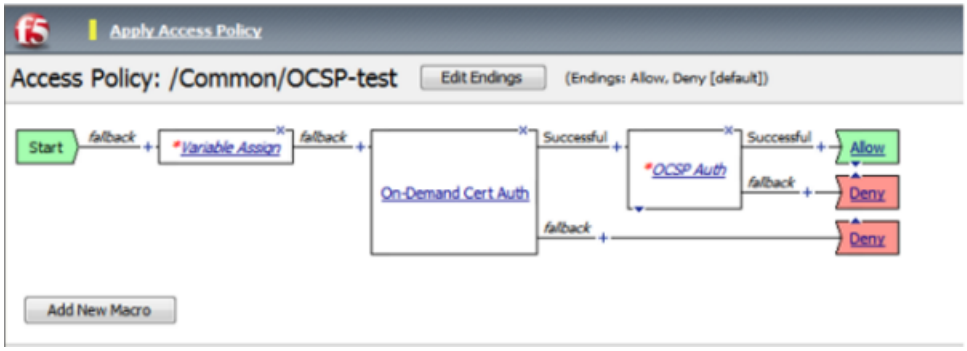


Figure 17: How OCSP works

OCSP session variables

When the OCSP Auth access policy item runs, it populates session variables, which are then available for use in access policy rules. This table lists the session variables for the OCSP access policy item and for the certificate item used in the access policy.

Session variables for OCSP

Session Variable	Description
`session.ssl.cert.whole`	Provides the client certificate received from the user in PAM format.
`session.ssl.cert.certissuer`	Provides the issuer certificate of the client certificate in PAM format.
`session.ocsp.last.result`	Sets the result of the OCSP authentication. The available values are: • 0: Failed • 1: Passed
`session.ocsp.last.status`	Sets the status of the authentication to Failed.

OCSP authentication troubleshooting tips

You might run into problems with OCSP authentication in some instances. Follow these tips to try to resolve any issues you might encounter.

OCSP auth and query troubleshooting

Possible error messages	Possible explanations and corrective actions
`No AAA server associated with the agent`	Make sure that a valid OCSP responder configuration is assigned to the OCSP agent in the access policy.
`User/Issuer certificate not found for the session`	The user/issuer certificate session variables are missing. Make sure that either the Client Cert Inspection agent or On-Demand Cert Auth agent is configured in the access policy (or use a variable assignment agent to create them).
`Failure to connect to OCSP responder (BIO callback failure)`	Make sure that the OCSP responder is up and running and reachable from the BIG-IP® system.
`Error parsing the OCSP response (invalid response)`	Indicates that no valid basic response was found in the OCSP response. Check the configuration on the remote OCSP responder.
`Error signing OCSP request`	Make sure that the signing certificate and key are valid.
`No valid nonce found in the response`	This happens when the nonce setting is enabled on the OCSP responder configuration and the received OCSP response does not contain a valid nonce. Check the remote OCSP responder connection and setting.
`Nonce verification failed`	This happens when the nonce received in the response does not match with the nonce sent in the request. Make sure that the connection from BIG-IP system to OCSP responder is secure.
`Failure to verify response`	Make sure that the OCSP responder has a valid CA and verify other certificate settings.
`Status times invalid`	Make sure that the BIG-IP system and OCSP responder clocks are in sync.
`OCSP response - Cert with serial number 'x' has been revoked`	Indicates that the status of the user certificate is revoked.
`Failed to add cert to OCSP request`	Indicates a failure in creating the OCSP request; either the supplied user/issuer certificates are not valid or the CertID digest configured in the OCSP responder setting is not valid.

Chapter 15

CRLDP Authentication

- *About CRLDP configuration*
- *About AAA high availability*
- *Task summary for CRLDP configuration*
- *Testing AAA high availability for supported authentication servers*
- *Example access policy for CRLDP authentication*
- *CRLDP session variables*
- *CRLDP authentication troubleshooting tips*

About CRLDP configuration

Access Policy Manager® supports retrieving Certificate Revocation Lists (CRLs) from network locations (distribution points). A Certificate Revocation List Distribution Point (CRLDP) AAA server defines how to access a CRL file from a distribution point. A distribution point is either an LDAP Uniform Resource Identifier (URI), a directory path that identifies the location where the CRLs are published, or a fully qualified HTTP URL.

About AAA high availability

Using AAA high availability with Access Policy Manager® (APM®), you can configure multiple authentication servers to process requests, so that if one authentication server goes down or loses connectivity, the others can resume authentication requests, and new sessions can be established, as usual.

Note: *Although new authentications fail if the BIG-IP® system loses connectivity to the server, existing sessions are unaffected provided that they do not attempt to re-authenticate.*

APM supports the following AAA servers for high availability: RADIUS, Active Directory, LDAP, CRLDP, and TACACS+. APM supports high availability by providing the option to create a pool of server connections when you configure the supported type of AAA server.

Note: *If you use AAA with pools, such as RADIUS pools or Active Directory pools, APM assigns each pool member with a different number for the pool member's priority group value. Since APM does not support AAA load balancing, APM must define each pool member with a different priority group. The priority group number increases automatically with each created pool member.*

Task summary for CRLDP configuration

This task list includes all steps required to set up this configuration. If you are adding CRLDP items to an existing access policy, you do not need to create another access profile.

Task list

Configuring an AAA server for CRLDP
Creating an access profile
Configuring an access policy that uses CRLDP authentication
Configuring a client SSL profile for CRLDP
Adding client-side SSL and access profiles to a virtual server

Configuring an AAA server for CRLDP

Create a CRLDP AAA configuration to specify how to access certificate revocation lists (CRLs).

1. On the Main tab, click **Access Policy** > **AAA Servers** > **CRLDP**.

The CRLDP Servers list screen opens.

2. Click **Create**.
   The New Server properties screen opens.
3. In the **Name** field, type a unique name for the authentication server.
4. For the **Server Connection** setting, select one of these options:
   - Select **Use Pool** to set up high availability for the AAA server.
   - Select **Direct** to set up the AAA server for standalone functionality.
   - Select **No Server** to use a fully qualified HTTP URL as the CRL location.

   Note: *The BIG-IP system uses the URI from the user's certificate.*

   Note: *When you select **No Server**, the screen updates to omit the fields that are not necessary, such as **Server Addresses**, **Server Port**, and so on.*
5. If you selected **Use Pool**, type a name in the **Server Pool Name field**.
   You create a pool of servers on this screen.
6. Provide the addresses required for your server connection:
   - If you selected **Direct**, type an IP address in the **Server Address** field.
   - If you selected **Use Pool**, for each pool member you want to add, type an IP address and click **Add**.

   Note: *When you configure a pool, you have the option to type the server address in route domain format: `IPAddress%RouteDomain`.*
7. If you selected **Use Pool**, you have the option to select a **Server Pool Monitor** to track the health of the server pool.
8. If you specified **Use Pool** or **Direct** for the server connection, the **Base DN** field displays; type a CRLDP base distinguished name into it.
   This setting applies for certificates that specify the CRL distribution point in directory name (dirName) format. Access Policy Manager® uses the Base DN when the value of the X509v3 attribute, `crlDistributionPoints`, is of type `dirName`. In this case, Access Policy Manager tries to match the value of the crlDistributionPoints attribute to the Base DN value. An example of a Base DN value is `cn=lxxx,dc=f5,dc=com`.

   Note: *If the client certificate includes the distribution point extension in LDAP URI format, the IP address, Base DN, and Reverse DN settings configured on the agent are ignored; they are specific to directory-based CRLDP. All other settings are applicable to both LDAP URI and directory-based CRL DPs.*
9. Click **Finished**.
   The new server displays on the list.

An CRLDP AAA server is available for use in a CRLDP Auth agent in an access policy.

Creating an access profile

You create an access profile to provide the access policy configuration for a virtual server that establishes a secured session.

1. On the Main tab, click **Access Policy** > **Access Profiles**.

The Access Profiles List screen opens.

2. Click **Create**.
   The New Profile screen opens.
3. Type a name for the access profile.
4. From the **Profile Type** list, select one:
   - **APM-LTM** - Select for a web access management configuration.
   - **SSO** - Select only when you do not need to configure an access policy.
   - **SWG - Explicit** - Select to configure access using Secure Web Gateway explicit forward proxy.
   - **SWG - Transparent** - Select to configure access using Secure Web Gateway transparent forward proxy.
   - **SSL-VPN** - Select for other types of access, such as network access, portal access, application access. (Most access policy items are available for this type.)
   - **ALL** - Select for any type of access.

   Additional settings display.
5. In the Language Settings area, add and remove accepted languages, and set the default language.
   A browser uses the highest priority accepted language. If no browser language matches the accepted languages list, the browser uses the default language.
6. Click **Finished**.

This creates an access profile with a default access policy.

Configuring an access policy that uses CRLDP authentication

You add CRLDP authentication to an access policy when you want to verify certificate revocation status before granting a user access.

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. In the Access Policy column, click the **Edit** link for the access profile you want to configure.
   The visual policy editor opens the access policy in a separate screen.
3. Click the **(+)** icon anywhere in the access policy to add a new action item.
   A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
4. From the Authentication tab, select either `Client Cert Inspection` or `On-Demand Cert Auth`, and click **Add item**.
   Client Cert Inspection checks the result of an SSL handshake request that occurs at the start of an SSL session. On Demand Cert Auth performs an SSL re-handshake and checks the result. The CRLDP and OCSP Auth actions require certificate information made available by one of these access policy items.
5. Click **Save.**
   The popup screen closes.
6. Click the **(+)** icon anywhere in the access policy to add a new action item.
   A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
7. On the Authentication tab, select **CRLDP Auth**, then click **Add item**.
   A properties popup screen opens.
8. From the **CRLDP Server** list, select a server.
9. Click **Save**.
   The popup screen closes.

10. To grant access at the end of any branch, change the ending from **Deny** to **Allow**:
    a) Click **Deny**.
    The default branch ending is **Deny**.
    A popup screen opens.
    b) Select **Allow** and click **Save**.
    The popup screen closes. The **Allow** ending displays on the branch.

11. Click **Apply Access Policy** to save your configuration.

The access policy is complete.

To put the access policy into effect, attach the access profile to a virtual server.

Configuring a client SSL profile for CRLDP

You need a client SSL profile to use CRLDP authentication from an access policy.

1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Client**.
   The Client profile list screen opens.
2. Click **Create**.
   The New Client SSL Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, select **clientssl**.
5. If the access policy uses On-Demand certificate authentication, perform these substeps:
   a) From the **Configuration** list, select **Advanced**.
   Additional settings display.
   b) Select the **Custom** check box for **Configuration**.
   The settings become available.
   c) In the **Ciphers** field, type the name of a NATIVE cipher.
   The list of supported NATIVE ciphers includes these:
   - RC4-MD5
   - RC4-SHA
   - AES128-SHA
   - AES256-SHA
   - DES-CBC3-SHA
   - DES-CBC-SHA
   - EXP1024-RC4-MD5
   - EXP1024-RC4-SHA
   - EXP1024-DES-CBC-SHA
   - EXP-RC4-MD5
   - EXP-DES-CBC-SHA
   - NULL-MD5
   - NULL-SHA

6. From the **Client Certificate** list, select the option that is applicable to the item you selected when you edited the access policy.
   - Select **request** if the Client Cert Inspection agent is used in the access policy.
   - Select **ignore** if the On-Demand Cert Auth agent is used.

7. From the **Trusted Certificate Authorities** list, select the Certificate Authority that issues the user certificates.
8. (Optional) From the **Advertised Certificate Authorities** list, select the Certificate Authority that issues the user certificates.
9. Click **Finished**.

A new client SSL profile is available.

Note: *CRLDP authentication does not verify a certificate revocation list if one is selected in the client SSL profile. CRLDP authentication verifies the certificate revocation list (CRL) at a distribution point defined in the CRLDP AAA server.*

Adding client-side SSL and access profiles to a virtual server

You associate the client SSL and access profiles with the virtual server so that the BIG-IP® system handles client-side SSL traffic as specified, and so that Access Policy Manager®can apply the access profile to incoming traffic.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the name of the virtual server you want to modify.
3. For the **SSL Profile (Client)** setting, from the **Available** list, select the name of the Client SSL profile you previously created, and using the Move button, move the name to the **Selected** list.
4. In the Access Policy area, from the **Access Profile** list, select the access profile.
5. Click **Update** to save the changes.

The access policy and client-side SSL profiles are now associated with the virtual server.

Testing AAA high availability for supported authentication servers

To effectively test that high availability works for your authentication servers, you should have two servers that are accessible, where you can remove one of them from the network.

Note: *High availability is supported for these authentication server types only: RADIUS, Active Directory, LDAP, CRLDP, and TACACS+.*

If you configured a supported authentication server type to use a pool of connection servers, you can test the configuration using these steps.

1. Begin a `tcpdump` on the Access Policy Manager®, using a protocol analyzer, and scanning for packets destined for the specific port for your authentication server.
2. Log in to the virtual server with both servers active.
3. Using the `tcpdump` records, verify that the requests are being sent to the higher priority server.
4. Log out of the virtual server.
5. Disable the higher-priority server.
6. Log in to the virtual server again.
7. Verify that the request is being sent to the other server.

8. Log out again, re-enabling the server, and try one more time to verify that the new requests are being sent to the high priority server.

Example access policy for CRLDP authentication

This is an example of an access policy with all the associated elements needed to retrieve CRLs using CRLDP. Notice that you must add either the Client Cert Inspection agent or On-Demand Cert Auth agent before the CRLDP object in your access policy. One of those agents is required in order to receive the X.509 certificate from the user. This is also important because both agents store the user information, as well as the issuer certificates, in the session variables. This allows the CRDLP Auth agent to check the revocation status of the user's certificate.

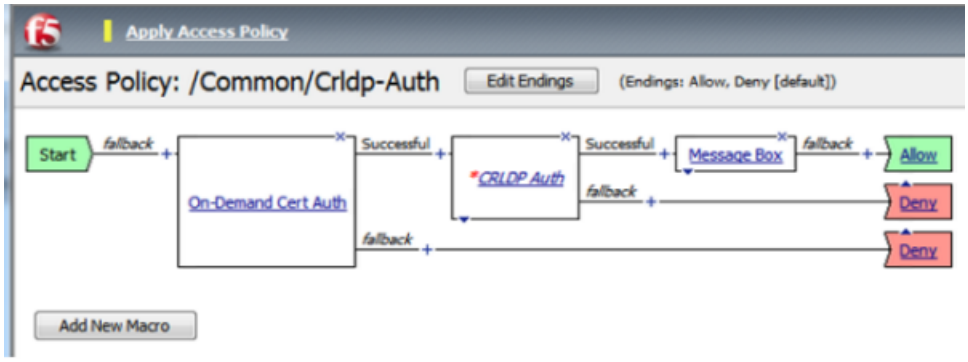


Figure 18: How CRLDP works

CRLDP session variables

When the CRLDP Auth access policy item runs, it populates session variables which are then available for use in access policy rules. The table lists the session variables for the CRLDP access policy item and for the certificate item used in the access policy.

Session variables for CRLDP

Session Variable	Description
`session.ldap.ssl.cert.whole`	Provides the client certificate received from the user in PAM format.
`session.ssl.cert.certissuer`	Provides the issuer certificate of the client certificate in PAM format.
`session.crldp.last.result`	Sets the result of the CRLDP authentication. The available values are: • 0: Failed • 1: Passed
`session.crldp.last.status`	Sets the status of the authentication to Failed.

CRLDP authentication troubleshooting tips

You might run into problems with CRLDP authentication in some instances. Follow these tips to try to resolve any issues you might encounter.

CRLDP auth and query troubleshooting

Possible error messages	Possible explanations and corrective actions
`No AAA server associated with the agent`	Make sure that a valid CRLDP responder configuration is assigned to the CRLDP agent in the access policy.
`User/Issuer certificate not found for the session`	The user/issuer certificate session variables are missing. Make sure that either the Client Cert Inspection agent or On-Demand Cert Auth agent is configured in the access policy (or use a variable assignment agent to create them).
`Failure to connect to CRLDP server`	Make sure that the CRLDP server is up and running and reachable from the BIG-IP system.
`No LDAP URL found in the DP list`	Indicates that no valid CRL DP is configured on the LDAP server. Make sure that the LDAP server used in the CRLDP server configuration has valid CRL DPs configured.
`CRLDP response - Cert with serial number 'x' has been revoked`	Indicates that the status of the user certificate is revoked.

Chapter 16

On-Demand Certificate Authentication

- *Overview: Requesting and validating an SSL certificate on demand*

Overview: Requesting and validating an SSL certificate on demand

Typically, when a client makes an HTTPS request, an SSL handshake request occurs at the start of an SSL session. You can configure a client SSL profile to skip the initial SSL handshake and add the On-Demand certificate authentication agent to the access policy to re-negotiate the SSL connection later. Access Policy Manager® can perform the certificate request and validation task that is normally performed by the target server, on demand.

Use the agent when you want to request and validate a certificate only after a user has already completed some other steps (logged on, gone through an authentication process, or anything else you require). Wherever you place the On-Demand authentication action in your access policy, it performs an SSL re-handshake.

You might want to use this agent, for example, if all employees must gain access to the network before only a few employees can gain access to servers with sensitive information.

Exchanging SSL Certificates

Before you can use On-Demand certificate authentication successfully, you must exchange certificates between clients and the BIG-IP® system.

The client needs a valid certificate with which to respond to a certificate request. The BIG-IP system includes a self-signed certificate that you can export and install on the client. As an alternative to the self-signed certificate, you can import a certificate and corresponding key (issued by your organization CA) into the BIG-IP system and install that on the client.

The BIG-IP systems needs the client root certificate installed on it.

Tasks

Creating a custom Client SSL profile

Configure a client SSL profile to skip the initial SSL handshake and thereby support Access Policy Manager® (APM™) to perform an SSL handshake on demand.

1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Client**.
   The Client profile list screen opens.
2. Click **Create**.
   The New Client SSL Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. Select **clientssl** in the **Parent Profile** list.
5. Scroll down to the Client Authentication area.
6. Select the **Custom** check box for **Client Authentication**.
   The settings become available.
7. For the **Client Certificate** setting, select **ignore**.
   When ignore is selected, the BIG-IP system skips the initial SSL handshake.
8. For the **Trusted Certificate Authorities** setting, select a trusted certificate authority.
9. Click **Finished**.

Adding On-Demand certificate authentication to an access policy

To successfully pass the On-Demand certificate authentication, the client browser must have a valid SSL certificate for the BIG-IP system.

Note: *The client browser might stop responding if the client fails to provide a certificate. We strongly recommend that you add a Decision Box action in which you ask the user whether a valid certificate is installed and provide an option to not proceed to the On-Demand Cert Auth action when a valid certificate is not installed.*

Add an On-Demand Cert Auth agent to an access policy to request and validate an SSL certificate anywhere in the session.

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. In the Access Policy column, click the **Edit** link for the access profile you want to configure.
   The visual policy editor opens the access policy in a separate screen.
3. Click the **(+)** icon anywhere in the access policy to add a new action item.
   A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
4. Select the **Authentication** tab.
   The tab displays a list of authentication actions.
5. Select **On-Demand Cert Auth** and click **Add Item**.
   A properties screen opens.
6. From the **Auth Mode** list, select one of these:
   - **Request** This is the default mode.
   - **Required** For an iPod or an iPhone, you must select this mode. (You can select this mode for other clients as well.)

     Note: *To pass a certificate check using Safari, you will be asked to select the certificate multiple times. This is expected behavior.*
7. Click **Save**.
   The properties screen closes and the visual policy editor displays.
8. Click the **Apply Access Policy** link to apply and activate the changes to the access policy.

The On-Demand Cert Auth action is included and applied to the access policy.

Adding client-side SSL and access profiles to a virtual server

You associate the client SSL and access profiles with the virtual server so that the BIG-IP® system handles client-side SSL traffic as specified, and so that Access Policy Manager®can apply the access profile to incoming traffic.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the name of the virtual server you want to modify.
3. For the **SSL Profile (Client)** setting, from the **Available** list, select the name of the Client SSL profile you previously created, and using the Move button, move the name to the **Selected** list.

4. In the Access Policy area, from the **Access Profile** list, select the access profile.
5. Click **Update** to save the changes.

The access policy and client-side SSL profiles are now associated with the virtual server.

Chapter 17

Client Certificate Inspection

- *About client certificate inspection*
- *Task summary for client certificate inspection*

About client certificate inspection

The Client Cert Inspection access policy item checks the result of the SSL handshake that occurs at the start of a session. It does not, however, negotiate an SSL session. It relies on settings in a client SSL profile that is added to the virtual server. The Client Cert Inspection item can provide the result of the SSL handshake, including certificate revocation status when the client SSL profile specifies a certificate revocation list (CRL).

Task summary for client certificate inspection

To complete this configuration, you need an access profile and a virtual server configured. Checking the validity of a client certificate is very likely to be one of many items you add to an access policy.

Task list

Creating a client SSL profile for certificate inspection

Configuring an access policy to confirm client certificate validity

Creating a client SSL profile for certificate inspection

The BIG-IP® system supplies a default certificate and a `ca-bundle.crt` file that includes all well-known public certificate authority (CA) certificates for client-side processing. Before you create a client SSL profile, you might want to configure a trusted certificate to use for client-side processing. To verify certificate revocation status, you must have obtained a certificate revocation list (CRL) and imported it to the SSL Certificate List.

You create a custom client SSL profile to request an SSL certificate from the client at the start of the session. This enables a Client Cert Inspection item in an access policy to check whether a valid certificate was presented.

1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Client**.
   The Client profile list screen opens.
2. Click **Create**.
   The New Server SSL Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, select **clientssl**.
5. Scroll down to the Client Authentication area.
6. Select the **Custom** check box for **Client Authentication**.
   The settings become available.
7. From the **Client Certificate** list, select **request**.

   Alternatively, select **require**; however, if you do, the user must provide a valid client certificate or the connection is not allowed.
8. (Optional) If you imported a CRL, select it from the **Certificate Revocation List (CRL)** list.

   If you are using this client SSL profile in conjunction with an access policy that performs OCSP Responder authentication or CRLDP authentication, do not select a CRL.
9. Click **Finished**.

To put this client SSL profile into effect, select it in a virtual server that is configured to accept HTTPS traffic.

Configuring an access policy to confirm client certificate validity

Add a client certificate inspection item to an access policy when you want to check whether the client presented a valid certificate at the start of the session.

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. In the Access Policy column, click the **Edit** link for the access profile you want to configure.
   The visual policy editor opens the access policy in a separate screen.
3. Click the **(+)** icon anywhere in the access policy to add a new action item.
   A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
4. In the search field type `client`, then select `Client Cert Inspection` from the results list, and click **Add item**.
   A popup Properties screen displays.
5. Click **Save**.
   The properties screen closes and the visual policy editor displays.
6. Complete the access policy:
   a) Add any additional access policy items you require.
   b) Change the ending from **Deny** to **Allow** on any access policy branch on which you want to grant access.
7. Click **Apply Access Policy** to save your configuration.

To put an access policy into effect, add it to a virtual server.

Chapter 18

One-Time Password Authentication

- *Overview: Providing a one-time password using email*
- *Overview: Providing a one-time password using an external SMS*

Overview: Providing a one-time password using email

Access Policy Manager® supplies an OTP Generate access policy item that generates a one-time time-sensitive password and an OTP Verify item that verifies that a user entered the correct password before that password expired. In between the two actions, you must configure an action that delivers the one-time password to the user. To send the password in an email message, use the Email access policy item. You must have an external SMTP server and you must create an SNMP server configuration for it on the BIG-IP system.

Related access policy macro

A macro template to configure OTP over email is available for use in an access policy. Look at the macro, AD auth query OTP by email and resources, from the visual policy editor to determine whether to use it to help you configure the access policy more quickly.

Task summary

Creating an SMTP server configuration

You specify the SMTP server configuration so that you can send emails through an SMTP server.

1. On the Main tab, click **System** > **Configuration** > **Device** > **SMTP**.
2. Click the **Create** button.
   The New SMTP Configuration screen opens.
3. In the **Name** field, type a name for the SMTP server that you are creating.
4. In the **SMTP Server Host Name** field, type the fully qualified domain name for the SMTP server host.
5. In the **SMTP Server Port Number** field, type a port number.
   For no encryption or TLS encryption, the default is `25`. For SSL encryption, the default is `465`.
6. In the **Local Host Name** field, type the host name used in the SMTP headers in the form of a fully qualified domain name.
   This host name is not the same as the BIG-IP system's host name.
7. In the **From Address** field, type the email address that you want displayed as the reply-to address for the email.
8. From the **Encrypted Connection** list, select the encryption level required for the SMTP server.
9. To require that the SMTP server validates users before allowing them to send email, select the **Use Authentication** check box, and type the user name and password required to validate the user.
10. Click the **Finish** button.

You can now configure the system to use this SMTP server to send emails. For the SMTP mailer to work, you must make sure the SMTP server is on the DNS lookup server list, and configure the DNS server on the BIG-IP® system.

Creating an access policy to send an OTP using email

Before you start this task, configure an access profile.

Create an access policy like this when you need to generate and send a one-time password over email.

Note: *Look at the macro, AD query auth OTP by email and resources, to determine whether to use it to configure an access policy similar to this one.*

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. In the Access Policy column, click the **Edit** link for the access profile you want to configure.
   The visual policy editor opens the access policy in a separate screen.
3. Add actions to authenticate the user and find an email address and a mobile phone number.
   a) Click the **(+)** icon anywhere in your access profile to add a new action item.
   An popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
   b) On the Authentication tab, select **AD Auth** and click **Add Item**.
   A popup properties screen displays.
   c) From the **Server** list, select a server and click **Save**.
   The properties screen closes.
   d) On the Successful branch after the previous action, click the **(+)** icon.
   An Add Item screen opens, listing predefined actions that are grouped on tabs such as General Purpose, Authentication, and so on.
   e) On the Authentication tab, select **AD Query** and click **Add Item**.
   An AD Query is only one way to find the email address for a user. If users normally log on to your system with an email address as their username, you can get the email address using a Logon Page action.
   A popup properties screen displays.
   f) From the **Server** list, select a server.
   g) Click **Add new entry**.
   An empty entry displays under Required Attributes (optional).
   h) Type **mobile** into the **Required Attributes (optional)** field
   After the query, the session.ad.last.attr.mobile variable holds the value.
   i) Click **Add new entry**.
   An empty entry displays under Required Attributes (optional).
   j) Type **mail** into the **Required Attributes (optional)** field
   After the query, the session.ad.last.attr.mail variable holds the value.
   k) Click **Save**.
   The properties screen closes.
4. Generate a one-time password.
   a) On the Successful branch after the previous action, click the **(+)** icon.
   An Add Item screen opens, listing predefined actions that are grouped on tabs such as General Purpose, Authentication, and so on.
   b) On the Authentication tab, select **OTP Generate** and click **Add Item**.
   c) Click **Save**.
   The properties screen closes and the visual policy editor is displayed.
5. Send the OTP to the user through the Email agent.
   a) On the Successful branch after the previous action, click the **(+)** icon.
   An Add Item screen opens, listing predefined actions that are grouped on tabs such as General Purpose, Authentication, and so on.
   b) On the General Purpose tab, select **Email** and click **Add Item**.
   c) From the **SMTP Configuration** list, select a configuration.
   The configuration specifies an external SMTP server to send the email.

d) In the **From** field, type an email address on the system.
e) In the **To** field, type an email address, a session variable, or a session variable and a string.
For example, type `%{session.ad.last.attr.mobile}@providerservice.com` where providerservice.com is supplied by a mobile phone provider.
f) Type a subject in the **Subject** field.
g) In the **Message** field, type the one-time password and anything else the user should know.
One Time Passcode: %{session.otp.assigned.val} Expires after use or in %{session.otp.assigned.ttl} seconds
h) Click **Save**.
The properties screen closes and the visual policy editor is displayed.

6. Add a Logon Page action that requests the one-time password only.
a) On the Successful branch after the previous action, click the **(+)** icon.
An Add Item screen opens, listing predefined actions that are grouped on tabs such as General Purpose, Authentication, and so on.
b) On the Logon tab, select **Logon Page** and click the **Add Item** button.
The Logon Page Agent properties screen opens.
c) From the Logon Page Agent area, on line 1 select **none** from the Type column to remove the user name input field from the logon page; do not change line 2 (password).
d) From the Customization area in **Logon Page Input Field # 2**, type a prompt for the field.
For example, type One-Time Passcode.
e) Click **Save**.
The properties screen closes and the visual policy editor is displayed.

7. Verify the one-time password.
a) On the Successful branch after the previous action, click the **(+)** icon.
An Add Item screen opens, listing predefined actions that are grouped on tabs such as General Purpose, Authentication, and so on.
b) On the Authentication tab, select **OTP Verify** and click **Add Item**.
c) Click **Save**.
The properties screen closes and the visual policy editor is displayed.

8. (Optional) Add any other branches and actions that you need to complete the access policy.
9. Change the Successful rule branch from **Deny** to **Allow** and click the **Save** button.
10. At the top of the window, click the **Apply Access Policy** link to apply and activate your changes to this access policy.
11. Click the **Close** button to close the visual policy editor.

You have an access policy that provides a user with a one-time time-based password over SMTP.

To put the access policy into effect, you must attach it to a virtual server.

Overview: Providing a one-time password using an external SMS

Access Policy Manager® supplies an OTP Generate action that generates a one-time time-sensitive password and an OTP Verify action that verifies that a user entered the correct password before it expired. In between the two actions, you must configure an action that delivers the one-time password to the user. To send the password in a text message, you can use a form-based HTTP authentication agent (if you do not want to

use an Email agent). You pass the one-time password in hidden parameters to a form action. You must create a form action that sends the OTP using an external SMS.

Configuration process

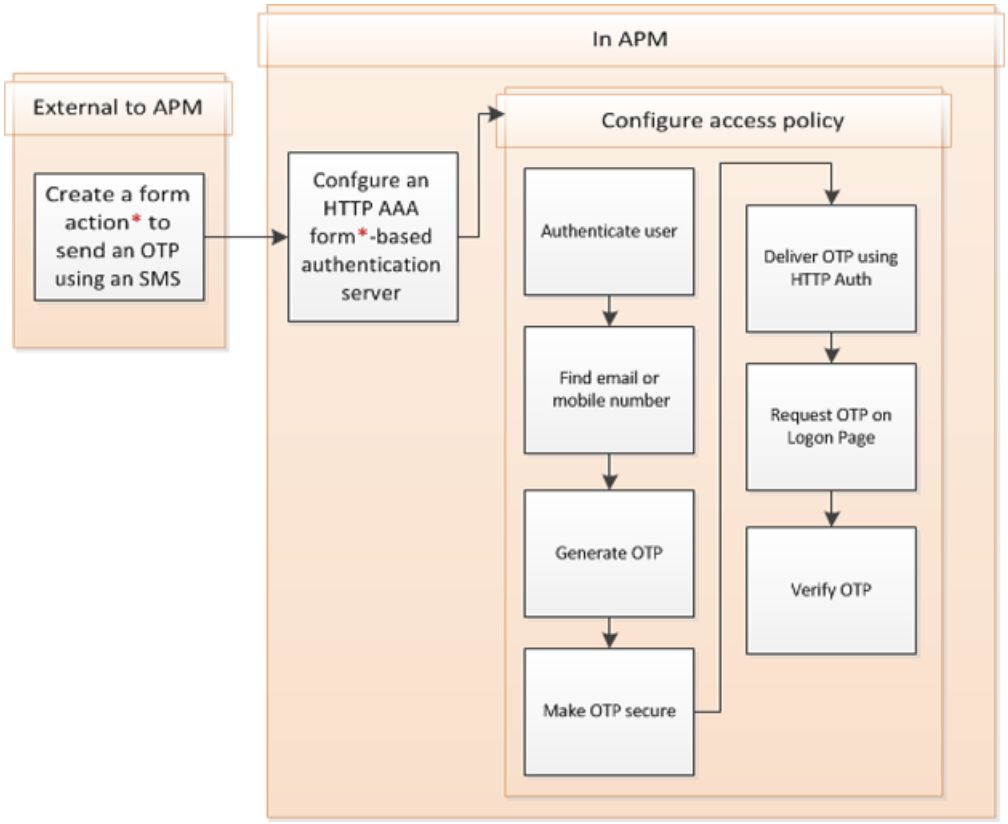


Figure 19: Creating a configuration to send an OTP over SMS using HTTP authentication

Related access policy macro

A macro template to configure an OTP and use the HTTP Auth agent to deliver it is available for use in an access policy. Look at the macro, AD query auth OTP by HTTP and resources, from the visual policy editor to determine whether to use it to help you configure the access policy more quickly.

Task summary

Configuring HTTP form-based authentication to deliver a one-time password

Configure an AAA HTTP server to use a form action that you configured previously to send a one-time password through an external SMS.

1. On the Main tab, click **Access Policy** > **AAA Servers** > **HTTP**.
   The HTTP Servers screen displays.
2. Click **Create**.
   The New Server properties screen opens.
3. In the **Name** field, type a unique name for the authentication server.
4. From the Configuration area, select `Form Based` for the **Authentication Type**.
5. Let the **Form Method** remain at the default setting, **POST**.

6. In the **Form Action** field, type the complete destination URL to process the form.
   Specify a URL for a form action that you created to send a user a one-time password using an SMS.
7. In the **Hidden Form Parameters/Values** field, type parameters and values for the one-time password, the phone number, and any other values that the form action requires.
   Here is an example.

```
     otp_http_mobile "%{session.ad.last.attr.mobile}"
     otp_http_email "%{session.ad.last.attr.mail}"
     otp_http_body "One Time Passcode: %{session.otp.assigned.val} Expires
after use or in %{session.otp.assigned.ttl} seconds"
```

8. From the **Successful Logon Detection Match Type** list, select the method that the authenticating server uses.
9. In the **Successful Logon Detection Match Value** field, type the value that denotes successful logon.
   Type a cookie name, a URL, or a string, depending on the successful logon detection match type you selected.
10. Click **Finished**.
    The new server displays on the list.

An HTTP server for form-based authentication with a one-time password is ready for use.

Creating an access policy to send an OTP using an SMS

Before you start this task, configure an access profile and configure a form action that uses an external SMS to send the one-time password.

Create an access policy like this when you need to generate and send a one-time password as a text message and you do not want to send it using email.

Note: *The macro, AD auth query OTP by HTTP and resources, is available from the visual policy editor and might be useful to configure an access policy similar to this one.*

---

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. In the Access Policy column, click the **Edit** link for the access profile you want to configure.
   The visual policy editor opens the access policy in a separate screen.
3. Add actions to authenticate the user and find a mobile phone number.
   a) Click the **(+)** icon anywhere in your access profile to add a new action item.
      An popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
   b) From the Authentication tab, select **AD Auth** and click **Add Item**.
      A pop-up properties screen displays.
   c) From the **Server** list, select a server and click **Save**.
      The properties screen closes.
   d) On the Successful branch after the previous action, click the **(+)** icon.
      An Add Item screen opens, listing predefined actions that are grouped on tabs such as General Purpose, Authentication, and so on.
   e) On the Authentication tab, select **AD Query** and click **Add Item**.
      A pop-up properties screen displays.

f) From the **Server** list, select a server.
g) Click **Add new entry**.
An empty entry displays under Required Attributes (optional).
h) Type **mobile** into the **Required Attributes (optional)** field
i) Click **Save**.
The properties screen closes.

4. Generate a one-time password.
a) On the Successful branch after the previous action, click the **(+)** icon.
An Add Item screen opens, listing predefined actions that are grouped on tabs such as General Purpose, Authentication, and so on.
b) From the Authentication tab, select **OTP Generate** and click **Add Item**.
c) Click **Save**.
The properties screen closes and the visual policy editor is displayed.

5. Make the OTP secure.
a) On the Successful branch after the previous action, click the **(+)** icon.
An Add Item screen opens, listing predefined actions that are grouped on tabs such as General Purpose, Authentication, and so on.
b) From the Assignment tab, select **Variable Assign** and click **Add Item**.
A properties screen opens.
c) Click **Add new entry**.
An **Empty** entry displays.
d) Click the **change** link in the new entry.
A popup screen opens.
e) From the **Unsecure** list, select **Secure**.
f) In the Custom Variable text box, type `session.user.otp.pwd`.
g) In the Custom Expression text box, type `expr { [mcget {session.user.otp.pw}]}`.
h) Click **Finished**.
The popup screen closes.

6. Send the OTP through the HTTP Auth agent.
a) On the Successful branch after the previous action, click the **(+)** icon.
An Add Item screen opens, listing predefined actions that are grouped on tabs such as General Purpose, Authentication, and so on.
b) From the Authentication tab, select **HTTP Auth** and click **Add Item**.
c) From the AAA server list, select the HTTP form-based server that you configured previously.
d) Click **Save**.
The properties screen closes and the visual policy editor is displayed.

7. Add a Logon Page action that requests only the one-time password.
a) On the Successful branch after the previous action, click the **(+)** icon.
An Add Item screen opens, listing predefined actions that are grouped on tabs such as General Purpose, Authentication, and so on.
b) From the Logon Page tab, select **Logon Page** and click **Add Item**.
A pop-up properties screen displays.
c) From the Logon Page Agent area, on line 1 select **password** from the Type column and change the post and session variable names.
The variable name password is acceptable.
d) From the Customization area in **Logon Page Input Field # 1**, type a prompt for the field.

For example, type One-Time Passcode.

e) Click **Save**.
The properties screen closes and the visual policy editor is displayed.

8. Verify the one-time password.

a) On the Successful branch after the previous action, click the **(+)** icon.
An Add Item screen opens, listing predefined actions that are grouped on tabs such as General Purpose, Authentication, and so on.

b) From the Authentication tab, select **OTP Verify** and click **Add Item**.

c) Click **Save**.
The properties screen closes and the visual policy editor is displayed.

9. (Optional) Add any other branches and actions that you need to complete the access policy.

10. Change the Successful rule branch from **Deny** to **Allow** and click the **Save** button.

11. At the top of the window, click the **Apply Access Policy** link to apply and activate your changes to this access policy.

12. Click the **Close** button to close the visual policy editor.

You have an access policy that uses HTTP authentication to provide a user with a one-time time-based password over SMS.

To put the access policy into effect, you must attach it to a virtual server.

Chapter 19

TACACS+ Authentication and Accounting

- *About TACACS+ authentication and accounting*
- *About AAA high availability*
- *Task summary for TACACS+ authentication and accounting*
- *Testing AAA high availability for supported authentication servers*
- *Example access policy for TACACS+ authentication and accounting*
- *TACACS+ session variables for access policy rules*
- *TACACS+ authentication troubleshooting tips*

About TACACS+ authentication and accounting

Access Policy Manager® (APM®) supports authenticating and authorizing the client against Terminal Access Controller Access Control System (TACACS+) servers. *TACACS+* is a mechanism used to encrypt the entire body of the authentication packet. If you use TACACS+ authentication, user credentials are authenticated on a remote TACACS+ server. If you use the TACACS+ Accounting feature, the accounting service sends `start` and `stop` accounting records to the remote server.

APM supports TACACS+ authentication with the TACACS+ Auth access policy item and supports TACACS+ accounting with the TACACS+ Acct access policy item.

Important: *APM must include a TACACS+ server configuration for every TACACS+ server that exists.*

About AAA high availability

Using AAA high availability with Access Policy Manager® (APM®), you can configure multiple authentication servers to process requests, so that if one authentication server goes down or loses connectivity, the others can resume authentication requests, and new sessions can be established, as usual.

Note: *Although new authentications fail if the BIG-IP® system loses connectivity to the server, existing sessions are unaffected provided that they do not attempt to re-authenticate.*

APM supports the following AAA servers for high availability: RADIUS, Active Directory, LDAP, CRLDP, and TACACS+. APM supports high availability by providing the option to create a pool of server connections when you configure the supported type of AAA server.

Note: *If you use AAA with pools, such as RADIUS pools or Active Directory pools, APM assigns each pool member with a different number for the pool member's priority group value. Since APM does not support AAA load balancing, APM must define each pool member with a different priority group. The priority group number increases automatically with each created pool member.*

Task summary for TACACS+ authentication and accounting

This task list includes all steps required to set up this configuration. If you are adding TACACS+ authentication or accounting to an existing access policy, you do not need to create another access profile and the access policy might already include a logon page.

Task list

Configuring a TACACS+ AAA server for authentication and authorization

Using TACACS+ authentication in an access policy

Configuring a TACACS+ AAA server for authentication and authorization

1. On the Main tab, click **Access Policy** > **AAA Servers** > **TACACS+**.
   The TACACS+ Servers list screen opens.
2. Click **Create**.
   The New Server properties screen opens.
3. In the **Name** field, type a unique name for the authentication server.
4. For the **Server Connection** setting, select one of these options:
   - Select **Use Pool** to set up high availability for the AAA server.
   - Select **Direct** to set up the AAA server for standalone functionality.
5. If you selected **Use Pool**, type a name in the **Server Pool Name field**.
   You create a pool of servers on this screen.
6. Provide the addresses required for your server connection:
   - If you selected **Direct**, type an IP address in the **Server Address** field.
   - If you selected **Use Pool**, for each pool member you want to add, type an IP address and click **Add**.

   Note: *When you configure a pool, you have the option to type the server address in route domain format:* `IPAddress%RouteDomain`.
7. If you selected **Use Pool**, you have the option to select a **Server Pool Monitor** to track the health of the server pool.
8. In the **Service Port** field, type a TACACS+ service port or select one from the list. The default is `49`.
9. In the **Secret** field, type a secret key to use to encrypt and decrypt packets sent or received from the server, and then re-type the secret key in the **Confirm Secret** field.
10. For the **Service** setting, select the name of the service for the user who is being authenticated to use.
    Identifying the service enables the TACACS+ server to behave differently for different types of authentication requests.
11. Click **Finished**.
    The new server displays on the list.

Using TACACS+ authentication in an access policy

You configure an access policy with a TACACS+ Auth action to provide TACACS+ authentication as an authentication option for users trying to gain access.

1. In the Access Policy column, click the **Edit** link for the access profile you want to configure.
   The visual policy editor opens the access policy in a separate screen.
2. Click the **(+)** icon anywhere in the access policy to add a new action item.
   A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
3. On the Logon tab, select **Logon Page** and click the **Add Item** button.
   The Logon Page Agent properties screen opens.
4. Make any changes that you require to the logon page properties and click **Save**.
   The properties screen closes and the visual policy editor displays.
5. Click the **(+)** icon anywhere in the access policy to add a new action item.

A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.

6. Select `TACACS+ Auth`, and click **Add item.**
   A properties popup screen opens.
7. From the **AAA Server** list, select the TACACS+ server to use for authentication.
8. (Optional) Add any other branches and actions that you need to complete the access policy.
9. Click **Save**.
   The properties screen closes and the visual policy editor displays.
10. Click **Apply Access Policy** to save your configuration.

This creates an access policy that presents a user with a logon page, and then uses the input credentials to authenticate the user with an external TACACS+ server specified in the TACACS+ AAA server that you select.

To apply this access policy to network traffic, add the access profile to a virtual server.

Testing AAA high availability for supported authentication servers

To effectively test that high availability works for your authentication servers, you should have two servers that are accessible, where you can remove one of them from the network.

Note: *High availability is supported for these authentication server types only: RADIUS, Active Directory, LDAP, CRLDP, and TACACS+.*

If you configured a supported authentication server type to use a pool of connection servers, you can test the configuration using these steps.

1. Begin a `tcpdump` on the Access Policy Manager®, using a protocol analyzer, and scanning for packets destined for the specific port for your authentication server.
2. Log in to the virtual server with both servers active.
3. Using the `tcpdump` records, verify that the requests are being sent to the higher priority server.
4. Log out of the virtual server.
5. Disable the higher-priority server.
6. Log in to the virtual server again.
7. Verify that the request is being sent to the other server.
8. Log out again, re-enabling the server, and try one more time to verify that the new requests are being sent to the high priority server.

Example access policy for TACACS+ authentication and accounting

This is an example of an access policy with all the associated elements needed to authenticate and authorize users with TACACS+ authentication. Note that the server used for authentication can be different from the server used for TACACS+ accounting service.

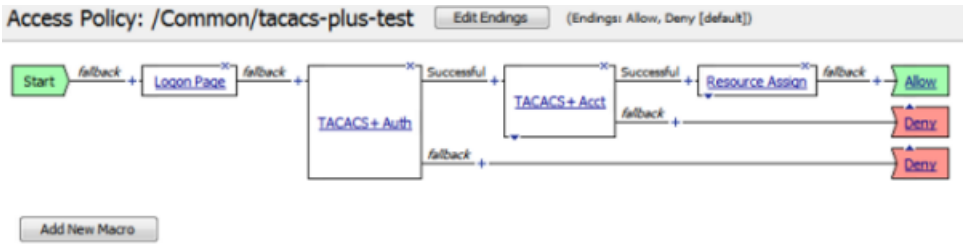


Figure 20: How TACACS Plus works

TACACS+ session variables for access policy rules

When the TACACS+ Auth (or TACACS+ Acct) access policy item runs, it populates session variables which are then available for use in access policy rules. The tables list the session variables for the TACACS+ access policy items and for a logon access policy item.

Session variables for TACACS+

Session Variable	Description
session.tacasplus.last.acct.start_date; session.tacasplus.last.acct.start_time	Provides TACACS+ accounting start time and date set by the accounting agent.
session.tacacsplus.last.acctresult	Allows the accounting agent to set the available values to either of the following values: • 0: Failed • 1: Succeeds
session.tacacsplus.last.errmsgs	Contains the error message string when the TACACS+ authentication or accounting fails.
session.tacacsplus.last.result	Sets to 1 when authentication succeeds, or 0 when it fails.

Common session variables

Session Variable	Description
session.logon.last.username	Provides user credentials. The username string is stored after encrypting, using the system's client key.
session.logon.last.password	Provides user credentials. The password string is stored after encrypting, using the system's client key.

TACACS+ authentication troubleshooting tips

You might run into problems with TACACS+ authentication in some instances. Follow these tips to try to resolve any issues you might encounter.

TACACS+ auth and query troubleshooting

Possible error messages	Possible explanations and corrective actions
`No AAA server associated with the agent`	Make sure that a valid TACACS+ server configuration is assigned to the agent (TACACS+ Auth or TACACS+ Acct) used in the access policy.
`Failure to connect to TACACS+ server`	Make sure that the TACACS+ server is up and running and reachable from the BIG-IP® system.
`Login incorrect`	Supplied user credentials are not valid.
`Invalid reply content, incorrect key`	Make sure that the shared encryption key configured on the TACACS+ server configuration matches with the key on the remote TACACS+ server.
`Invalid AUTHEN/START packet from server`	Indicates either the wrong keys or that the authentication action (LOGIN) is not supported on the server.
`Unacceptable authen method`	Indicates that the TACACS+ server does not support the authentication. Check the settings on the server.
`Unexpected failure return/legal status value from authentication function/Permission error`	Caused by internal errors on the remote TACACS+ server. Check the logs on the remote TACACS+ server and also the configuration.

Chapter 20

AAA High Availability and Upgrade

- *Upgrading an Access Policy Manager high availability failover pair*

Upgrading an Access Policy Manager high availability failover pair

To ensure that upgrading a failover pair is successful, make sure that the Local Traffic Manager active-standby units were configured correctly if you are migrating from a previous version.

Important: *During the upgrade, all users currently logged on to the system will have to log on again.*

1. Connect to a standby unit of a failover pair.
2. Upgrade the standby unit.
3. Press **Force offline** on the unit to trigger a failover to this newly upgraded unit.

   The newly upgraded unit will take over as the active unit.
4. Once the upgraded unit takes over as active, restart the upgraded unit.

   This extra step of additional restart is required to flush out any of the old sessions which may been introduced from the the previously active unit from an older version of the software.
5. Wait for the upgraded unit to come back up.
6. Once the upgraded unit becomes the active unit, bring the other unit back online by pressing **Release offline**.

   This unit is now the standby unit.
7. Upgrade the standby unit.

Chapter 21

Configuring Single Sign-On with Access Policy Manager

- *What is Single Sign-On?*

What is Single Sign-On?

Access Policy Manager® provides a Single Sign-On (SSO) feature that leverages the credential caching and credential proxying technology.

Credential caching and proxying is a two-phase security approach that asks users to enter their credentials once to access their secured web applications. By leveraging this technology, users request access to the secured back-end web server. After that occurs, Access Policy Manager creates a user session and collects the user identity based on the access policy. When the access policy completes successfully, the user identity is saved (cached) in a session database. Access Policy Manager subsequently reuses the cached identity to seamlessly log the user into the secured web applications, thus providing the user with a single sign-on experience.

The Single Sign-On (SSO) feature provides the following benefits:

- Eliminates the need to administer and maintain multiple user logins
- Eliminates the need for users to enter their credentials multiple times.

Chapter 22

Single Sign-On Methods

- *What are the supported SSO methods?*
- *Creating an HTTP Basic SSO configuration*
- *Creating an HTTP forms-based SSO configuration*
- *Creating an NTLMV1 SSO configuration*
- *Creating an NTLMV2 SSO configuration*

What are the supported SSO methods?

Access Policy Manager® supports the following SSO authentication methods.

SSO method	Description
HTTP Basic	Access Policy Manager uses the cached user identity and sends the request with the authorization header. This header contains the token `Basic` and the `base64-encoded` for the user name, colon, and the password.
HTTP Forms	Upon detection of the start URL match, Access Policy Manager uses the cached user identity to construct and send the HTTP form-based post request on behalf of the user.
HTTP Forms - Client Initiated	Upon detection of the request for logon page (URI, header, or cookie that is configured for matching the request), Access Policy Manager generates JavaScript code, inserts it into the logon page and returns the logon page to the client, where it is automatically submitted by inserted JavaScript. APM® processes the submission and uses the cached user identity to construct and send the HTTP form-based post request on behalf of the user.
HTTP NTLM Auth v1	NTLM employs a challenge-response mechanism for authentication, where the users can prove their identities without sending a password to the server.
HTTP NTLM Auth v2	NTLM employs a challenge-response mechanism for authentication, where the users can prove their identities without sending a password to the server. This version of NTLM is an updated version from NTLM v1.
Kerberos	This provides transparent authentication of users to Windows Web application servers (IIS) joined to Active Directory domain. It is used when IIS servers request Kerberos authentication; this SSO mechanism allows the user to get a Kerberos ticket and have Access Policy Manager present it transparently to the IIS application.
SAML	A SAML IdP service is a type of single sign-on (SSO) authentication service in Access Policy Manager that provides SSO authentication for external SAML service providers (SPs). You configure a SAML IdP service when you use a BIG-IP system as a SAML identity provider (IdP).

About the Single Sign-On configuration object

Access Policy Manager supports various SSO methods. Each method contains a number of attributes that you need to configure properly to support SSO.

Mis-configuring SSO objects for any of these authentication methods (HTTP Basic, NTLM v1 and v2, and Kerberos) could disable SSO for all authentication methods for a user's session when the user accesses a resource with the mis-configured object. The exceptions are Forms and Forms - Client Initiated, which are the only SSO methods that are not disabled when any other method fails due to a mis-configured SSO object.

Creating an HTTP Basic SSO configuration

With the HTTP Basic method of authentication, the SSO plug-in uses the cached user identity and sends the request with the authorization header. This header contains the Basic token and the base64-encoding of the user name, colon, and the password.

1. On the Main tab, click **Access Policy** > **SSO Configurations** > **HTTP Basic**.
   The SSO Configurations screen opens for HTTP Basic type.
2. Click **Create**.
   The New SSO Configuration screen opens.
3. In the **Name** field, type a name for the SSO configuration.
4. In the Credentials Source area, specify the credentials that you want cached for Single Sign-On.
5. In the SSO Method Configuration area, specify the relevant settings.
6. Click **Finished**.

HTTP Basic SSO configuration settings

These settings are available when you create an HTTP Basic SSO configuration.

General Properties settings for HTTP Basic SSO configuration

Setting	Value	Additional Information
General Properties	**Basic** or **Advanced**. Defaults to **Basic**.	Additional settings are available when you select **Advanced**.
Name	Name of the SSO configuration.	The name must begin with a letter, or underscore, and contain only letters, numbers, underscores, dashes, and periods. Avoid using global reserved words in the name, such as all, delete, disable, enable, help, list, none, show, or None.
Headers	Header name-value pairs to send with the SSO method.	Available when you select **Advanced** from the **General Properties** list.

Credentials Source settings for HTTP Basic SSO configuration

Setting	Value	Additional Information
Username Source	Specifies the user name to cache for single sign-on. Defaults to a session variable.	Supported session variable: `session.sso.token.last.username`
Password Source	Specifies the password to cache for single sign-on. Defaults to a session variable.	Supported session variable: `session.sso.token.last.password`

SSO configuration settings for HTTP Basic SSO configuration

Setting	Value	Additional Information
Username Conversion	This check box is clear by default.	Select the check box to convert the PREWIN2k/UPN user name input format to the format you want to use for SSO.

Setting	Value	Additional Information
		For example, convert `domain\username` or `username@domain` to `username`.

Creating an HTTP forms-based SSO configuration

With the HTTP forms method of authentication, upon detection of the start URL match, the SSO plug-in uses the cached user identity to construct and send the HTTP form-based POST request on behalf of the user.

1. On the Main tab, select **Access Policy** > **SSO Configurations** > **Forms**.
   The SSO Configurations screen opens for the form-based type.
2. Click **Create**.
   The New SSO Configuration screen opens.
3. In the **Name** field, type a name for the SSO configuration.
4. From the **Use SSO Template** list, select the template you want to use.
   The screen refreshes to show additional settings applicable to the specific template.
5. In the Credentials Source area, specify the credentials that you want cached for Single Sign-On.
6. If you selected **None** from the **Use SSO Template** list, fill in the relevant settings in the SSO Method Configuration area.

   Otherwise, these settings are taken from the template that you selected.
7. Click **Finished**.

HTTP Form SSO configuration settings

These settings are available when you create an HTTP form-based SSO configuration.

General Properties settings for HTTP form-based SSO configuration

Setting	Value	Additional Information
General Properties	**Basic** or **Advanced**. Defaults to **Basic**.	Additional settings are available when you select **Advanced**.
Name	Name of the SSO configuration.	The name must begin with a letter, or underscore, and contain only letters, numbers, underscores, dashes, and periods. Avoid using global reserved words in the name, such as all, delete, disable, enable, help, list, none, show, or None.
Use SSO Template		If you select **None**, you must fill in the SSO Method Configuration area. Otherwise, the SSO Method Configuration area is not available; settings are configured with data supplied by the template you select.
Headers	Header name-value pairs to send with the SSO method.	Available when you select **Advanced** from the **General Properties** list.

Credentials Source settings for HTTP form-based SSO configuration

Setting	Value	Additional Information
Username Source	Specifies the user name to cache for single sign-on. Defaults to a session variable.	Supported session variable: `session.sso.token.last.username`
Password Source	Specifies the password to cache for single sign-on. Defaults to a session variable.	Supported session variable: `session.sso.token.last.password`

SSO configuration settings for HTTP form-based SSO configuration

Setting	Value	Additional Information
Start URI	Defines the start URI value. HTTP form-based authentication executes for SSO if the HTTP request URI matches the start URI value.	Multiple start URI values in multiple lines can be entered for this attribute. Supported session variable: `start_uri`
Pass Through	If you select the **Enable** check box, cookies presented in the form propagate to the client browser. Defaults to cleared.	
Form Method	Defines the SSO authentication method : **GET** or **POST**. Defaults to **POST**.	If you specify **GET**, the SSO authentication method is an HTTP GET request.
Form Action	Defines the form action URL used for HTTP authentication request for SSO.	For example, `/access/oblix/apps/webgate/bin/webgate.dll`. If left blank, the original request URL is used for SSO authentication. Supported session variable: `form_action`
Form Parameter For User Name	Defines the parameter name of the logon user name.	For example, the user ID is specified as the attribute value if the HTTP server expects the user name in the form of `userid=`. Supported session variable: `form_parameter`
Form Parameter for Password	Defines the name of the logon password.	For example, `Pass` is specified as the attribute value if the HTTP server expects the password in the form of `Pass`.

Hidden Form Parameters/Values	Defines the hidden form parameters required by the authentication server logon form at your location.	Hidden parameters must be formatted as shown in this example:
		`param1 value1`
		`param2 value2`
		Separate each parameter name and value by a space. Each parameter must start on a new line.

Successful Logon Detection Match Type	Defines how Access Policy Manager detects whether	• **None** No check is made for authentication success. • **By Resulting Redirect URL** Authentication success is checked for by examining the redirect URL from

Setting	Value	Additional Information
	the user was successfully authenticated by the server. Defaults to **None**. You can select one option.	the HTTP response. Multiple values can be specified for this option. • **By Presence Of Specific Cookie** Authentication success is checked for by searching for the named cookie in the response. Supported session variable: `success_match_value`
Successful Logon Detection Match Value	Defines the value for the specific success detection type: the redirect URL or cookie name.	

Creating an NTLMV1 SSO configuration

The NTLM authentication method employs a challenge-response mechanism for authentication, where the users can prove their identities without sending a password to a server.

1. On the Main tab, click **Access Policy** > **SSO Configurations** > **NTLMV1**.
   The SSO Configurations screen opens for the NTLMV1 type.
2. Click **Create**.
   The New SSO Configuration screen opens.
3. Specify all relevant parameters.
4. Click **Finished**.

NTLMV1 SSO configuration settings

These configuration settings are available when you configure an NTLMV1 SSO method.

General Properties settings for NTLMV1 SSO configuration

Setting	Value	Additional Information
General Properties	**Basic** or **Advanced**. Defaults to **Basic** .	Additional settings are available when you select **Advanced**.
Name	Name of the SSO configuration.	The name must begin with a letter, or underscore, and contain only letters, numbers, underscores, dashes, and periods. Avoid using global reserved words in the name, such as all, delete, disable, enable, help, list, none, show, or None.
Headers	Header name-value pairs to send with the SSO method.	Displayed when you select **Advanced** from the **General Properties** list.

Credentials Source settings for NTLMV1 SSO configuration

Setting	Value	Additional Information
Username Source	Specifies the user name to cache for single sign-on. Defaults to a session variable.	Supported session variable: `session.sso.token.last.username`
Password Source	Specifies the password to cache for single sign-on. Defaults to a session variable.	Supported session variable: `session.sso.token.last.password`
Domain Source	Specifies the domain to cache for single sign-on. Defaults to a session variable.	Supported session variable: `session.logon.last.domain`

SSO configuration settings for NTLMV1 SSO configuration

Setting	Value	Additional Information
Username Conversion	Check box is cleared by default.	Select the check box to convert the PREWIN2k/UPN user name input format to the format you want to use for SSO. For example, convert `domain\username` or `username@domain` to `username`.
NTLM Domain	Specifies the location of the domain where all users and groups are authenticated. Defaults to a session variable.	Supported session variable: `session.logon.last.domain`

Creating an NTLMV2 SSO configuration

With this method of authentication, NTLM employs a challenge-response mechanism for authentication, where the users can prove their identities without sending a password to a server. This version of NTLM has been updated from version 1.

1. On the Main tab, click **Access Policy** > **SSO Configurations** > **NTLMV2**.
   The SSO Configurations screen opens for the NTLMV2 type.
2. Click **Create**.
   The New SSO Configuration screen opens.
3. In the **Name** field, type a name for the SSO configuration.
4. In the Credentials Source area, specify the credentials that you want cached for Single Sign-On.
5. In the SSO Method Configuration area, specify the relevant settings.
6. Click **Finished**.

NTLMV2 SSO configuration settings

These configuration settings are available when you configure an NTLMV2 SSO method.

General Properties settings for NTLMV2 SSO configuration

Setting	Value	Additional Information
General Properties	Basic or **Advanced**. Defaults to **Basic**.	Additional settings are available when you select **Advanced**.
Name	Name of the SSO configuration.	The name must begin with a letter, or underscore, and contain only letters, numbers, underscores, dashes, and periods. Avoid using global reserved words in the name, such as all, delete, disable, enable, help, list, none, show, or None.
Headers	Header name-value pairs to send with the SSO method.	Displayed when you select **Advanced** from the **General Properties** list.

Credentials Source settings for NTLMV2 SSO configuration

Setting	Value	Additional Information
Username Source	Specifies the user name to cache for single sign-on. Defaults to a session variable.	Supported session variable: `session.sso.token.last.username`
Password Source	Specifies the password to cache for single sign-on. Defaults to a session variable.	Supported session variable: `session.sso.token.last.password`
Domain Source	Specifies the domain to cache for single sign-on. Defaults to a session variable.	Supported session variable: `session.logon.last.domain`

SSO configuration settings for NTLMV2 SSO configuration

Setting	Value	Additional Information
Username Conversion	Check box is cleared by default.	Select the check box to convert the PREWIN2k/UPN user name input format to the format you want to use for SSO. For example, convert `domain\username` or `username@domain` to `username`.
NTLM Domain	Specifies the location of the domain where all users and groups are authenticated. Defaults to a session variable.	Supported session variable: `session.logon.last.domain`

Chapter 23

Form-Based Client-Initiated Single Sign-On Method

- *About form-based client-initiated SSO authentication*
- *Configuring form-based client-initiated SSO*
- *Form-based client-initiated SSO configuration examples*

About form-based client-initiated SSO authentication

With the HTTP form-based client-initiated method of authentication, when Access Policy Manager® detects the request for a logon page (URI, header, or cookie that is configured for matching the request), APM® generates JavaScript code, inserts it into the logon page, and returns the logon page to the client, where it is automatically submitted by the inserted JavaScript. APM processes the submission and uses the cached user identity to construct and send the HTTP form-based post request on behalf of the user.

Basic configuration of form-based client-initiated SSO

To create a form-based client-initiated SSO configuration object, you must configure at least one form and include at least one form parameter. A *form parameter* represents an input element on an HTML logon form, such as a form field for entering a user name or password, or, optionally, for entering a hidden form parameter.

Form-based client-initiated SSO configuration supports three sets of matching criteria that you can define.

- **Form Detection** (Required) Configures the SSO module to detect the HTTP request for the logon page by matching the HTTP URI, header, or cookie that you specify, and supports entry of multiple URIs. Requires data that is specific to the application. Form detection is successful when the request matches one of the configured items either partially or fully, depending on whether Request Prefix is enabled in Advanced Properties .
- **Form Identification** (Optional) Specifies how to detect the form within the HTTP body of the logon page. The default is form parameters, which enables identification of the logon form parameter fields based on the values entered for the form parameters in the General Properties. Alternatively, you can specify that the form be identified using other data present in the form, such as the ID, name, or action attributes, or the form order.
- **Logon Detection** (Optional) Configures the SSO module to detect whether logon was successful by checking for the presence of a cookie or a redirect URI. The default is **None** (logon detection is not performed).

The majority of web applications have a single logon page with one logon form. You need to define a single form for these applications. In less usual cases when an application has multiple logon pages with different logon forms, you need to create multiple forms, one for each logon page. If multiple logon pages use the same form, you need only one form with a list of URIs for all logon pages.

How does form-based client-initiated SSO authentication work by default?

This figure illustrates the default behavior of the form-based client-initiated SSO authentication method.

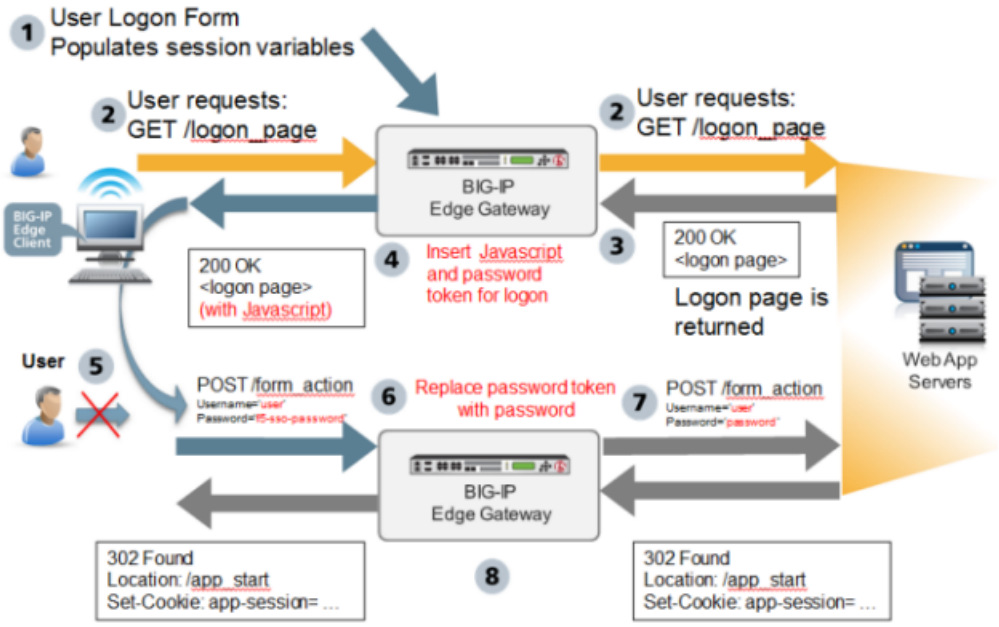


Figure 21: Form-based client-initiated SSO default behavior

1. The user logs on to Access Policy Manager® and APM® runs the access policy. This populates the session variables with the user credentials.
2. The user requests the application logon page. This GET request is passed to the application web server, verbatim.
3. The application web server replies with 200 OK and serves the logon page.
4. APM generates JavaScript and inserts it into the logon page before returning it to the user. The JavaScript assigns values to form parameters, as specified in the form configuration. The password parameter is assigned a password token rather than the actual user password.
5. The JavaScript runs on the client side. The logon page is not displayed to the user; user input is locked out. Without delay, the form is submitted using POST. The form parameters and their values, including user name and password token, are sent to APM.
6. APM then replaces the password token with the actual user password, as well as other form parameters specified in the form configuration with their configured values.
7. The POST, along with the real user credentials from step 1, is sent to the web server.
8. The application start page is served by the webserver, and sent to the client, verbatim. Optionally, APM performs detection of successful logon by examining HTTP response headers, looking for a cookie or redirect Location URI.

About advanced configuration options for form-based client-initiated SSO authentication

You can change some aspects of the form-based client-initiated SSO default behavior by configuring optional properties.

- You can change the default properties for form request and form submittal using advanced properties.
- You can change the automatically generated JavaScript code that is inserted into the logon page in one of three ways using the JavaScript Insertion options. You can replace it completely with custom code or add extra code to it by specifying the application JavaScript functions to call prior to submitting a logon form.

- You can configure the SSO module to automatically detect the application HTTP request that submits user credentials using Form Submit Detection. If you disable automatic detection, the SSO module instead detects form submittal by using an HTTP header, cookie, or HTTP URIs that you specify.

Configuring form-based client-initiated SSO

You can use the form-based client-initiated SSO method to create form-based SSO configurations. For example, you can use this SSO method to support web applications that run JavaScript in the browser and need to maintain application state during the login process. You can also use it to support web applications that present multiple login screens.

1. On the Main tab, click **Access Policy** > **SSO Configurations** > **Forms - Client Initiated**.
   The Forms - Client Initiated screen for SSO configurations opens.
2. Click **Create**.
   A screen, Create New Forms-Client Initiated Configuration, opens.
3. In the **SSO Configuration Name** field, type a name.
4. Select **Form Settings** from the left pane.
   Fields in SSO Configuration displays in the right pane.
5. Click **Create**.

   The **Create** button is not active until you complete they General Settings by typing a name for the SSO configuration.

   Note: *You must create at least one form to complete the SSO configuration.*

   The Create New Form Definition screen opens.
6. Type a name in the **Form Name** field.
7. Select **Form Parameters** from the left pane.
   Form Parameters displays in the right pane.
8. For each form parameter that you want to create, repeat these steps:
   a) Click **Create**.
      The Create New Form Parameter screen opens.
   b) In the **Form Parameter Name** field, type or select a name.
   c) In the **Form Parameter Value** field, type or select a value.
   d) Click **OK**.

   The screen closes, showing the Create New Form Definition window, which displays the new form parameter.
9. Click **Form Detection** from the left pane.
   The right pane displays required fields.
10. From the **Detect Form by** list, select an option and type required data.
    - **URI** Type a URI in the **Request URI** field.
    - **Cookie** Type a name in the **Cookie Name** field.
    - **Header** Type a name in the **Header Name** field.

    The **OK** button is available.
11. Click **OK**.

    You can create another form next or you can save the configuration.

    The new form is created. Its name displays on the Create New Forms-Client Initiated Configuration screen.

12. Click **OK**.
    The screen closes, displaying the Forms - Client Initiated screen for SSO Configurations.

The new form-based client-initiated SSO configuration is available for use.

Forms-based client-initiated SSO configuration settings

These settings are available when you create a form-based client-initiated SSO configuration.

General settings

Setting	Description
SSO Configuration Name	Specifies the name of the configuration. It must be unique.
SSO Description	Specifies a description. Optional.
Log Level	Specifies at what level of details the system logs. Valid values are listed. Defaults to **Notice**.

Form settings

Table 1: General Properties

Setting	Description
Form Name	Specifies the name of the form. It can be any name and need not match the actual name of the HTML form.
Form Description	Specifies an optional description of the form.

Table 2: Form Parameter Properties

Setting	Description
Form Parameter Name	Specifies the name of a form parameter.
Form Parameter Value	Specifies the value of the form parameter. This is usually the name of a session variable. The value could also be a literal string or a combination of strings and session variable names. ***Note:*** *If the session variable is not found when the SSO request is processed, the value of the corresponding POST parameter will be empty.*
Secure	Specifies whether the parameter is secure. Defaults to **No**.

Table 3: Form Detection

Setting	Description
Detect Form by	Specifies which element of the HTTP request headers is used to identify the application request for logon page: Cookie, Header, or URI. Defaults to URI.
Cookie	Specifies that the system identifies the form by the presence (default) or absence (configurable with Advanced Properties) of this cookie.

Setting	Description
Header	Specifies that the system identifies the form by the presence (default) or absence (configurable with Advanced Properties) of a header.
URI	Specifies that the system identifies the form by a successful match (default) or failed match (configurable with Advanced Properties) against one or multiple URIs.

Table 4: Form Identification

Setting	Description
Identify Form by	Specifies how the HTML logon form is found in the HTML body of the logon page. If there is more than one form on the logon page matching the criteria, the first match is used. Options are: • **ID Attribute-**Specifies that a form ID is used to find the form. • **Name Attribute**-Specifies that • **Action Attribute**-Specifies that • **Form Order**-Specifies that • **Form Parameters** (default)--Specifies that the form parameters, which have already been defined, are used to find the form. There is nothing more to configure.
Form ID	Specifies the form ID that is used to identify the form.
Form Name	Specifies the specific form name.
Form Action	Specifies the value of the action attribute.
Form Order	Specifies the relative order of the form on the logon page (starting from 1).

Table 5: Logon Detection

Setting	Description
Detect Login by	Specifies whether and how to detect a successful logon. Options are: • **Presence of Cookie** • **Redirect URI** • **None** (default)
Cookie Name	Specifies the cookie name that identifies successful logon.
Redirect URI	Specifies the redirect URI that identifies successful logon.

Table 6: Advanced Settings - Form Request

Setting	Description
Request Method	Specifies whether the request method is **GET** or **POST**. Defaults to **GET**.
Request Negative	When selected, specifies that the system detects the form that fails to match the criteria specified for Form Detection. The system then detects the form by the absence of the specific cookie or header, or by its failure to match the URIs. The default is cleared.

Setting	Description
Request Prefix	When selected, specifies that the system matches on a partial string. If this option is not selected, the match must be verbatim. The default is selected.

Table 7: Advanced Settings - Submit Request

Setting	Description
Submit Request Negative	When selected, specifies that the system detects the form that fails to match the criteria specified for Form Detection. The system then detects the form by the absence of the specific cookie or header or by its failure to match the URIs. The default is cleared.
Submit Request Prefix	When selected, specifies that the system matches on a partial string. If this option is not selected, the match must be verbatim. The default is selected.

Table 8: JavaScript Injection

Setting	Description
Injection Method	Specifies whether to use the default JavaScript that APM™ creates. Defaults to Auto. • Auto • Extra • Custom
Extra Javascript	Specifies more JavaScript to run at the end of the automatically generated JavaScript. ***Note:*** *Review the logon page source to determine whether any JavaScript functions are called on submit.*
Custom Javascript	Specifies JavaScript to run in place of the automatically generated JavaScript.

Table 9: Submit Detection

Setting	Description
Disable Auto detect submit	Defaults to **No.**
Scheme	Available when Disable Auto detect submit is set to Yes. Specifies how to detect submit. Options are: • **URI** • **Cookie** • **Header**

Header Settings

Setting	Description
Header Name	Name
Header Value	Value

Form-based client-initiated SSO configuration examples

Using the examples provided for various applications, you can quickly create form-based client-initiated SSO configurations.

DWA form-based client-initiated SSO example

This example lists settings and values for creating a form-based client-initiated SSO configuration for Domino Web Access (DWA).

Setting	Sample value
SSO Configuration Name	`ssov2-dwa`
Form Name	`testform`
• **Form Parameter Name** • **Form Parameter Value** • **Secure**	• `Username` • `%{session.sso.token.last.username}` • **No** (Default)
• **Form Parameter Name** • **Form Parameter Value** • **Secure**	• `Password` • `%{session.sso.token.last.password}` • **Yes**
Detect Form by	`URI`
Request URI	/
Identify Form by	`Name Attribute`
Form Name	`STLogonForm`
Detect Logon by	`Presence of Cookie`
Cookie Name	`DomAuthSessId`
Request Prefix	Not selected

Bugzilla form-based client-initiated SSO example

This example lists settings and values for creating a form-based client-initiated SSO configuration for Bugzilla.

Setting	Sample value
SSO Configuration Name	`ssov2-bugzilla`
Form Name	`tform`
• **Form Parameter Name** • **Form Parameter Value** • **Secure**	• `Bugzilla_login` • `%{session.sso.token.last.username}` • **No** (Default)
• **Form Parameter Name** • **Form Parameter Value** • **Secure**	• `Bugzilla_password` • `%{session.sso.token.last.password}` • **Yes**

Setting	Sample value
Detect Form by	**URI**
Request URI	`/`
Identify Form by	**ID Attribute**
Form ID	`mini_login_top`
Detect Logon by	**Presence of Cookie**
Cookie Name	`Bugzilla_logincookie`
Request Prefix	Not selected

Ceridian form-based client-initiated SSO example

This example lists settings and values for creating a form-based client-initiated SSO configuration for Ceridian.

Settings to configure form-based client-initiated SSO for Ceridian

Setting	Sample value
SSO Configuration Name	`ssov2_ceridian`
SSO Description	`sourcetimepro1.ceridian.com`
Form Name	`auth_form`
• **Form Parameter Name** • **Form Parameter Value** • **Secure**	• `ClientIDInput` • `%{session.logon.last.clientid}` • **No** (Default)
• **Form Parameter Name** • **Form Parameter Value** • **Secure**	• `SerialNumberInput` • `%{session.sso.token.last.username}` • **No** (Default)
• **Form Parameter Name** • **Form Parameter Value** • **Secure**	• `PasswordInput` • `%{session.sso.custom.last.password}` • **No** (Default)
Detect Form by	**URI**

Request URI	`/`
	`/sta.asp`
	`/ctagw/`
	`/ctagw/sta.asp`

Identify Form by	**Form Parameters**
Detect Logon by	**Redirect URI**
Redirect URI	`https://sourcetimepro1.ceridian.com/CTA660/cta.asp?RequestID=*`
Request Prefix	Not selected
Injection Method	**Custom**
Custom Javascript	See sample code that follows.
Disable Auto detect submit	**Yes**

Setting	Sample value
Scheme	**URI**
URI	`/sta.asp` `/ctagw/sta.asp`

Custom JavaScript

```
<script>
function checkInternetExplorerVersion()
// Returns 'true' if the version of Internet Explorer > 8
{
  var r = -1; // Return value assumes agreement.
  if (navigator.appName == 'Microsoft Internet Explorer')
  {
    var ua = navigator.userAgent;
    var re  = new RegExp("MSIE ([0-8]{1,}[\.0-9]{0,})");
    if (re.exec(ua) != null)
      r = parseFloat( RegExp.$1 );
  }
  return ( r==-1 ) ? true : false;
}
if (checkInternetExplorerVersion()) {
  document.body.style.visibility='hidden';
  document.body.style.display='none';
}
document.body.onkeydown=function(e){return false;};
function __f5submit() {
var __f5form = document.forms[0];
__f5form.SerialNumberInput.value='%{session.sso.token.last.username}';
__f5form.PasswordInput.value='%{session.sso.custom.last.password}';
__f5form.ClientIDInput.value='%{session.logon.last.clientid}';
f_submit();
}
if (window.addEventListener) {
  window.addEventListener('load',__f5submit,false);
} else if (window.attachEvent) {
  window.attachEvent('onload',__f5submit);
} else {
  window.onload=__f5submit;
}
</script>
```

Logon Page customization in access policy

Logon Page Agent (field 3):

- **Type:** `text`
- **Post Variable Name:** `clientid`
- **Session Variable Name:** `clientid`

Logon Page Input Field #3: `Company ID`

Variable Assign definition in access policy

```
session.sso.custom.last.password = expr { [mcget -secure
{session.sso.token.last.password}] }
```

Citrix 4.5 and 5 form-based client-initiated SSO example

This example lists settings and values for creating a form-based client-initiated SSO configuration for Citrix® versions 4.5 and 5.

Setting	Sample value
SSO Configuration Name	`sso_fbv2`
Form Name	`testform`
• **Form Parameter Name** • **Form Parameter Value** • **Secure**	• `domain` • `%{session.logon.last.domain}` • **No** (Default)
• **Form Parameter Name** • **Form Parameter Value** • **Secure**	• `user` • `%{session.sso.token.last.username}` • **No** (Default)
• **Form Parameter Name** • **Form Parameter Value** • **Secure**	• `password` • `%{session.sso.token.last.password}` • **Yes**
Detect Form by	**URI**
Request URI	`/Citrix/AccessPlatform/auth/login.aspx` `/Citrix/XenApp/auth/login.aspx`
Identify Form by	**Action Attribute**
Form Action	`login.aspx`
Detect Logon by	**Redirect URI**
Redirect URI	`*/Citrix/XenApp/site/default.aspx` `*/Citrix/AccessPlatform/site/default.aspx`

Devcentral form-based client-initiated SSO example

This example lists settings and values for creating a form-based client-initiated SSO configuration for Devcentral.

Settings to configure form-based client-initiated SSO for Devcentral

Table 10: Devcentral Configuration Example

Setting	Sample value
SSO Configuration Name	`ssov2_devcentral`
SSO Description	`devcentral.f5.com`
Form Name	`auth_form`
• **Form Parameter Name** • **Form Parameter Value** • **Secure**	• `dnn$ctr1093548$Login$Login_DNN$cmdLogin` • `Login` • **No** (Default)

Setting	Sample value
• **Form Parameter Name** • **Form Parameter Value** • **Secure**	• `dnn$ctr1093548$Login$Login_DNN$txtUsername` • `%{session.sso.token.last.username}` • **No** (Default)
• **Form Parameter Name** • **Form Parameter Value** • **Secure**	• `dnn$ctr1093548$Login$Login_DNN$txtPassword` • `%{session.sso.token.last.password}` • **Yes**
Detect Form by	`URI`
Request URI	`/Community/Login/tabid/1082224/Default.aspx` `/tabid/1082224/Default.aspx`
Identify Form by	`Form Parameters`
Detect Logon by	`Cookie`
Cookie Name	`authentication`
Injection Method	`Extra`
Extra Javascript	See sample code that follows.

Extra Javascript

```
WebForm_DoPostBackWithOptions(new
WebForm_PostBackOptions("dnn$ctr1093548$Login$Login_DNN$cmdLogin", "", true,
 "", "", false, false));
__f5form.enctype = 'application/x-www-form-urlencoded';
__f5form.encoding = 'application/x-www-form-urlencoded';
```

Google form-based client-initiated SSO example

This example lists settings and values for creating a form-based client-initiated SSO configuration for Google.

Setting	Sample value
SSO Configuration Name	`ssov2_google`
Description	`accounts.google.com`
Form Name	`form_auth`
• **Form Parameter Name** • **Form Parameter Value** • **Secure**	• `Email` • `%{session.sso.token.last.username}` • **No** (Default)
• **Form Parameter Name** • **Form Parameter Value** • **Secure**	• `Passwd` • `%{session.sso.token.last.password}` • **Yes**
Detect Form by	**URI**
Request URI	`/ServiceLogin`
Identify Form by	**Form Parameters**

Setting	Sample value
Detect Logon by	**Presence of Cookie**
Cookie Name	`SID`

Note: *For Internet Explorer 7 (and 8), disable the advanced setting* ***Display a notification about every script error****.*

Oracle Application Server form-based client-initiated SSO example

This example lists settings and values for creating a form-based client-initiated SSO configuration for Oracle 10g Release 2 (10.1.2).

Setting	Sample value
SSO Configuration Name	`ssov2_oracle`
Form Name	`tform`
• **Form Parameter Name** • **Form Parameter Value** • **Secure**	• `ssousername` • `%{session.sso.token.last.username}` • **No** (Default)
• **Form Parameter Name** • **Form Parameter Value** • **Secure**	• `password` • `%{session.sso.token.last.password}` • **Yes**
Detect Form by	`URI`
Request URI	`/sso/pages/login.jsp?site2pstoretoken=v1.2`
Identify Form by	`Form Parameters`
Detect Logon by	`Cookie`
Cookie Name	`SSO_ID`

OWA 2010 and 2007 form-based client-initiated SSO example

This example lists settings and values for creating a form-based client-initiated SSO configuration for Outlook Web App (OWA) 2010 and OWA 2007.

Table 11: OWA 2010 and OWA 2007 Configuration Example

Setting	Sample value
SSO Configuration Name	`ssov2-owa`
Form Name	`tform`
• **Form Parameter Name** • **Form Parameter Value** • **Secure**	• `username` • `%{session.sso.token.last.username}` • **No** (Default)
• **Form Parameter Name** • **Form Parameter Value**	• `password` • `%{session.sso.token.last.password}`

Setting	Sample value
• **Secure**	• **Yes**
Detect Form by	`URI`
Request URI	`/owa/auth/logon.aspx?replaceCurrent=1&url=` `/owa/auth/logon.aspx?url=`
Identify Form by	`Form Parameters`
Detect Logon by	`Presence of Cookie`
Cookie Name	`sessionid`
Injection Method	`Extra`
Extra Javascript	`clkLgn()`

OWA 2003 form-based client-initiated SSO example

This example lists settings and values for creating a form-based client-initiated SSO configuration for Outlook Web App (OWA) 2003.

Setting	Sample value
SSO Configuration Name	`ssov2-owa2003`
Form Name	`tform2003`
• **Form Parameter Name** • **Form Parameter Value** • **Secure**	• `username` • `%{session.sso.token.last.username}` • **No** (Default)
• **Form Parameter Name** • **Form Parameter Value** • **Secure**	• `password` • `%{session.sso.token.last.password}` • **Yes**
Detect Form by	`URI`
Request URI	`/exchweb/bin/auth/owalogon.asp?url=https://ata.bldg12.grpy.company.com/exchange/&reason=0`
Identify Form by	`Form Parameters`
Detect Logon by	`Presence of Cookie`
Cookie Name	`sessionid`

Perforce form-based client-initiated SSO example

This example lists settings and values for creating a form-based client-initiated SSO configuration for Perforce.

Setting	Sample value
SSO Configuration Name	`perforce-sso`
Form Name	`p4`
• **Form Parameter Name** • **Form Parameter Value**	• `u` • `%{session.sso.token.last.username}`

Setting	Sample value
• **Secure**	• **No** (Default)
• **Form Parameter Name** • **Form Parameter Value** • **Secure**	• `p` • `%{session.sso.token.last.password}` • **Yes**
Detect Form by	`URI`
Request URI	`/p4web`
Identify Form by	`Form Parameters`
Detect Logon by	`Presence of Cookie`
Cookie Name	`P4W8080`
Request Prefix	Not selected

Reviewboard form-based client-initiated SSO example

This example lists settings and values for creating a form-based client-initiated SSO configuration for Reviewboard.

Setting	Sample value
SSO Configuration Name	`reviewboard-sso`
Form Name	`rb_logon`
• **Form Parameter Name** • **Form Parameter Value** • **Secure**	• `username` • `%{session.sso.token.last.username}` • **No** (Default)
• **Form Parameter Name** • **Form Parameter Value** • **Secure**	• `password` • `%{session.sso.token.last.password}` • **Yes**
Detect Form by	`URI`
Request URI	`/account/login`
Identify Form by	`Form Parameters`
Detect Logon by	`Redirect URI`
Redirect URI	`*/dashboard`
Request Prefix	Not selected

SAP form-based client-initiated SSO example

This example lists settings and values for creating a form-based client-initiated SSO configuration for SAP.

Setting	Sample value
SSO Configuration Name	`ssov2_sap`
Form Name	`tform`
• **Form Parameter Name**	• `j_user`

Setting	Sample value
• **Form Parameter Value** • **Secure**	• `%{session.sso.token.last.username}` • **No** (Default)
• **Form Parameter Name** • **Form Parameter Value** • **Secure**	• `j_password` • `%{session.sso.token.last.password}` • **Yes**
• **Form Parameter Name** • **Form Parameter Value** • **Secure**	• `uidPasswordLogon` • `Log On` • **No** (Default)
Detect Form by	`URI`
Request URI	`/irj/portal`
Identify Form by	`Form Parameters`
Detect Logon by	`Presence of Cookie`
Cookie Name	`MYSAPSSOV2`
Request Prefix	Not selected

Salesforce form-based client-initiated SSO example

This example lists settings and values for creating a form-based client-initiated SSO configuration for Salesforce.

Setting	Sample value
SSO Configuration Name	`ssov2_salesforce`
Form Name	`auth_form`
• **Form Parameter Name** • **Form Parameter Value** • **Secure**	• `username` • `%{session.sso.token.last.username}` • **No** (Default)
• **Form Parameter Name** • **Form Parameter Value** • **Secure**	• `pw` • `%{session.sso.token.last.password}` • **Yes**
Detect Form by	`URI`
Request URI	`/`
Identify Form by	`Form Parameters`
Detect Logon by	`Cookie`
Cookie Name	`inst`
Injection Method	`Custom`
Custom Javascript	See sample code that follows.

Custom Javascript

```
<script>
      function checkInternetExplorerVersion()
      // Returns 'true' if the version of Internet Explorer > 8
      {
      var r = -1; // Return value assumes agreement.
      if (navigator.appName == 'Microsoft Internet Explorer')
      {
      var ua = navigator.userAgent;
      var re  = new RegExp("MSIE ([0-8]{1,}[\.0-9]{0,})");
      if (re.exec(ua) != null)
      r = parseFloat( RegExp.$1 );
      }
      return ( r==-1 ) ? true : false;
      }
      if (checkInternetExplorerVersion()) {
      document.body.style.visibility='hidden';
      document.body.style.display='none';
      }
      document.body.onkeydown=function(e){return false;};
      function __f5submit() {
      var __f5form = document.forms[0];
      __f5form.username.value='%{session.sso.token.last.username}';
      __f5form.password.value='f5-sso-token';
      ;
      var __f5action = __f5form.action;
      var __f5qsep = (__f5action.indexOf('?') == -1) ? '?' : '&';
      __f5form.action = __f5action + __f5qsep + 'f5-sso-form=auth_form';
      __f5form.Login.click();
      }
      if (window.addEventListener) {
      window.addEventListener('load',__f5submit,false);
      } else if (window.attachEvent) {
      window.attachEvent('onload',__f5submit);
      } else {
      window.onload=__f5submit;
      }
      </script>
```

Sharepoint 2010 form-based client-initiated SSO example

This example lists settings and values for creating a form-based client-initiated SSO configuration for Sharepoint.

Setting	Sample value
SSO Configuration Name	`ssov2_shp2010`
Form Name	`form_auth`
• **Form Parameter Name** • **Form Parameter Value** • **Secure**	• `ctl00$PlaceHolderMain$signInControl$UserName` • `%{session.sso.token.last.username}` • **No** (Default)
• **Form Parameter Name** • **Form Parameter Value** • **Secure**	• `ctl00$PlaceHolderMain$signInControl$password` • `%{session.sso.token.last.password}` • **Yes**
• **Form Parameter Name** • **Form Parameter Value** • **Secure**	• `ctl00$PlaceHolderMain$signInControl$login` • `Sign In` • **Yes**

Setting	Sample value
Detect Form by	`URI`
Request URI	`/_forms/default.aspx?ReturnUrl=`
Identify Form by	`Form Parameters`
Detect Logon by	`Cookie`
Cookie Name	`FedAuth`

Weblogin form-based client-initiated SSO example

This example lists settings and values for creating a form-based client-initiated SSO configuration for Weblogin.

Setting	Sample value
SSO Configuration Name	`ssov2-weblogin`
Form Name	`tform`
• **Form Parameter Name** • **Form Parameter Value** • **Secure**	• `user` • `%{session.sso.token.last.username}` • **No** (Default)
• **Form Parameter Name** • **Form Parameter Value** • **Secure**	• `pass` • `%{session.sso.token.last.password}` • **Yes**
• **Form Parameter Name** • **Form Parameter Value** • **Secure**	• `submit_form` • `Submit` • **No** (Default)
Detect Form by	`URI`
Request URI	`/sso/login.php?redir=`
Identify Form by	`Name Attribute`
Form Name	`theForm`
Detect Logon by	`Cookie`
Cookie Name	`issosession`

Yahoo form-based client-initiated SSO example

This example lists settings and values for creating a form-based client-initiated SSO configuration for Yahoo.

Setting	Sample value
SSO Configuration Name	`sso_yahoo`
SSO Description	`login.yahoo.com`
Form Name	`form_login`
• **Form Parameter Name** • **Form Parameter Value**	• `login` • `%{session.sso.token.last.username}`

Setting	Sample value
• **Secure**	• **No** (Default)
Detect Form by	URI
Request URI	/
Identify Form by	ID Attribute
Form ID	login_form
Detect Logon by	Cookie
Cookie Name	PH
Injection Method	Custom
Custom Javascript	See example custom Javascript that follows.
Disable Auto detect submit	Selected
Javascript	/config/login

Custom Javascript

```
<script>
 //Logon page will not be hidden in IE7/8.
 //This is workaround for the problem with JS method .focus()
 //"Can't move focus to the control because it is invisible, not enabled, or
 of a type that does not accept the focus."
function checkInternetExplorerVersion()
// Returns 'true' if the version of Internet Explorer > 8
{
  var r = -1; // Return value assumes agreement.
  if (navigator.appName == 'Microsoft Internet Explorer')
  {
    var ua = navigator.userAgent;
    var re  = new RegExp("MSIE ([0-8]{1,}[\.0-9]{0,})");
    if (re.exec(ua) != null)
      r = parseFloat( RegExp.$1 );
  }
  return ( r==-1 ) ? true : false;
}
if (checkInternetExplorerVersion()) {
  document.body.style.visibility='hidden';
  var inter = setInterval(function ()
  {
    var err = document.getElementsByClassName('yregertxt')[0];
    var wcl = document.getElementById('captcha_c');
    if (err) {
      document.body.style.visibility = 'visible';
      clearInterval(inter);
    }
    if (wcl) {
      if ( wcl.style.visibility == 'hidden') {
        document.body.style.visibility = 'visible';
        clearInterval(inter);
      }
    }
  }, 1000);
};
function __f5submit() {
var adv = document.getElementById('adFrame');
if (adv) adv.style.visibility='hidden';
var __f5form = document.forms[0];
if (__f5form.login)
  __f5form.login.value='%{session.sso.token.last.username}';
```

```
__f5form.passwd.value='%{session.sso.custom.last.password}';
__f5form[".save"].click();
}
if (window.addEventListener) {
  window.addEventListener('load',__f5submit,false);
} else if (window.attachEvent) {
  window.attachEvent('onload',__f5submit);
} else {
  window.onload=__f5submit;
}
</script>
```

Variable Assign definition used in access policy

```
session.sso.custom.last.password = expr { [mcget -secure
{session.sso.token.last.password}] }
```

Chapter 24

Kerberos Single Sign-On Method

- *About Kerberos SSO*
- *How does Kerberos SSO work in Access Policy Manager?*
- *Task summary for configuring Kerberos SSO*
- *Kerberos SSO configuration settings*
- *Kerberos SSO session variable list*
- *Tips for successfully deploying Kerberos SSO*

About Kerberos SSO

Access Policy Manager® provides seamless authentication to application servers (web servers) using Kerberos SSO. It is the only SSO method that can be used when authentication methods used by the access policy do not provide the user's password in clear text. Examples of such methods include client certificate authentication, NTLM authentication, or any other challenge/response authentication method where the password is not transmitted in clear text. To use Kerberos SSO, you must have Kerberos implemented in your environment, such as using Active Directory domain with IIS servers configured for Integrated Windows authentication.

How does Kerberos SSO work in Access Policy Manager?

You can leverage Kerberos SSO in the following ways:

- Using a virtual server with an access policy associated with it.
- Handling the SSO event through the use of Portal Access Resource. In this scenario, the Portal Access resource is assigned to the Access Policy and the virtual server attaches a rewrite profile.

Here is a typical scenario showing what occurs when Kerberos SSO is used if client certificate authentication is present:

1. When a user connects to the virtual server, Access Policy Manager validates the credentials and extracts the UPN from the certificate through the access policy.
2. When the client accesses an application that requires a Kerberos ticket, the UPN and the configured Kerberos SSO object are used to retrieve the ticket from Active Directory. The ticket is then cached for the particular client and presented to the application for access.

Important: *Under other circumstances, the access policy may not ask for credentials within the certificate, because, for example, a logon page may be present. In such a case, the user name supplied by the client is used at the UPN. Other factors, such as the use of other types of authentication methods, must be present in order to ensure that the credentials are valid in order to retrieve the Kerberos ticket.*

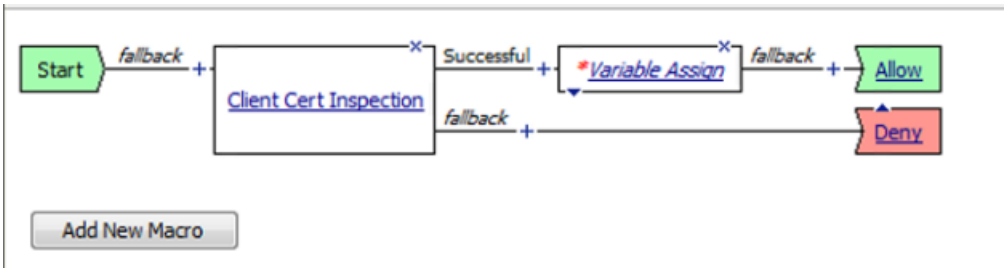


Figure 22: Example access policy for Kerberos SSO

Task summary for configuring Kerberos SSO

Access Policy Manager lets you configure for Kerberos SSO.

To set up this configuration, follow the procedures in the task list.

Task List

Setting up a delegation account to support Kerberos SSO
Creating a Kerberos SSO configuration in APM
Editing an access policy to support Kerberos SSO
Binding a Kerberos SSO object to an access profile
Attaching an access profile to a virtual server for Kerberos SSO

Setting up a delegation account to support Kerberos SSO

Before you can configure Kerberos SSO in Access Policy Manager®, you must create a delegation account in Active Directory. Note that for every server realm, you must create a delegation account in that realm.

1. Open the Active Directory Users and Computers administrative tool and create a new user account.
   The user account should be dedicated for delegation and the **Password never expires** setting enabled.
2. Run the `setspn` command-line tool for the user account from an elevated command prompt:
   The setspn command-line tool is available in Windows 2000 and Windows Server 2003 from Support Tools; it needs to be installed. The `setspn` tool is built into Windows Server 2008; it is available if you have the Active Directory Domain Services (AD DS) server role installed.
   a) Click the Windows Start button, right-click **Command Prompt**, and then click **Run as administrator**. An elevated command prompt opens.
   b) Type an `setspn` command.
   Example of command usage on Windows Server 2003: `setspn -A HTTP/bigcompany.lab.appnet.com domainname\userfordelegation`

   Note: *In Windows Server 2008, use the newly introduced `-L` option instead of `-A`. It validates the unicity of the service principal name (SPN) in the domain.*
3. Return to the Active Directory Users and Computers screen to open your account again.
   A Delegation tab should appear.
4. Click the Delegation tab.
5. Select **Trust this user for delegation to specified services only**.
6. Select **Use any authentication protocol**, and add all your services to the list under **Services to which this account can present delegated credentials**.
   Every service should have Service Type HTTP (or http) and host name of the pool member or web application resource host that you will use in your configuration.
7. Click **OK**.
   This creates the new delegation account.

Creating a Kerberos SSO configuration in APM

Before you create a Kerberos SSO configuration in Access Policy Manager®, create a delegation account in Active Directory.

To support Kerberos single sign-on authentication from APM®, you must create a Kerberos SSO configuration.

1. On the Main tab, click **Access Policy** > **SSO Configurations** > **Kerberos**.
   The SSO Configurations screen opens for Kerberos type.
2. Click **Create**.
   The New SSO Configuration screen opens.
3. In the **Name** field, type a name for the SSO configuration.
4. In the Credentials Source area, specify the credentials that you want cached for Single Sign-On.
5. In the **Kerberos Realm field,** type the name of the realm in uppercase.
   For example, `MY.HOST.LAB.MYNET.COM`
6. In the **Account Name** field, type the name of the Active Directory account configured for delegation.
7. In the **Account Password** and **Confirm Account Password** fields, type the delegation account password.
8. Click **Finished**.

Editing an access policy to support Kerberos SSO

After you create an access profile to support Kerberos SSO, you must edit the policy and add the appropriate agents.

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. From the list, select an access profile to which you want to add Kerberos SSO support.
   The properties screen for that access profile opens.
3. Click **Edit Access Policy for Profile *profile_name***.
   The visual policy editor opens the access policy in a separate screen.
4. Click the **(+)** icon anywhere in the access policy to add a new action item.
   A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
5. On an access policy branch, click the **(+)** icon to add an item to the access policy.
   A popup screen displays actions on tabs, such as General Purpose and Authentication, and provides a search field.
6. From Authentication, select **Client Cert Inspection**, and click **Add item**.
   A properties window opens.
7. Click **Save**.
   The properties screen closes and the visual policy editor displays.
8. Click the **(+)** icon anywhere in the access policy to add a new action item.
   A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
9. On an access policy branch, click the **(+)** icon to add an item to the access policy.
   A popup screen displays actions on tabs, such as General Purpose and Authentication, and provides a search field.
10. From **General Purpose**, select **Variable Assign** and click **Add item**.
    A properties window opens.
11. Click **Save**.
    The properties screen closes and the visual policy editor displays.

You have created an access policy to support Kerberos SSO.

The next step is to bind the SSO object to the access profile.

Binding a Kerberos SSO object to an access profile

Before beginning this task, configure an SSO object with Kerberos authentication or ensure that such an SSO object exists.

To bind a Kerberos SSO object to an access profile, add an SSO configuration (Kerberos SSO object) to it.

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. From the list, select an access profile to which you want to add Kerberos SSO support.
   The properties screen for that access profile opens.
3. Click the **SSO Auth/Domains** tab.
4. From the **SSO Configuration** list, select an SSO configuration with Kerberos authentication that you previously identified or configured.
5. Click **Update**.

Attaching an access profile to a virtual server for Kerberos SSO

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, in the **Address** field, type the IP address you want to use for the virtual server.

   The IP address you type must be available and not in the loopback network.
5. In the **Service Port** field, type a port number or select a service name from the **Service Port** list.
6. In the Access Policy area, from the **Access Profile** list, select the access profile that you configured for Kerberos SSO.
7. Click **Finished**.

Kerberos SSO configuration settings

These settings are available when you configure a Kerberos SSO method.

General Properties settings for Kerberos SSO configuration

Setting	Value	Additional Information
General Properties	Basic or **Advanced**. Defaults to Basic.	Additional settings are available when you select **Advanced**.
Name	Name of the SSO configuration.	The name must begin with a letter, or underscore, and contain only letters, numbers, underscores, dashes, and periods. Avoid using global reserved words in the

Setting	Value	Additional Information
		name, such as all, delete, disable, enable, help, list, none, show, or None.
Headers	Header name-value pairs to send with the SSO method.	Displayed when you select **Advanced** in the **General Properties** list.

Credentials Source settings for Kerberos SSO configuration

Setting	Value	Additional Information
Username Source	Specifies the user name to cache for single sign-on. Defaults to a session variable.	Supported session variable: `session.sso.token.last.username`
User Realm Source	Displays the session variable, if configured, that specifies the realm for the user. If the variable is set, it must contain the Kerberos realm for the user.	If this field is left empty or the variable does not exist or has no value, the user is assumed to be in the same Kerberos realm as the server. Supported session variable: `session.logon.last.domain`

SSO configuration settings for Kerberos SSO configuration

Setting	Value	Additional Information
Kerberos Realm	Specifies the realm of application servers, such as pool members or portal access resource hosts.	If servers are located in multiple realms, you must create a separate SSO configuration for each realm. Realm must be specified in uppercase letters or can be specified using the *session.logon.last.domain* session variable. ***Note:*** *The **KeepAlive** setting on your backend webserver must be enabled for Kerberos authentication to work properly.*
KDC	Specifies the IP Address or the host name of the Kerberos Key Distribution Center (KDC) (normally an Active Directory domain controller) for the server realm.	***Note:*** *KDC must be empty when the user realm is different from the server realm and in the case of multi-domain realms.* If KDC is empty, the KDC must be discoverable through DNS. For example, the BIG-IP system must be able to fetch SRV records for the server realm's domain, where the domain name is the same as the realm name. If the domain name is different from the realm name, it must be specified in the `/etc/krb5.conf` file. Kerberos SSO processing is fastest when KDC is specified by its IP address, slower when specified by host name, and, due to additional DNS queries, even slower when empty.
Account Name	Specify the name of the Active Directory account configured for delegation.	This account must be configured in the Kerberos realm (AD Domain) of the server. ***Note:*** *If servers are from multiple realms, each realm (AD Domain) must have its own delegation account.*
Account Password	Specifies the password for the delegation account	

Setting	Value	Additional Information
	specified in the **Account Name** field.	
Confirm Account Password	Verifies the password specified in the **Account Password** field.	
SPN Pattern	An optional field for modifying how the Service Principal Name (SPN) for the servers is constructed.	Leave this field empty unless you need non-standard SPN format. For example, HTTP/%s@REALM, where %s is replaced by the server host name discovered through reverse DNS lookup using the server IP address. When entering a string, replace REALM with an actual realm name (as specified in Kerberos Realm setting).
Ticket Lifetime	Represents the maximum ticket lifetime in minutes. Defaults to 600. Minimum is 10.	Should not be set higher than the value configured for the Active Directory delegation account (which defaults to 600). ***Note:*** *The actual lifetime can be less than the configured value by up to 1 hour because the user's ticket lifetime is the same as the Kerberos Ticket Granting Ticket (TGT) ticket lifetime.* The TGT for the delegation account specified in this configuration is obtained. A new TGT is fetched every time the latest TGT is older than one hour, but only when an SSO request is processed.
Send Authorization	Specifies when to submit the Kerberos ticket to application servers: **Always** or **On 401 Status Code**. Defaults to **Always**.	The Kerberos ticket is submitted in the HTTP Authorization header. The header value starts with the word Negotiate, followed by one space and a base64 encoded GSSIAPI token that contains the Kerberos ticket. If the request contains an Authorization header from the client browser, it is deleted. The options are defined here. • **Always** The Authorization header with a Kerberos ticket is inserted into every HTTP request whether or not it requires authentication; in other words, it is inserted preemptively. The Kerberos ticket GSSAPI representation uses KRB5 Kerberos 5 mechanism displays (OID 1.2.840.113554.1.2.2). Selecting **Always** results in the additional overhead of generating a Kerberos token for every request. Kerberos tickets are fetched for first request only for the user and then cached for up to the configured ticket lifetime, so that subsequent requests involve local processing only. • **On 401 Status Code** The BIG-IP system forwards the user's HTTP request to the web server first without inserting a new Authorization header; (any Authorization header from a browser is also deleted). If the server requests authentication by responding with a 401 status code, the BIG-IP system retries the request with the Authorization header. The Kerberos ticket GSSAPI representation uses the SPNEGO mechanism displays (OID 1.3.6.1.5.5.2).

Setting	Value	Additional Information
		Selecting **On 401 Status Code** results in an additional BIG-IP system and server request round trip when authentication is required for the request.
Username Conversion	Check box is cleared by default.	When the check box is selected, the PREWIN2k/UPN user name input format is converted to the format you want to use for SSO. For example, convert `domain\username` or `username@domain` to `username`.

Kerberos SSO session variable list

The following session variables are used by Kerberos SSO.

Session Variable name	Description
session.logon.last.domain	Contains the user's Kerberos realm. If unset, the user's realm is the same as the server's realm. The variable name is specified as *User Realm Source* in the SSO configuration and can be changed.
session.logon.last.username	Contains the user's login name. This can be extracted from the client certificate or supplied by the user on the login screen. The variable name is specified as *UsernameSource* in the SSO configuration and can be changed.
session.logon.last.username.sso.state	This is set to `1` internally when Kerberos SSO fails. When this variable is set, all subsequent requests are passed to the application server without applying SSO for the remainder of the user session. The variable name is constructed by appending *.sso.state* to the name specified in **Username Source**.

Tips for successfully deploying Kerberos SSO

If you run into problems with Kerberos SSO, follow these tips to try to resolve issues.

Microsoft® IIS servers

Only Microsoft® IIS servers are supported for pool members or web application resources. First, make sure the server computers running IIS are members of your AD Domain. Then follow these steps to enable Kerberos in IIS Manager:

1. From Active Directory administrative tool, right-click **Web Sites** and select **Properties**.
2. Select the Directory Security tab and in the Authentication and Access Control area click **Edit**.
3. Clear **Enable Anonymous Access**.
4. Check **Integrated Windows Authentication** and click **OK**. You might need to restart IIS or reboot the server for this to take effect.

Reverse DNS resolution

Kerberos SSO relies on reverse DNS resolution for determining the SPN (Service Principal Name) for each server host, such as a load balanced pool member or a web application resource host. Access Policy Manager should be configured to use DNS servers that have the appropriate forward and reverse DNS records for those servers. If DNS is lacking, those record host entries can be configured on the BIG-IP system.

DNS and KDC

Kerberos SSO relies on DNS for KDC discovery when KDC is not specified in an SSO configuration. The DNS server should have SRV records pointing to the KDC servers for the realm's domain. When DNS is not properly configured, or if the realm's DNS domain name is different from the realm's name, you can specify the KDC by adding a realm section to `/etc/krb5.conf` file on the BIG-IP system. For DNS discovery to work, the `dns_lookup_kdc` option in the `[libdefaults]` section of that file must be set to `true`.

Credential Caching

Kerberos uses credential caching to store Kerberos tickets. Access Policy Manager uses the websso process to maintain credential caches in memory, so restarting the websso process will discard all Kerberos tickets used for SSO.

Credential caching and high availability

The Ticket cache is not synchronized between units in high availability. Each user's Kerberos tickets are stored in a separate cache, where the name is constructed from the username, the user Kerberos realm, and the server Kerberos realm. Each cache contains a copy of the delegation account TGT for the server realm, a S4U2Self ticket for the user for the server realm, and multiple S4U2Proxy tickets for the servers. Once all tickets are fetched and stored in the cache, they remain there until they expire according to their lifetime. Processing of subsequent SSO requests should not require any more queries to the KDC. Since all tickets obtained from the same TGT have that TGT's lifetime, all tickets in the cache expire simultaneously. Each user's cache exists independently from the user's session. If the user has multiple concurrent or sequential sessions, the sessions all share the same cache, as long as it remains valid. The cache continues to exist even without any active sessions.

Maximum number of cache entries

The maximum number of cache entries is set to 20000. If the number is exceeded, it destroys older entries using the LRU algorithm. Delegation account TGT for each server realm is fetched when the first user request for that realm is processed. The TGT is cached and copied into every user's cache when the user accesses servers in that realm. If the TGT remaining lifetime becomes more than one hour shorter than the configured lifetime, the TGT is re-fetched. This is done to ensure that the new user's tickets are fetched with the initial lifetime closer to the configured value, and to avoid all tickets expiring at the same time, causing a performance impact.

Chapter 25

Single Sign-On and Multi-Domain Support

- *About multi-domain support for SSO*
- *How does multi-domain support work for SSO?*
- *Task summary for configuring domain support for SSO*

About multi-domain support for SSO

Access Policy Manager® (APM) provides a method to enable users to use a single login or session across multiple virtual servers in separate domains. Users can access back-end applications through multiple domains or through multiple hosts within a single domain, eliminating additional credential requests when they go through those multiple domains. With multi-domain support, you have the option of applying different SSO methods across different domains.

Attention: *To enable multi-domain support, all virtual servers must be on a single BIG-IP® system.*

These are some of the benefits that APM provides when you use it to set up multi-domain support for SSO.

- Users can sign out from all domains at once.
- Users can move from one domain to another seamlessly. This eliminates the need re-run the access policy, and thus maintains the established session for the user.
- Administrators can configure different cookie settings (Secure, Host/Domain and Persistent) for different domains, and for different hosts within same domain
- Administrators can set up multiple SSO configurations to sign users in to multiple back-end applications for a single APM® session

How does multi-domain support work for SSO?

The configuration process in which you successfully set up multi-domain support for SSO requires the following elements.

- An access profile that includes a set of participating domains.
- An SSO configuration associated with each of the domains. Additionally, a designated URL that specifies the primary authentication service is included in the access profile.

  Note: *The host name of the URL is a virtual server that provides an access policy to retrieve the credentials from the user. If an un-authenticated user reaches any domain specified in the domain group, a re-direct is first made to the primary authenticating service so that credentials are collected in order to establish a session.*

- A virtual server.
- The access profile associated with each of the virtual servers participating in the domain group.

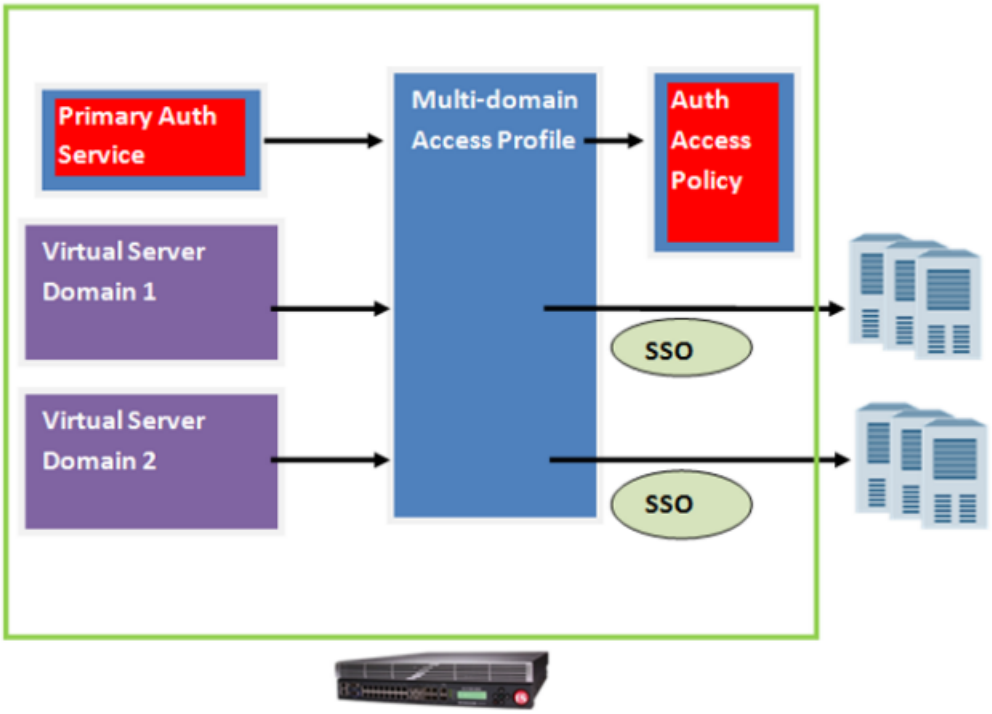


Figure 23: Configuration process for multi-domain support for SSO

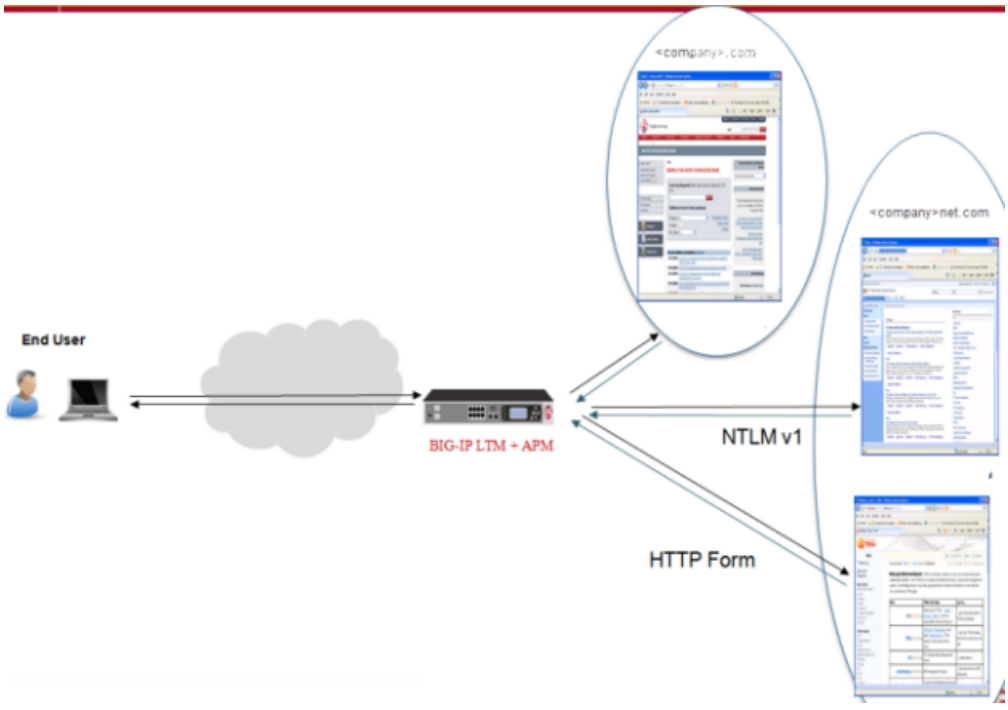


Figure 24: How multi-domain support for SSO works

Task summary for configuring domain support for SSO

Access Policy Manager SSO lets you configure either a single domain or multiple domains for SSO.

To set up this configuration, follow the procedures in the task list.

Task List

Configuring an access policy for SSO single domain support
Configuring an access policy for SSO multi-domain support
Creating a virtual server for SSO multi-domain support

Configuring an access policy for SSO single domain support

These steps apply only if you are setting up your access policy for SSO single domain support.

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. From the list, select an access profile in which you want to add SSO capability.
   The properties screen for that access profile opens.
3. On the menu bar, click **SSO/Auth Domains**.
4. For **Domain Mode**, select **Single Domain**.
5. For the **SSO Configuration** setting, select an available SSO configuration from the list to apply to your access policy.
6. Click **Update**.
7. On the menu bar, click **Access Policy**.
8. Click the name of the access profile for which you want to edit the access policy.
   The properties screen opens for the profile you want to edit.
9. Click **Edit Access Policy for Profile *profile_name***.
   The visual policy editor opens the access policy in a separate screen.
10. Click the **(+)** icon anywhere in the access policy to add a new action item.
    A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
11. For Predefined Actions, under General Purpose, select **SSO Credential Mapping**, and click **Add item**.
12. Click **Save**.
    You have now added SSO capability to your access policy.

Configuring an access policy for SSO multi-domain support

A user should be able to connect to any one of the virtual servers that participate in the domain group, and receive a request for credentials only once. Subsequent connections to other virtual servers within the domain group should not require the users to provide their credentials.

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. From the list, select an access profile to which you want to add SSO capability.
   The properties screen for that access profile opens.

3. On the menu bar, click **SSO/Auth Domains**.
4. For the **Domain Mode** setting, select **Multiple Domains**.
5. For **Primary Authentication URI**, type the URI the client is directed to, for example, `http://login.com`, in order to receive an Access Policy Manager session.

   Each domain that you configure indicates the domain the Access Policy Manager session (established by the primary authentication URI) is bound to.
6. In the Authentication Domain Configuration area, configure the **Cookie** setting by selecting **Host** or **Domain**, and typing the IP address for the host or, for domain, typing the fully qualified domain name.
7. Select **Cookie Options**. By default, **Secure** is selected.
8. From the **SSO Configuration** list, select the configuration that you want to associate to each host or domain. (Defaults to **None**.)
9. Click **Update**.

Creating a virtual server for SSO multi-domain support

For every domain, a virtual server should be configured.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. From **Access Profile**, select the profile you wish to attach to the virtual server.
5. Click **Finished**.

These steps should be repeated for every domain you specify in your access policy.

Chapter

26

Common Deployment Examples for Single Sign-On

- *Common use cases for Single Sign-On deployment*
- *Task summary for configuring web application over network access tunnel for SSO*

Common use cases for Single Sign-On deployment

You can deploy Single Sign-On in a variety of ways, depending on the needs within your networking environment. Deployment options include the following choices.

Use case deployment type	Description
For local traffic pool members	Deploy SSO for local traffic with pool members.
For web application access over network access	Deploy SSO through a network access with layered virtual servers.
For web applications	Deploy SSO so users can access their web applications. You can assign an SSO object as part of the web application resource item, or assign the object at the access profile level instead.

Task summary for configuring web application over network access tunnel for SSO

Using Access Policy Manager®, you can configure Single Sign-On for web applications access over a network access tunnel.

To set up this configuration, follow the procedures in the task list.

Task List

Configuring network access for SSO with web applications
Configuring network access properties
Configuring and managing the access profile using SSO
Configuring an HTTP virtual server for the network access
Configuring a layered virtual server for your web service
Configuring portal access resources for SSO

Configuring network access for SSO with web applications

1. On the Main tab, click **Access Policy** > **Network Access** > **Network Access List**.
   The Network Access List screen opens.
2. Click the **Create** button.
   The New Resource screen opens.
3. In the **Name** field, type a name for the resource.
4. To configure the general properties for the network resource, click **Properties** on the menu bar.
5. Configure your network client settings.
6. Click the **Finished** button.
   The Network Access configuration screen opens, and you can configure the properties for the network access resource.

Configuring network access properties

1. On the Main tab, click **Access Policy** > **Network Access** > **Network Access List**.
   The Network Access List screen opens.
2. Click the name to select a network access resource on the Resource List.
   The Network Access editing screen opens.
3. To configure the general properties for the network resource, click **Properties** on the menu bar.
4. To configure DNS and hosts settings for the network access resource, click **DNS/Hosts** on the menu bar.
5. To configure the drive mappings for the network access resource, click **Drive Mappings** on the menu bar.
6. To configure applications to start for clients that establish a network access connection with this resource, click **Launch Applications** on the menu bar.

Configuring and managing the access profile using SSO

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. Click **Create**.
   The New Profile screen opens.
3. In the **Name** field, type a name for the access profile.
4. Ensure that the **SSO Configuration** setting specifies **None**, and leave all the other settings at their default values.
5. Click **Finished**.
6. Click the name of the access profile for which you want to edit the access policy.
   The properties screen opens for the profile you want to edit.
7. On the menu bar, click **Access Policy**.
   The Access Policy screen opens.
8. Click **Edit Access Policy for Profile *profile_name***.
   The visual policy editor opens the access policy in a separate screen.
9. Add your objects to the access policy.

Configuring an HTTP virtual server for the network access

Create a virtual server to which the network access associates your access policy.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, in the **Address** field, type the IP address you want to use for the virtual server.

   The IP address you type must be available and not in the loopback network.

5. In the **Service Port** field, type a port number or select a service name from the **Service Port** list.
6. From the **Configuration** list, select **Advanced**.
7. From the **HTTP Profile** list, select **http**.
8. In the Configuration area, specify both **SSL Profile (Client)** and **SSL Profile (Server).**
9. From the **Source Address Translation** list, select **Auto Map**.
10. In the Access Policy area, select the **Access Profile** you created.
11. Click **Finished**.

Your user is now able to log on to Access Policy Manager and have full access to all their web services.

If you want to eliminate the need for users to enter their credential multiple times to access each web service, you now need to configure a layered virtual server for each of your web service.

Configuring a layered virtual server for your web service

Create a layered virtual server for every web service that the users access to eliminate the need for them to enter credential multiple times.

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. Create an access profile with a dummy default access policy.
3. Configure the access profile with the appropriate access policy, for example, **SSO Credential Mapping**.
4. Click **Update**.
5. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
6. Select the layered virtual server you created for your web service.
   The General Properties screen opens.
7. In the Configuration area for the **VLAN and Tunnel Traffic** setting, select **All VLANS and Tunnels** to ensure that the layered virtual server sends traffic from the network traffic to the network access tunnel interface.
8. Associate the dummy access profile you created by selecting it from the Access Profile list.
9. From the Configuration list, select **Advanced**, scroll down, and make sure that both **Address Translation**and **Port Translation** settings remained cleared.
10. Click **Update**.
    The users are now able to access multiple web services without having to enter their credential multiple times.

Configuring portal access resources for SSO

You can assign an SSO object as part of the portal access resource item. If you do not configure an SSO object at that level, you can use the SSO object at the access profile level instead.

1. On the Main tab, select **Access Policy** > **SSO Configurations**.
   The SSO Configurations list screen opens.
2. Click **Create**.
   The New SSO Configuration screen opens.
3. From the SSO Configurations by Type menu, choose an SSO type.
   A screen appears, displaying SSO configurations of the type you specified.

4. In the **Name** field, type a name for the SSO configuration.
5. Specify all relevant parameters.
6. Click **Finished**.
7. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
8. Click the name of the access profile for which you want to edit the access policy.
   The properties screen opens for the profile you want to edit.
9. From the **SSO Configurations** list, select an SSO configuration.
10. Click **Finished**.

Chapter 27

Introducing Access Policy Manager SAML Support

- *About SAML*
- *About SAML metadata*
- *About SAML single logout service*
- *About the benefits of using APM for SAML support*
- *When should I configure a BIG-IP system as a SAML IdP?*
- *When should I configure a BIG-IP system as a SAML service provider?*
- *Overview: Exchanging certificates among SAML entities*

About SAML

Security Assertion Markup Language (SAML) defines a common XML framework for creating, requesting, and exchanging authentication and authorization data among entities known as Identity Providers (IdPs) and Service Providers (SPs). This exchange enables single sign-on among such entities.

- *IdP* is a system or administrative domain that asserts information about a subject. The information that an IdP asserts pertains to authentication, attributes, and authorization. An assertion is a claim that an IdP makes about a subject.
- *Service Provider* is a system or administrative domain that relies on information provided by an IdP. Based on an assertion from an IdP, a service provider grants or denies access to protected services.

In simple terms, an IdP is a claims producer and a service provider is a claims consumer. An IdP produces assertions about users, attesting to their identities. Service providers consume and validate assertions before providing access to resources.

SAML 2.0 is an OASIS open standard. The SAML core specification defines the structure and content of assertions.

About SAML metadata

SAML metadata specifies how configuration information is defined and shared between two communicating entities: a SAML Identity Provider (IdP) and a SAML service provider. Service provider metadata provides information about service provider requirements, such as whether the service provider requires a signed assertion, the protocol binding support for endpoints (AssertionConsumerService) and which certificates and keys to use for signing and encryption. IdP metadata provides information about IdP requirements, such as the protocol binding support for endpoints (SingleSignOnService), and which certificate to use for signing and encryption.

About SAML single logout service

Single logout (SLO) service is a way to allow a user to terminate all sessions in an automatic manner without user intervention. A SAML Identity Provider (IdP) or the SAML service provider (SP) can initiate logout. The SAML IdP coordinates all logouts. When a SAML SP initiates a logout it contacts the SAML IdP to carry out the coordinated logout on its behalf.

Access Policy Manager® (APM®) supports SLO when all participating entities (SAML SPs and IdPs) support SLO. APM supports HTTP-POST binding for SLO messages.

About the benefits of using APM for SAML support

Access Policy Manager as a SAML Identity Provider (IdP)

When you use Access Policy Manager®(APM®) as a SAML IdP, APM can authenticate and generate assertions for a user who can then gain access to resources protected by SAML. APM provides SAML

assertions (claims) that service providers verify and consume. In this role, APM acts as an authentication server and provides single sign-on to service provider resources.

Access Policy Manager as a SAML Service Provider (SP)
When you use APM as a SAML service provider, APM consumes SAML assertions (claims) and validates their trustworthiness. After successfully verifying the assertion, APM creates session variables from the assertion contents. In an access policy, you can use these session variables to finely control access to resources and to determine which ACLs to assign. Based on the values of session variables, you can create multiple branches in the policy, assigning different resources and different ACLs on each branch. When it runs, the access policy follows a branch depending on the values of session variables.

Federation
APM systems operate with one another when one APM system is configured as an IdP and other APM systems are configured as service providers. This allows a user to authenticate with one APM acting as an IdP, and then use any number of APM systems, serving as service providers, without having to re-authenticate.

Metadata import and export
You can simplify SAML configuration using metadata files. When you use APM as an IdP, you can configure a SAML service provider by importing a metadata file that you obtain from the vendor. Similarly, when you use APM as a service provider, you can configure an IdP by importing a metadata file that you obtain from the vendor. You can export the metadata for APM as a SAML IdP from APM and import the metadata file into a service provider (or use information from the metadata file to configure the service provider). You can export the metadata for APM as a SAML service provider from APM and import the metadata file into an IdP (or use information from the metadata file to configure the IdP).

Templates
APM provide a few templates that you can use to create service provider connectors and a few that you can use to create IdP connectors with a minimal amount of typing.

Custom service providers and custom IdPs
In addition to configuring service provider connectos or an IdP connector from vendor metadata files or APM templates, you can configure custom service provider and IdP connectors.

IdP-initiated and service provider-initiated client connections
Access Policy Manager supports client connections that initiate at the IdP or at the service provider.

Signed assertions
By default, APM produces signed assertions. An assertion signed by the asserting party (the IdP) supports assertion integrity, authentication of the asserting party to a SAML relying party (a service provider), and, if the signature is based on the SAML authority's public-private key pair, non-repudiation of origin.

Encrypted assertions
For increased security, APM can optionally encrypt the entire assertion. APM supports encryption methods AES128, AES192 and AES256.

Support for SAML profiles
APM supports the Web Browser SSO profile with HTTP redirect and HTTP POST bindings.

When should I configure a BIG-IP system as a SAML IdP?

Configure a BIG-IP® system as a SAML identity provider (IdP) when you have one BIG-IP system and you want it to provide single sign-on authentication service for a group of external SAML service providers.

When should I configure a BIG-IP system as a SAML service provider?

Configure a BIG-IP® system as a SAML service provider when you have one BIG-IP system and you want it to protect services that are behind it, and direct users to an external SAML identity provider for authentication.

Overview: Exchanging certificates among SAML entities

For security purposes, each SAML service provider (SP) should have a certificate from the SAML Identity Provider (IdP) that manages identities for it; each IdP should have certificates from the SPs for which it manages identities.

Certificates on the BIG-IP system

Metadata normally includes a certificate. When you import metadata into a BIG-IP system from an external SP or an external IdP, the certificate that was included in the metadata is stored on the BIG-IP system. When you configure security-related settings on the BIG-IP system, you select certificates from the store.

If you do not have metadata that you can import from external SPs or IdPs, then you need to do one of the following:

- Get certificate files that you can import from the external systems into the BIG-IP system.
- Get certificate information from each external system that you can then paste into a user interface to create certificate files for them on the BIG-IP system.

BIG-IP system certificates on external systems

To get a certificate from the BIG-IP system, you can export it. You can potentially also get a certificate from a BIG-IP system by exporting SAML metadata for use on the external system.

When you export metadata from a BIG-IP system, it includes a certificate. However, when an external system requires signed metadata, the external system must already have a certificate from the BIG-IP system to validate the metadata.

Task Summary

Importing an SSL certificate
Exporting an SSL certificate

Importing an SSL certificate

Before you can perform this procedure, an SSL certificate must be available.

A BIG-IP® system requires a certificate from an external SAML service provider (SP) when the BIG-IP system is configured as a SAML Identity Provider (IdP) and must verify a signed authentication request from the SP. A BIG-IP system requires a certificate from an external IdP when the BIG-IP system is configured as an SP and must verify a signed authentication request from the IdP.

1. On the Main tab, click **System** > **File Management** > **SSL Certificate List**.
   The SSL Certificate List screen opens.

2. Click **Import**.
3. From the **Import Type** list, select **Certificate**.
4. For the **Certificate Name** setting, do one of the following:
   - Select the **Create New** option, and type a unique name in the field.
   - Select the **Overwrite Existing** option, and select a certificate name from the list.
5. For the **Certificate Source** setting, select **Upload File** and browse to select the certificate you obtained from the vendor.
6. Click **Import**.

The SSL certificate for the vendor is installed.

Exporting an SSL certificate

You export an SSL certificate when you configure a BIG-IP® system for SAML and you need a certificate from the BIG-IP system on an external SAML system.

1. On the Main tab, click **System** > **File Management** > **SSL Certificate List**.
   The SSL Certificate List screen opens.
2. Click the name of the certificate you want to export.
   The General Properties screen displays.
3. Click **Export**.
   The Certificate Export screen displays the contents of the certificate in the **Certificate Text** box.
4. To obtain the certificate, do one of the following:
   - Copy the text from the **Certificate Text** field, and paste it as needed into an interface on another system.
   - At the **Certificate File** option, click **Download filename** where filename is the name of the certificate file, such as mycert.crt.

Chapter

28

Using APM as a SAML IdP (SSO portal)

- *Overview: Configuring a BIG-IP system as IdP with an SSO portal*
- *Task summary*

Overview: Configuring a BIG-IP system as IdP with an SSO portal

This configuration supports:

- An SSO portal on the BIG-IP system configured as a SAML Identity Provider (IdP)
- Service providers (SPs) with the same or different requirements for assertion type and value and attributes (provided by the IdP)
- SP- and IdP-initiated connections

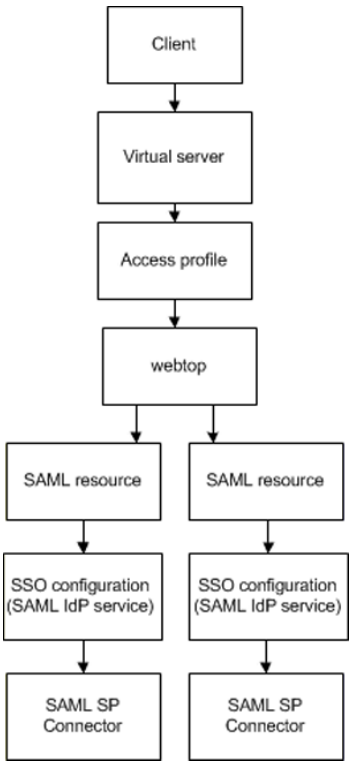
Configuration requirements to support IdP- and SP-initiated connections

When you want to use Access Policy Manager® as a SAML IdP and must support connections that start at the IdP or at service providers, you need to meet these configuration requirements:

- SAML IdP services: One for each SAML service provider.
- SAML SP connectors: One for each SAML service provider.
- SSL certificate: One for each SAML service provider, imported into the store on the BIG-IP® system.
- SAML resources: One for each SAML IdP service.
- A full webtop.
- An access profile.
- An access policy that:
  - Performs authentication
  - Assigns SAML resources and full webtop
- A virtual server that assigns the access profile

Configuration requirements are summarized in this diagram.

Figure 25: Configuration requirements to support IdP- and SP-initiated connections



About local IdP service

A *SAML IdP service* is a type of single sign-on (SSO) authentication service in Access Policy Manager® (APM®). When you use a BIG-IP® system as a SAML identity provider (IdP), a SAML IdP service provides SSO authentication for external SAML service providers (SPs). You must bind a SAML IdP service to SAML SP connectors, each of which specifies an external SP. APM responds to authentication requests from the service providers and produces assertions for them.

About SP connectors

A SAML service provider connector (an SP connector) specifies how a BIG-IP® system, configured as a SAML Identity Provider (IdP), connects with an external service provider.

What are the available ways I can configure a SAML SP connector?

You can use one or more of these methods to configure SAML service provider (SP) connectors in Access Policy Manager®.

- From metadata - Obtain a metadata file from the vendor and import it into Access Policy Manager. The advantage to this method is that the vendor provides the majority of all required data, including certificates. You can complete the configuration by simply typing a unique name for the SP connector, a very few additional required fields, and browsing to and importing the file. Access Policy Manager then configures the SP connector.
- From template - Use templates that Access Policy Manager provides for some vendors; for example, Google. The advantages to this method are that:

  - Most required data is included in the template
  - Additional required data is minimal. You can obtain it and certificates from the vendor

  After you select a template and type data into a few fields, Access Policy Manager configures the SP connector.
- Custom - Obtain information from the vendor and type the settings into the Configuration utility. To use this method, you must also obtain certificates from the vendor and import them into the BIG-IP® system. Use this method when a metadata file or a template for an SP connector is not available.

Task summary

Setting up a BIG-IP® system as a SAML identity provider (IdP) system involves two major activities:

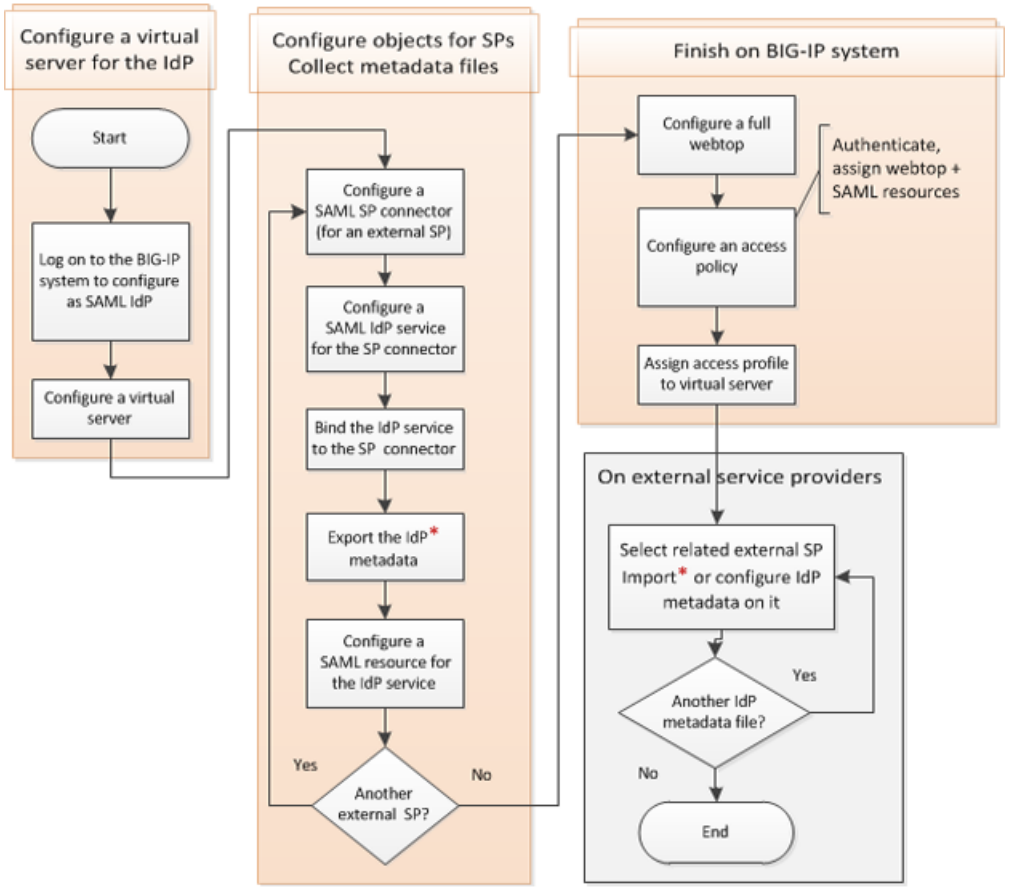
- First, you set up connection from the BIG-IP system to the external SAML service providers (SPs)
- Then, you set up connection from the external SAML SPs to the BIG-IP system

Task list

Creating a virtual server for a BIG-IP (as SAML IdP) system
Configuring SAML SP connectors
Configuring a full webtop
Configuring an access policy for a SAML SSO portal
Adding the access profile to the virtual server
Adding IdP metadata from APM to external SAML SPs

Flowchart: Configuration to support a SAML SSO portal

This flowchart illustrates the process for configuring a BIG-IP® system as a SAML identity provider (IdP) that provides an SSO portal.



Creating a virtual server for a BIG-IP (as SAML IdP) system

Before you start this task, configure a client SSL profile and a server SSL profile if you are going to create an SSL virtual server.

Note: *Access Policy Manager® supports using a non-SSL virtual server for the BIG-IP® system configured as a SAML Identity Provider (IdP). However, we recommend using an SSL virtual server for security reasons. The following procedures include steps that are required for configuring an SSL virtual server, such as selecting client and server SSL profiles, and setting the service port to HTTPS.*

Specify a host virtual server to use as the SAML IdP.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, select **Host** and in the **Address** field, type the IP address for the virtual server.
5. In the **Service Port** field, type `443` or select **HTTPS** from the list.
6. For the **HTTP Profile** setting, verify that the default HTTP profile, **http**, is selected.
7. For the **SSL Profile (Client)** setting, from the **Available** list, select the name of the Client SSL profile you previously created, and using the Move button, move the name to the **Selected** list.

8. For the **SSL Profile (Server)** setting, from the **Available** list, select the name of the Server SSL profile you previously created, and using the Move button, move the name to the **Selected** list.
9. Click **Finished**.

The virtual server for the BIG-IP system configured as an IdP now appears on the Virtual Server List. The virtual server destination is available for use in one or more SAML IdP service configurations.

Configuring SAML SP connectors

Obtain an SSL certificate from the SAML service provider (SP) and import it into the certificate store on the BIG-IP® system.

Configure information about a SAML service provider so that Access Policy Manager® (APM®) can act as a SAML Identity Provider (IdP) for it.

Note: *Configure one SAML SP connector for each external SAML service provider for which this BIG-IP system provides SSO authentication service.*

---

1. On the Main tab, click **Access Policy** > **SAML** > **BIG-IP as IdP**.
   The BIG-IP as IdP screen displays a list of SAML IdP services.
2. On the menu bar, click **External SP Connectors**.
   A list of SAML SP connectors displays.
3. Click **Create**.
   The Create New SP Connector screen opens.
4. In the **Service Provider Name** field, type a unique name for the SAML SP connector.
5. In the **SP Entity ID** field, type a unique identifier for the service provider.

   This is usually a unique URI that represents the service provider. You should obtain this value from the service provider.
6. Select **Endpoint Settings** from the left pane.

   The appropriate settings are displayed.
   a) In the **Assertion Consumer Service URL** field, type the URL where the IdP can send an assertion to this service provider.

      APM supports HTTP-POST binding to this service.
   b) (Optional) In the **Relay State** field, type a value.

      The relay state can be an absolute path, such as `/hr/index.html`; it can be a URL, such as `https://www.abc.com/index.html`; or, it can be anything that the service provider understands. The service provider uses this information to redirect users after they are authenticated. APM sends this value back to the service provider as part of the assertion response in the `RelayState` parameter.

      When the `RelayState` parameter is already part of the authentication request to the BIG-IP system, APM returns the value that was sent in the request. Otherwise, APM uses the value from this configuration.
7. Select **Security Settings** from the left pane.
   a) (Optional) In the Authentication Request sent by this device to IdP Settings area, select a setting from the **Will be signed** list. The default setting is **No**.

      The setting indicates whether this service provider signs the authentication requests it sends to the SAML IdP (this BIG-IP system).
   b) In the Assertion sent to SP by this device area, select **Must be signed** (default setting).

When this setting is selected, APM signs the assertion that it sends to this service provider. Clearing this setting is not recommended.

c) If this service provider requires an encrypted assertion from the IdP (this BIG-IP system), select **Must be encrypted** and select an **Encryption Type**.

APM supports AES128, AES192, and AES256 encryption types.

d) In the Certificate Settings area, select a certificate from the **SP's Certificate** list.

This device (BIG-IP system as IdP) uses the certificate to verify the signature of the authentication request from the SP. It also uses it to encrypt the assertion sent to the SP from this device.

8. Select **SLO Service Settings** from the left pane.

SLO stands for Single Logout.

a) (Optional) In the **Single Logout Request URL** field, type a URL where APM should send a logout request to this service provider when the BIG-IP system initiates a logout request.

b) In the **Single Logout Response URL** field, type a URL to which the SP should send a logout response to the BIG-IP system to indicate that single logout is complete.

Note: *APM supports HTTP-POST binding for the SLO service. For SLO to work, all entities (SPs and IdPs), must support SLO.*

APM creates a SAML SP connector. It is available to bind to a SAML IdP service.

Configuring a SAML IdP service for one SP connector

Configure a SAML Identity Provider (IdP) service for Access Policy Manager®, as a SAML IdP, to provide single sign-on authentication for one SAML service provider (SP).

Note: *Configure one IdP service for each SAML service provider.*

---

1. On the Main tab, click **Access Policy** > **SAML** > **BIG-IP as IdP**.
The BIG-IP as IdP screen displays a list of SAML IdP services.

2. Click **Create**.
The Create New IdP Service popup screen displays.

3. In the **IdP Service Name** field, type a unique name for the SAML IdP service.

4. In the **IdP Entity ID** field, type a unique identifier for the IdP (this BIG-IP® system). Include the URI that points to the virtual server with the BIG-IP system and a unique path.
For example, if you type `https://bigip-idp/idp`, `https://bigip-idp` should point to the virtual server you use for the BIG-IP system as a SAML IdP and `/idp` is a string that distinguishes one IdP from another when this BIG-IP system supports multiple SAML IdP services.

The path portion on the IdP Entity ID is not a physical location on the BIG-IP system.

5. Click **Assertion Settings** from the left pane.

The applicable settings display.

a) From the **Assertion Subject Type** list, select the type of subject for the IdP to authenticate.

b) From the **Assertion Subject Value** list, select the name of a session variable.

This variable, `%{session.logon.last.username}`, is generally applicable. Some session variables are applicable depending on the type of authentication that you use for your site.

c) Select the **Enable encryption of Subject** check box to encrypt the subject.
The **Encryption Strength** list becomes available.

d) From the **Encryption Strength** list, select a value.
Supported values are AES128, AES192, and AES256.

6. Click **SAML Attributes** from the left pane.
The SAML Attributes list displays. For each attribute that you want to include in the attribute statement, repeat these substeps.
   a) Click **Add**.
   b) Type a name and a value in the new row.
   Usually, the name is a fixed string; it can be a session variable. You can use a session variable for the value. This example shows using a fixed string for the name and a session variable for the value. Name: `user_telephonenumber` and value: `%{session.ad.last.attr.telephoneNumber}`.
   c) Select the **Encrypt** check box and select a value from the **Type** list.
   Select the check box to encrypt the attribute. Supported values for type are AES128, AES192, and AES256.
   d) Click **Update**.

7. Click **Security Settings** from the left pane.
   a) From the **This device's Assertion Signing Key** list, select the key from the BIG-IP system store.
   None is selected by default.
   b) From the **This device's Public Certificate** list, select the certificate from the BIG-IP system store.
   When selected, the IdP (the BIG-IP system) publishes this certificate to the service provider so that the service provider can verify the assertion. **None** is selected by default.

8. Click **OK**.
The popup screen closes. The new IdP service appears on the list.

APM creates a SAML IdP service. It is available to bind to an SP connector.

Binding a SAML IdP service to one SP connector

Bind a SAML Identity Provider (IdP) service and a SAML service provider (SP) connector so that the BIG-IP® system can provide authentication (SAML IdP service) to the external SAML service provider.

1. On the Main tab, click **Access Policy** > **SAML** > **BIG-IP as IdP**.
The BIG-IP as IdP screen displays a list of SAML IdP services.
2. Select a SAML IdP service from the list.
Select an IdP service that you configured for use with one particular SP connector only.
3. Click **Bind/Unbind SP Connectors**.
The screen displays a list of available SAML SP connectors.
4. Select the one SAML SP connector that you want to pair with this IdP service.
5. Select **OK**.
The screen closes.

The SAML SP connector that you selected is bound to the SAML IdP service.

Exporting SAML IdP metadata from APM

You need to convey the SAML Identity Provider (IdP) metadata from APM to the external service provider that uses this SAML IdP service. Exporting the IdP metadata for a SAML IdP service to a file provides you with the information that you need to do this.

1. On the Main tab, click **Access Policy** > **SAML** > **BIG-IP as IdP**.
   The BIG-IP as IdP screen displays a list of SAML IdP services.
2. Select a SAML IdP service from the table and click **Export Metadata**.
   A popup screen opens, with **No** selected on the **Sign Metadata** list.
3. For APM to sign the metadata, perform these steps:
   a) Select Yes from the **Sign Metadata** list.
   b) Select a key from the **Signing Key** list.
      APM uses the key to sign the metadata.
   c) Select a certificate from the **Signature Verification Certificate** list.
      APM exports the certificate to the metadata file. The system on which you import the metadata file can use the certificate to verify the metadata signature.
4. Select **OK**.
   APM downloads an XML file.

You must either import the IdP metadata XML file on the service provider system or use the information in the file to configure the SAML IdP on the service provider system.

Configuring a SAML resource and attaching a SAML IdP service

Configure a SAML resource to provide access to services on a SAML service provider when using APM as a SAML IdP.

Note: *Configure one SAML resource for each SAML IdP service that you have configured.*

---

1. On the Main tab, click **Access Policy** > **SAML** > **SAML Resources**.
   The SAML Resource list screen opens.
2. Click the **Create** button.
   The SAML Resource New Resource window opens
3. In the **Name** field, type a unique name for the SAML resource.
4. Do not clear the **Publish on Webtop** check box unless when you want to remove this resource from the webtop.
   When **Publish on Webtop** is selected, the SAML resource is displayed on a webtop where a user can initiate connection to an SP by clicking the icon. If you want users to initiate connection to this resource from an external SAML service provider only and do not want to show this resource on a webtop, clear the check box.
5. In the Configuration area from the **SSO Configuration** list, select the SAML IdP service that is bound to the SAML SP connector with the resources you want.
6. In the **Customization Settings for English** area in the **Caption** field, type a caption for this SAML resource.
7. Click **Finished**.
   The SAML resource is created and associated with a SAML IdP service that is bound to one external service provider.

Configuring a full webtop

A full webtop allows your users to connect and disconnect from a network access connection, portal access resources, SAML resources, app tunnels, remote desktops, and administrator-defined links.

1. On the Main tab, click **Access Policy** > **Webtops**.
2. Click **Create** to create a new webtop.
3. Type a name for the webtop you are creating.
4. From the **Type** list, select **Full**.
5. Click **Finished**.

The webtop is now configured, and appears in the list. You can edit the webtop further, or assign it to an access policy.

To use this webtop, it must be assigned to an access policy with an advanced resource assign action or with a webtop and links assign action. All resources assigned to the full webtop are displayed on the full webtop.

Configuring an access policy for a SAML SSO portal

Before you configure this access policy, configure an access profile without selecting an SSO configuration for it.

Configure an access policy so that the BIG-IP® system, as a SAML Identity Provider (IdP) can authenticate users using any non-SAML authentication type, and assign SAML resources and a webtop to the session.

Note: *This access policy supports users that initiate a connection at a SAML service provider or at the SAML IdP.*

---

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. In the Access Policy column, click the **Edit** link for the access profile you want to configure.
   The visual policy editor opens the access policy in a separate screen.
3. Click the **(+)** icon anywhere in the access policy to add a new action item.
   A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
4. On the Logon tab, select **Logon Page** and click the **Add Item** button.
   The Logon Page Agent properties screen opens.
5. Make any changes that you require to the logon page properties and click **Save**.
   The properties screen closes and the visual policy editor displays.
6. Add one or more authentication checks on the fallback branch after the **Logon Page** action.

   Select the authentication checks that are appropriate for application access at your site.
7. On a successful branch after an authentication check, assign SAML resources and a full webtop to the session.
   a) Click plus **[+]** on a successful branch after an authentication check.
      The Add Item window opens.
   b) On the Assignment tab, select the **Advanced Resource Assign** agent, and click **Add Item**.
      The Resource Assignment window opens.
   c) Click **Add new entry**.
      An **Empty** entry displays.

d) Click the **Add/Delete** link below the entry.
The screen changes to display resources on multiple tabs.
e) Select the SAML tab, then from it select the SAML resources that represent the service providers that authorized users can access.
f) Click **Update**.
The window changes to display the Properties screen, where the selected SAML resources are displayed.
g) Click the **Add/Delete** link below the entry.
The screen changes to display resources on multiple tabs.
h) Select the Webtop tab, then select a full webtop on which to present the selected resources.
You must assign a full webtop to the session even if you have configured all SAML resources to not publish on a webtop.
i) Click **Update**.
The window changes to display the Properties screen. The selected webtop and SAML resources are displayed.
j) Click **Save**.
The Properties window closes and the Access Policy window is displayed.

You have configured a webtop to display resources that are available from service providers and that an authorized user can access.

8. (Optional) Add any other branches and actions that you need to complete the access policy.
9. Change the Successful rule branch from **Deny** to **Allow** and click the **Save** button.
10. Click the **Apply Access Policy** link to apply and activate the changes to the access policy.
11. Click the **Close** button to close the visual policy editor.

You have an access policy that presents a logon page, authenticates the user, and assigns SAML resources and a full webtop on which to present them to the user.

Simple access policy for access to services on SAML service providers

To put the access policy into effect, you must attach it to a virtual server.

Adding the access profile to the virtual server

You associate the access profile with the virtual server so that Access Policy Manager® can apply the profile to incoming traffic.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
The Virtual Server List screen opens.
2. Click the name of the virtual server you want to modify.
3. In the Access Policy area, from the **Access Profile** list, select the access profile.
4. Click **Update** to save the changes.

Your access policy is now associated with the virtual server.

Adding IdP metadata from APM to external SAML SPs

To complete the agreement between Access Policy Manager® as the SAML IdP and a SAML service provider, you must configure IdP metadata at the service provider.

Note: *Complete this step on each SAML service provider for which an SP connector is bound to the SAML IdP service in APM®.*

Using the method that the vendor provides, either:

- Import the SAML IdP metadata file that you exported from APM for the SAML IdP service that this service provider uses.
- Or take information from the SAML IdP metadata file that you exported from APM for the SAML IdP service and add it to the service provider using the vendor's interface. Pay particular attention to the values for entityID, AssertionConsumerService, SingleSignOnService, and the certificate.

Note: *Regardless of the value of entityID in the metadata file, type an SSO URI that consists of the virtual server host and /saml/idp/profile/redirectorpost/sso. For example, if the host virtual server is https://Bigip-idp, type:* `https://Bigip-idp/saml/idp/profile/redirectorpost/sso`

Chapter 29

Using APM as a SAML IdP (no SSO portal)

- *Overview: Configuring a BIG-IP system as IdP for SP-initiated connections only*
- *Task summary*

Overview: Configuring a BIG-IP system as IdP for SP-initiated connections only

Note: *A configuration that allows users to initiate connection from service providers (SPs) only, works only when all service providers require the same assertion type, and value, and the same attributes from the IdP.*

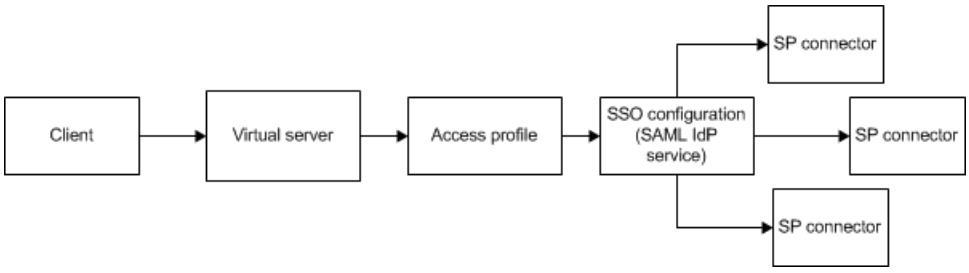
Configuration requirements for supporting SP-initiated connections only

For Access Policy Manager as a SAML identity provider (IdP) to support only connections that start at a service provider, you need to meet these configuration requirements:

- SAML IdP services: One.
- SAML SP connectors: One for each SAML service provider.
- SSL certificate and key: One set for each SAML service provider, imported into the store on the BIG-IP® system.
- An access profile.
- An access policy.
- A virtual server that assigns the access profile.

Configuration requirements are summarized in this diagram.

Figure 26: Configuration requirements for supporting SP-initiated connections



About local IdP service

A *SAML IdP service* is a type of single sign-on (SSO) authentication service in Access Policy Manager® (APM®). When you use a BIG-IP® system as a SAML identity provider (IdP), a SAML IdP service provides SSO authentication for external SAML service providers (SPs). You must bind a SAML IdP service to SAML SP connectors, each of which specifies an external SP. APM responds to authentication requests from the service providers and produces assertions for them.

About SP connectors

A SAML service provider connector (an SP connector) specifies how a BIG-IP® system, configured as a SAML Identity Provider (IdP), connects with an external service provider.

What are the available ways I can configure a SAML SP connector?

You can use one or more of these methods to configure SAML service provider (SP) connectors in Access Policy Manager®.

- From metadata - Obtain a metadata file from the vendor and import it into Access Policy Manager. The advantage to this method is that the vendor provides the majority of all required data, including certificates. You can complete the configuration by simply typing a unique name for the SP connector, a very few additional required fields, and browsing to and importing the file. Access Policy Manager then configures the SP connector.
- From template - Use templates that Access Policy Manager provides for some vendors; for example, Google. The advantages to this method are that:
  - Most required data is included in the template
  - Additional required data is minimal. You can obtain it and certificates from the vendor

  After you select a template and type data into a few fields, Access Policy Manager configures the SP connector.
- Custom - Obtain information from the vendor and type the settings into the Configuration utility. To use this method, you must also obtain certificates from the vendor and import them into the BIG-IP® system. Use this method when a metadata file or a template for an SP connector is not available.

Task summary

Setting up a BIG-IP® system as a SAML identity provider (IdP) system involves two major activities:

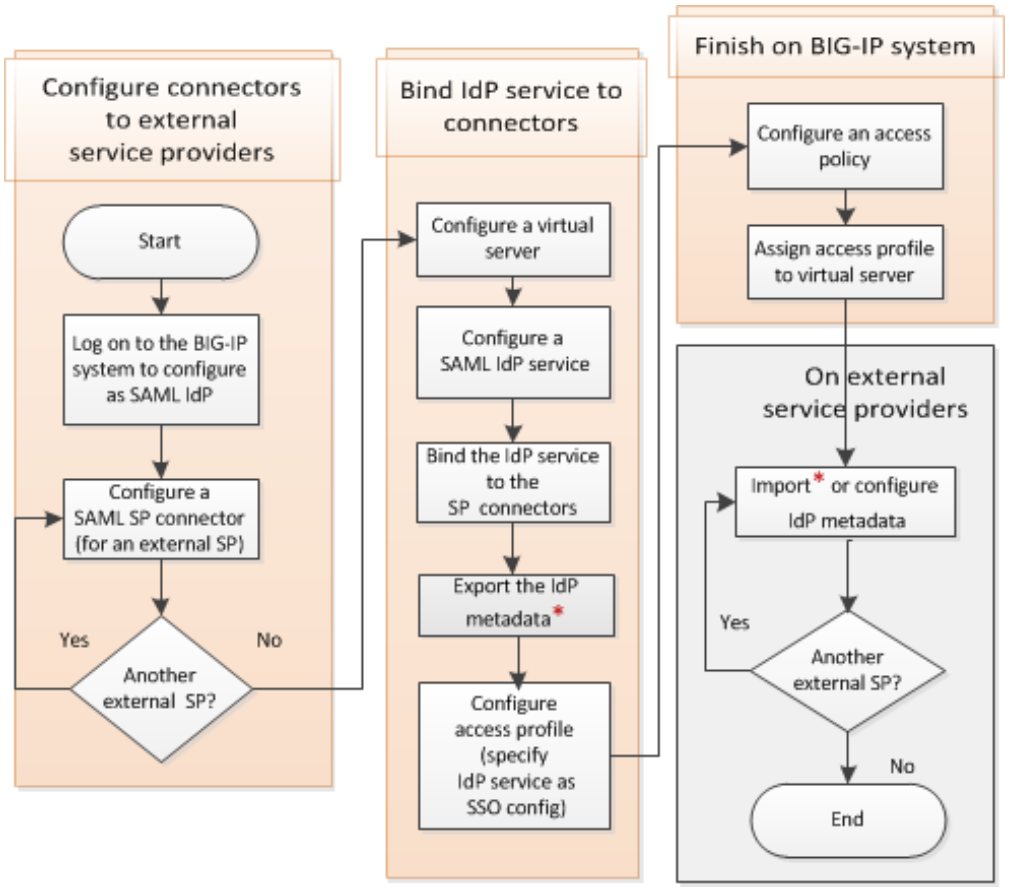
- First, you set up connection from the BIG-IP system to the external SAML service providers (SPs)
- Then, you set up connection from the external SAML SPs to the BIG-IP system

Task list

Creating a virtual server for a BIG-IP (as SAML IdP) system
Configuring SAML SP connectors
Configuring a SAML IdP service
Binding a SAML IdP service to multiple SP connectors
Exporting SAML IdP metadata from APM
Creating an access profile associated with the SAML IdP service
Configuring an access policy to provide authentication from the local IdP
Adding the access profile to the virtual server
Adding IdP metadata from APM to external SAML SPs

Flowchart: Configuration to support SP-initiated connections only

This flowchart illustrates the process for configuring a BIG-IP® system as a SAML identity provider (IdP) without providing an SSO portal.



Creating a virtual server for a BIG-IP (as SAML IdP) system

Before you start this task, configure a client SSL profile and a server SSL profile if you are going to create an SSL virtual server.

Note: *Access Policy Manager® supports using a non-SSL virtual server for the BIG-IP® system configured as a SAML Identity Provider (IdP). However, we recommend using an SSL virtual server for security reasons. The following procedures include steps that are required for configuring an SSL virtual server, such as selecting client and server SSL profiles, and setting the service port to HTTPS.*

Specify a host virtual server to use as the SAML IdP.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, select **Host** and in the **Address** field, type the IP address for the virtual server.
5. In the **Service Port** field, type `443` or select **HTTPS** from the list.
6. For the **HTTP Profile** setting, verify that the default HTTP profile, **http**, is selected.
7. For the **SSL Profile (Client)** setting, from the **Available** list, select the name of the Client SSL profile you previously created, and using the Move button, move the name to the **Selected** list.

8. For the **SSL Profile (Server)** setting, from the **Available** list, select the name of the Server SSL profile you previously created, and using the Move button, move the name to the **Selected** list.
9. Click **Finished**.

The virtual server for the BIG-IP system configured as an IdP now appears on the Virtual Server List. The virtual server destination is available for use in one or more SAML IdP service configurations.

Configuring SAML SP connectors

Obtain an SSL certificate from the SAML service provider (SP) and import it into the certificate store on the BIG-IP® system.

Configure information about a SAML service provider so that Access Policy Manager® (APM®) can act as a SAML Identity Provider (IdP) for it.

Note: *Configure one SAML SP connector for each external SAML service provider for which this BIG-IP system provides SSO authentication service.*

---

1. On the Main tab, click **Access Policy** > **SAML** > **BIG-IP as IdP**.
   The BIG-IP as IdP screen displays a list of SAML IdP services.
2. On the menu bar, click **External SP Connectors**.
   A list of SAML SP connectors displays.
3. Click **Create**.
   The Create New SP Connector screen opens.
4. In the **Service Provider Name** field, type a unique name for the SAML SP connector.
5. In the **SP Entity ID** field, type a unique identifier for the service provider.

   This is usually a unique URI that represents the service provider. You should obtain this value from the service provider.
6. Select **Endpoint Settings** from the left pane.

   The appropriate settings are displayed.
   a) In the **Assertion Consumer Service URL** field, type the URL where the IdP can send an assertion to this service provider.

      APM supports HTTP-POST binding to this service.
   b) (Optional) In the **Relay State** field, type a value.

      The relay state can be an absolute path, such as `/hr/index.html`; it can be a URL, such as `https://www.abc.com/index.html`; or, it can be anything that the service provider understands. The service provider uses this information to redirect users after they are authenticated. APM sends this value back to the service provider as part of the assertion response in the `RelayState` parameter.

      When the `RelayState` parameter is already part of the authentication request to the BIG-IP system, APM returns the value that was sent in the request. Otherwise, APM uses the value from this configuration.
7. Select **Security Settings** from the left pane.
   a) (Optional) In the Authentication Request sent by this device to IdP Settings area, select a setting from the **Will be signed** list. The default setting is **No**.

      The setting indicates whether this service provider signs the authentication requests it sends to the SAML IdP (this BIG-IP system).
   b) In the Assertion sent to SP by this device area, select **Must be signed** (default setting).

When this setting is selected, APM signs the assertion that it sends to this service provider. Clearing this setting is not recommended.

c) If this service provider requires an encrypted assertion from the IdP (this BIG-IP system), select **Must be encrypted** and select an **Encryption Type**.

APM supports AES128, AES192, and AES256 encryption types.

d) In the Certificate Settings area, select a certificate from the **SP's Certificate** list.

This device (BIG-IP system as IdP) uses the certificate to verify the signature of the authentication request from the SP. It also uses it to encrypt the assertion sent to the SP from this device.

8. Select **SLO Service Settings** from the left pane.

SLO stands for Single Logout.

a) (Optional) In the **Single Logout Request URL** field, type a URL where APM should send a logout request to this service provider when the BIG-IP system initiates a logout request.

b) In the **Single Logout Response URL** field, type a URL to which the SP should send a logout response to the BIG-IP system to indicate that single logout is complete.

Note: *APM supports HTTP-POST binding for the SLO service. For SLO to work, all entities (SPs and IdPs), must support SLO.*

APM creates a SAML SP connector. It is available to bind to a SAML IdP service.

Configuring a SAML IdP service

Configure a SAML Identity Provider (IdP) service for the BIG-IP® system, configured as a SAML IdP, to provide authentication service for SAML service providers (SPs).

Note: *Configure this IdP service to meet the requirements of all SAML service providers that you bind with it.*

---

1. On the Main tab, click **Access Policy** > **SAML** > **BIG-IP as IdP**.
The BIG-IP as IdP screen displays a list of SAML IdP services.
2. Click **Create**.
The Create New IdP Service popup screen displays.
3. In the **IdP Service Name** field, type a unique name for the SAML IdP service.
4. In the **IdP Entity ID** field, type a unique identifier for the IdP (this BIG-IP® system). Include the URI that points to the virtual server with the BIG-IP system and a unique path.
For example, if you type `https://bigip-idp/idp`, `https://bigip-idp` should point to the virtual server you use for the BIG-IP system as a SAML IdP and `/idp` is a string that distinguishes one IdP from another when this BIG-IP system supports multiple SAML IdP services.

   The path portion on the IdP Entity ID is not a physical location on the BIG-IP system.
5. Click **Assertion Settings** from the left pane.

   The applicable settings display.

   a) From the **Assertion Subject Type** list, select the type of subject for the IdP to authenticate.

   b) From the **Assertion Subject Value** list, select the name of a session variable.

   This variable, `%{session.logon.last.username}`, is generally applicable. Some session variables are applicable depending on the type of authentication that you use for your site.

c) Select the **Enable encryption of Subject** check box to encrypt the subject.
The **Encryption Strength** list becomes available.
d) From the **Encryption Strength** list, select a value.
Supported values are AES128, AES192, and AES256.

6. Click **SAML Attributes** from the left pane.
The SAML Attributes list displays. For each attribute that you want to include in the attribute statement, repeat these substeps.
a) Click **Add**.
b) Type a name and a value in the new row.
Usually, the name is a fixed string; it can be a session variable. You can use a session variable for the value. This example shows using a fixed string for the name and a session variable for the value. Name: `user_telephonenumber` and value: `%{session.ad.last.attr.telephoneNumber}`.
c) Select the **Encrypt** check box and select a value from the **Type** list.
Select the check box to encrypt the attribute. Supported values for type are AES128, AES192, and AES256.
d) Click **Update**.

7. Click **Security Settings** from the left pane.
a) From the **This device's Assertion Signing Key** list, select the key from the BIG-IP system store.
None is selected by default.
b) From the **This device's Public Certificate** list, select the certificate from the BIG-IP system store.
When selected, the IdP (the BIG-IP system) publishes this certificate to the service provider so that the service provider can verify the assertion. **None** is selected by default.

8. Click **OK**.
The popup screen closes. The new IdP service appears on the list.

APM creates a SAML IdP service. It is available to bind to SAML SP connectors. This service works with external service providers that share the same requirements for assertion settings and SAML attribute settings.

Binding a SAML IdP service to multiple SP connectors

Select a SAML Identity Provider (IdP) service and the SAML service provider (SP) connectors that use the service so that this BIG-IP® system can provide authentication (SAML IdP service) to external SAML service providers.

1. On the Main tab, click **Access Policy** > **SAML** > **BIG-IP as IdP**.
The BIG-IP as IdP screen displays a list of SAML IdP services.
2. Select a SAML IdP service from the list.
A SAML IdP service provides authentication service.
3. Click **Bind/Unbind SP Connectors**.
The screen displays a list of available SAML SP connectors.
4. Select only the SAML SP connectors that you want to use this service.
5. Click **OK**.
The screen closes.

The SAML IdP service is bound to the SAML service providers specified in the SAML SP connectors.

Exporting SAML IdP metadata from APM

You need to convey the SAML Identity Provider (IdP) metadata from APM to the external service providers that use the SAML IdP service. Exporting the IdP metadata for a SAML IdP service to a file provides you with the information that you need to do this.

1. On the Main tab, click **Access Policy** > **SAML** > **BIG-IP as IdP**.
   The BIG-IP as IdP screen displays a list of SAML IdP services.
2. Select a SAML IdP service from the table and click **Export Metadata**.
   A popup screen opens, with **No** selected on the **Sign Metadata** list.
3. For APM to sign the metadata, perform these steps:
   a) Select Yes from the **Sign Metadata** list.
   b) Select a key from the **Signing Key** list.
      APM uses the key to sign the metadata.
   c) Select a certificate from the **Signature Verification Certificate** list.
      APM exports the certificate to the metadata file. The system on which you import the metadata file can use the certificate to verify the metadata signature.
4. Select **OK**.
   APM downloads an XML file.

An XML file that contains IdP metadata is available.

Creating an access profile associated with the SAML IdP service

Use this procedure when this BIG-IP system, as a SAML Identity Provider (IdP), supports service provider-initiated connections only.

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. Click **Create**.
   The New Profile screen opens.
3. Type a name for the access profile.
4. In the SSO Across Authentication Domains (Single Domain mode) area, from the **SSO Configuration** list, select the name of the local SAML IdP service.
5. In the Language Settings area, add and remove accepted languages, and set the default language.
   A browser uses the highest priority accepted language. If no browser language matches the accepted languages list, the browser uses the default language.
6. Click **Finished**.

The access profile now shows up in the Access Profiles List.

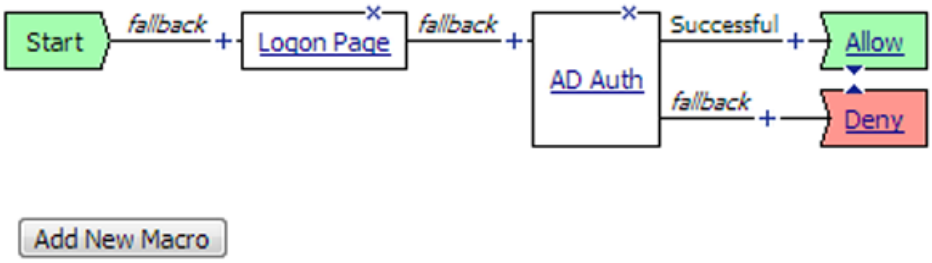
Configuring an access policy to provide authentication from the local IdP

Configure an access policy so that this BIG-IP® system, as a SAML Identity Provider (IdP) can provide authentication for SAML service providers.

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. In the Access Policy column, click the **Edit** link for the access profile you want to configure.
   The visual policy editor opens the access policy in a separate screen.
3. Click the **(+)** icon anywhere in the access policy to add a new action item.
   A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
4. On the Logon tab, select **Logon Page** and click the **Add Item** button.
   The Logon Page Agent properties screen opens.
5. Make any changes that you require to the logon page properties and click **Save**.
   The properties screen closes and the visual policy editor displays.
6. Add one or more authentication checks on the fallback branch after the **Logon Page** action.
   Select the authentication checks that are appropriate for application access at your site.
7. (Optional) Add any other branches and actions that you need to complete the access policy.
8. Change the Successful rule branch from **Deny** to **Allow** and click the **Save** button.
9. Click the **Apply Access Policy** link to apply and activate the changes to the access policy.
10. Click the **Close** button to close the visual policy editor.

You have an access policy that presents a logon page and authenticates the user..

Access policy to provide authentication for SAML service providers when this BIG-IP system is the IdP



To put the access policy into effect, you must attach it to a virtual server.

Adding the access profile to the virtual server

You associate the access profile with the virtual server so that Access Policy Manager® can apply the profile to incoming traffic.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the name of the virtual server you want to modify.

3. In the Access Policy area, from the **Access Profile** list, select the access profile.
4. Click **Update** to save the changes.

Your access policy is now associated with the virtual server.

Adding IdP metadata from APM to external SAML SPs

To complete the agreement between Access Policy Manager® as the SAML IdP and a SAML service provider, you must configure IdP metadata at the service provider.

Note: *Complete this step on each SAML service provider for which an SP connector is bound to the SAML IdP service in APM®.*

Using the method that the vendor provides, either:

- Import the SAML IdP metadata file that you exported from APM for the SAML IdP service that this service provider uses.
- Or take information from the SAML IdP metadata file that you exported from APM for the SAML IdP service and add it to the service provider using the vendor's interface. Pay particular attention to the values for entityID, AssertionConsumerService, SingleSignOnService, and the certificate.

  Note: *Regardless of the value of entityID in the metadata file, type an SSO URI that consists of the virtual server host and /saml/idp/profile/redirectorpost/sso. For example, if the host virtual server is https://Bigip-idp, type:* `https://Bigip-idp/saml/idp/profile/redirectorpost/sso`

Chapter

30

Using APM as a SAML Service Provider

- *About configuration requirements for APM as a SAML service provider*
- *Task summary*

About configuration requirements for APM as a SAML service provider

For Access Policy Manager® to act as a SAML service provider (SP), you must create this configuration.

- SAML SP service - One.
- SAML Identity Provider (IdP) connectors - One or more.
- An SSL certificate and key from each SAML IdP, imported into the store on the BIG-IP system.
- An access profile.
- An access policy that includes the SAML Auth agent.
- A virtual server that assigns the access profile.

About local SP service

A *SAML SP service* is a type of AAA service in Access Policy Manager® (APM®). It requests authentication from an external SAML Identity Provider (IdP) that is specified on APM in a SAML IdP connector. (You bind a SAML service provider (SP) service to one or more SAML IdP connectors.) APM requests authentication from an IdP and consumes assertions from it to allow access to resources behind APM.

About SAML IdP discovery

On a BIG-IP® system that you use as a SAML service provider (SP), you can bind an SP service to one or more SAML Identity Provider (IdP) connectors (each of which specifies an external IdP). When you bind an SP service to multiple IdP connectors, Access Policy Manager® chooses the correct IdP connector at run time through a filtering and matching process called IdP discovery.

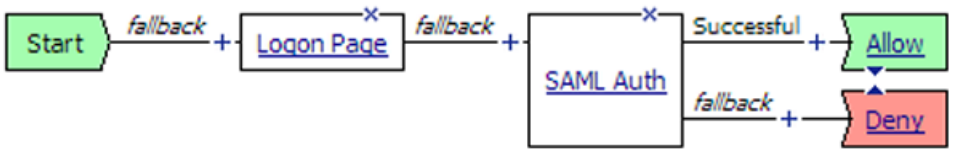
Scenario

You might bind multiple IdP connectors to an SP service on the BIG-IP system when you must provide services to different businesses and universities, each of which specifies an IdP to identify their users. When the user's information arrives at the SP service on the BIG-IP system, the SP service identifies the correct IdP and redirects the user to authenticate against that IdP before the SP service provides access to the service.

Note: *The SP service performs IdP discovery for a user only when the user initiates connection from an SP.*

Session variables and the typical access policy for BIG-IP system as SP

On a BIG-IP system configured as an SP, the typical access policy presents a logon page to the user. The Logon Page action populates session variables. You can customize the Logon Page action and affect session variable values. A SAML Auth action follows the logon page.



A SAML Auth action specifies an SP service. An SP service is an AAA service that requests authentication from an external IdP (specified in an IdP connector).

Session variables and SAML IdP discovery

Among multiple IdP connectors, the BIG-IP system must discover the correct external IdP with which to authenticate a user. For IdP discovery to work, you must specify matching criteria, a session variable name and value, for each IdP connector.

For example, users of a service might go to a particular landing page. When you bind the IdP connector, for the external IdP that serves those users, to the SP service, select the `%{session.server.landinguri}` session variable and supply a landing path value, such as, `/south*`. For users going to URLs such as `https://sp-service/southwest` and `https://sp-service/southeast`, the SP service selects the same IdP to authenticate them.

Logon Page action customization

These are some common customization examples for the Logon Page action.

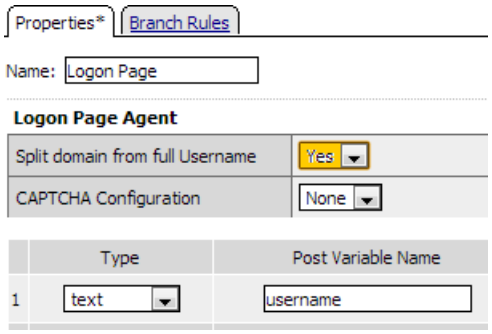


Figure 27: Setting the value of session.logon.last.domain variable to the domain name only

Select **Yes** for **Split domain from full Username**. The Logon Page agent takes the user name, such as joe@office.com, that was entered and creates the following session variables with these values.

Session Variable	Value
`%{session.logon.last.username}`	joe
`%{session.logon.last.domain}`	office.com
`%{session.logon.last.logonname}`	joe@office.com

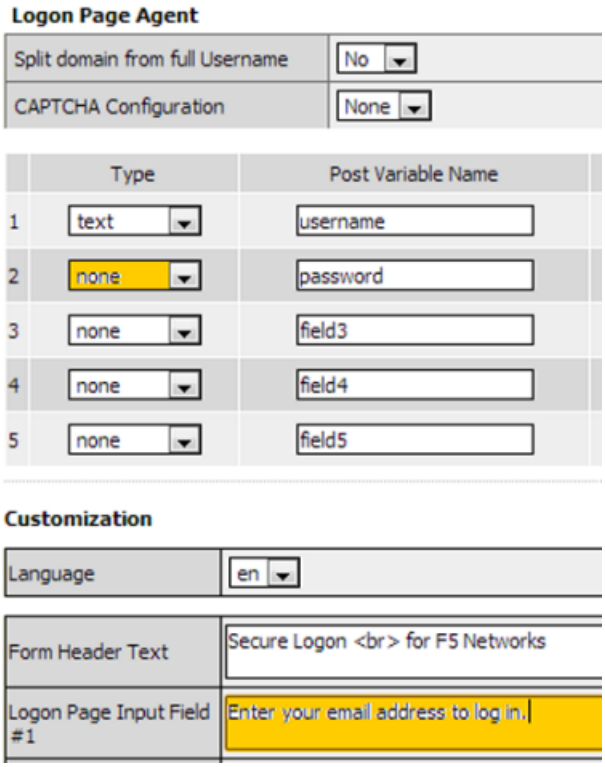


Figure 28: Obtaining and email address as the username

Change the prompt for the first field (username field). To omit the password, select Type none.

About IdP connectors

An IdP connector specifies how a BIG-IP® system, configured as a SAML service provider (SP), connects with an external SAML identity provider (IdP).

About methods for configuring SAML IdP connectors in APM

You can use one or more of these methods to configure SAML identity provider (IdP) connectors in Access Policy Manager® (APM®).

- From metadata - Obtain a metadata file from the vendor and import it into APM. The advantage to this method is that the vendor provides all required data, including the certificate. You can complete the configuration by simply typing a unique name for the identity provider, and browsing to and importing the file. APM imports the certificate to the BIG-IP® system and configures the SAML IdP connector.
- From template - Use templates that APM provides for some vendors. The advantages to this method are that:
  - Most required data is included in the template. (Note that the certificate is not included.)
  - Additional required data is minimal and is available from the vendor.

  APM configures the SAML IdP connector. You must obtain a certificate from the vendor and import it into the BIG-IP system.
- Custom - Research the identity provider requirements and type all settings into the Configuration utility. Use this method when a metadata file or a template for an identity provider is not available. APM

configures the SAML IdP connector. You must obtain a certificate from the vendor and import it into the BIG-IP system.

- IdP Automation - Provide files with cumulative IdP metadata on remote systems, then configure BIG-IP IdP automation to poll the files periodically and create IdP connectors and bind them to a specific service provider (SP) service.

Task summary

Setting up a BIG-IP® system as a SAML service provider (SP) involves two activities:

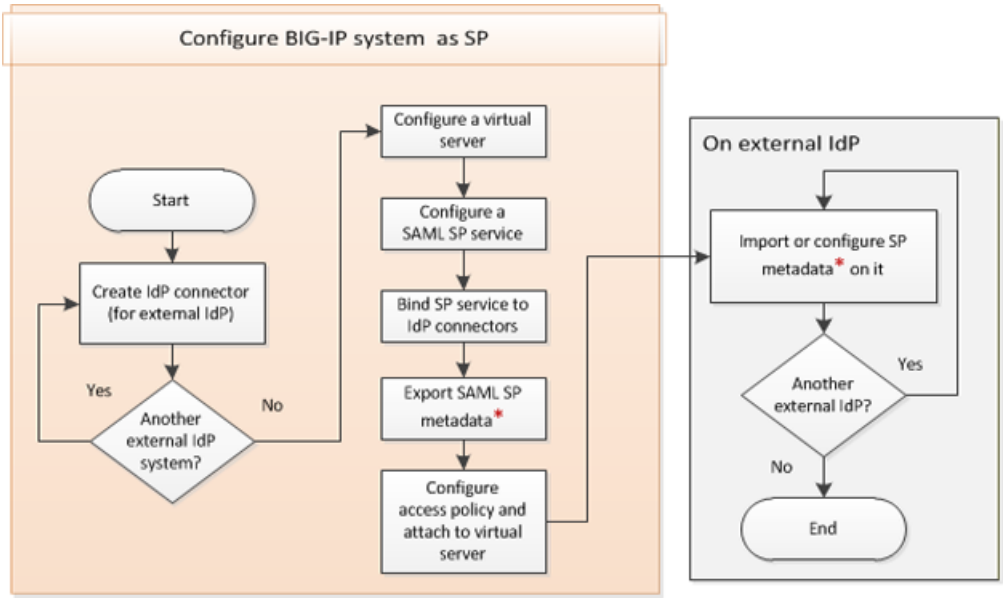
- First, you set up one BIG-IP system as a SAML service provider (SP) system
- Then, you go to one or more external SAML identity provider (IdP) systems and set up connectivity to the SP system

Task list

Configuring a custom SAML IdP connector
Creating a virtual server for a BIG-IP (as SAML SP) system
Configuring a SAML SP service
Binding a SAML SP service to SAML IdP connectors
Exporting SAML SP metadata from APM
Configuring an access policy to authenticate with an external SAML IdP
Adding the access profile to the virtual server
Adding SAML SP metadata from APM to an external SAML IdP

Flowchart: BIG-IP system as a SAML service provider configuration

This flowchart illustrates the process for configuring a BIG-IP® system as a SAML service provider (SP). In this configuration, the BIG-IP system relies on external SAML Identity Providers (IdPs).



Configuring a custom SAML IdP connector

Configure a SAML IdP connector so that Access Policy Manager® (APM®) (as a SAML service provider) can send authentication requests to this IdP, relying on it to authenticate users and to provide access to resources behind APM.

1. On the Main tab, click **Access Policy** > **SAML** > **BIG-IP as IdP**.
   The BIG-IP as IdP screen opens and displays a list of local IdP services.
2. On the menu bar, click **External IdP Connectors**.
   A list of SAML IdP connectors displays.
3. Click **Create** > **Custom**.
   The Create New IdP Connector screen opens.
4. In the **Name** field, type a unique name for the SAML IdP connector.
5. In the **IdP Entity ID** field, type a unique identifier for the IdP.
   This is usually a URI. Obtain this value from the vendor.
6. From the left pane, select **Endpoint Settings**.
   The screen changes to display the applicable settings.
7. In the **Single Sign On Service URL** field, type the location on the IdP where APM should send authentication requests.
8. (Optional) From the **Single Sign On Service Binding** field, select how APM should send authentication requests to the IdP:
   - POST (the default value)
   - Redirect
9. From the left pane, select **Assertion Settings**.
10. From **Identity Location**, select where to find the *principal* (usually, this is a user) to be authenticated:
    - **Subject** - In the subject of the assertion. This is the default setting.
    - **Attribute** - In an attribute. If selected, the **Identity Location Attribute** field displays and you must type an attribute name into it.

    Note: *If the assertion from the IdP does not include this attribute, the BIG-IP system (as SP) does not accept the assertion as valid.*
11. Select **Security Settings** from the left pane.
    a) (Optional) Select a setting from the **Must be signed** list. The default setting is **No**.
       The setting indicates whether APM must sign the authentication requests it sends to this IdP.
    b) From Certificate Settings, select a certificate from the **IdP's Assertion Verification Certificate** list.
       The BIG-IP system uses this certificate from the IdP to verify the signature of the assertion from the IdP. If the certificate from the IdP is not in the BIG-IP system store, obtain it and import it into the store. Then edit this IdP connector to select the certificate for it.
12. Select **SLO Service Settings** from the left pane.
    a) (Optional) In the **Single Logout Request URL** field, type a URL at the SAML Identity Provider where Access Policy Manager sends the logout request when a service provider initiates a logout.
    b) In the **Single Logout Response URL** field, type a URL for the SAML Identity Provider where APM sends the logout response when the IdP initiates the logout request.

Note: *APM supports HTTP-POST binding for the SLO service. For SLO to work, all entities (SPs and IdPs) must support SLO.*

13. Click **OK**.
    The screen closes.

APM creates a SAML IdP connector. It is available to be attached to a SAML SP service.

Creating a virtual server for a BIG-IP (as SAML SP) system

Before you start this task, configure a client SSL profile and a server SSL profile.

Note: *Access Policy Manager® supports using a non-SSL virtual server for the BIG-IP system (as SP). However, we highly recommend using an SSL virtual server for security reasons. The following procedure includes steps that are required for configuring an SSL virtual server. These are: selecting client and server SSL profiles and setting the service port to HTTPS.*

Specify a host virtual server to use as the SAML SP.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, select **Host** and in the **Address** field, type the IP address for the virtual server.
5. In the **Service Port** field, type `443` or select **HTTPS** from the list.
6. For the **HTTP Profile** setting, verify that the default HTTP profile, **http**, is selected.
7. For the **SSL Profile (Client)** setting, from the **Available** list, select the name of the Client SSL profile you previously created, and using the Move button, move the name to the **Selected** list.
8. For the **SSL Profile (Server)** setting, from the **Available** list, select the name of the Server SSL profile you previously created, and using the Move button, move the name to the **Selected** list.
9. Click **Finished**.

The virtual server for the BIG-IP system configured as an SP now appears on the Virtual Server List. The virtual server destination is available for use in a SAML SP service configuration.

Configuring a SAML SP service

Configure a SAML SP service for Access Policy Manager® to provide AAA authentication, requesting authentication and receiving assertions from a SAML IdP.

1. On the Main tab, click **Access Policy** > **SAML** > **BIG-IP as IdP**.
   The BIG-IP as IdP screen opens and displays a list of local IdP services.
2. Click **Create**.
   The Create New SAML SP Service screen opens.
3. In the **Name** field, type a unique name for the SAML SP service.
4. In the **Entity ID** field, type a unique identifier for the service provider that includes the URI that points to the virtual server with the BIG-IP system and a unique path.

For example, if you type `https://bigip-sp/sp`, then https:/bigip-sp points to the virtual server you use for APM as a SAML service provider and "/sp" is a unique string that APM uses to distinguish one service provider from another on this BIG-IP system.

Note: *The path is not a physical path on the BIG-IP system, but a string that distinguishes one SAML SP service from another when multiple SAML SP services are configured on this BIG-IP system.*

5. (Optional) In the **Relay State** field, type a value.

   The value can be an absolute path, such as hr/index.html or a URI, such as https://www.abc.com/index.html. It is where the service provider redirects users after they are successfully authenticated and have been allowed by the access policy. When APM receives the relay state from the Identity Provider in addition to assertion, then it uses the value received from the IdP to redirect the user. Otherwise, APM uses the value from this configuration.

6. From the left pane, select **Security Settings**.

   The screen changes to display the applicable settings.

   a) Select **Signed Authentication Request** if you want this BIG-IP system to send signed authentication requests to the SAML IdP.

   b) Select **Want Encrypted Assertion** if this BIG-IP system requires encrypted assertions from the SAML IdP.

   c) Select **Want Signed Assertion** if the BIG-IP service provider system requires signed assertions from the SAML IdP.

      This is selected by default. It is recommended that it be selected.

   d) From **SP's Authentication Signing/Assertion Decryption Private Key**, select a key from the BIG-IP system store on this device.

      You can select a private key only when you select at least one of these check boxes: **Signed Authentication Request** and **Want Encrypted Assertion**. APM uses this private key to sign the authentication request to the IdP and to decrypt an encrypted assertion from the IdP.

   e) From **SP Certificate**, select a certificate.

      APM includes this certificate in the SAML SP metadata that you export. After the SAML SP metadata is imported on the IdP, the IdP can use this certificate to verify a signed authentication request and to encrypt an assertion.

7. Click **OK**.
   The screen closes.

APM creates the SAML SP service. It is available to bind to SAML IdP connectors and to export to a metadata file.

Binding a SAML SP service to SAML IdP connectors

Select a SAML SP service and bind one or more SAML IdP connectors to it so that this device (BIG-IP system as a SAML service provider) can request authentication from the appropriate external IdP.

Note: *If you bind this SP service to more than one IdP connector, you must configure matching criteria for each IdP connector. When users initiate connections at service providers, the BIG-IP system uses matching criteria to identity the correct IdP among many using SAML IdP discovery.*

1. On the Main tab, click **Access Policy** > **SAML** > **BIG-IP as IdP**.
   The BIG-IP as IdP screen opens and displays a list of local IdP services.
2. Select a SAML SP service from the list.

3. Click **Bind/Unbind IdP Connectors**.
   A pop-up screen displays a list of any IdP connectors that are associated with this SP service.
4. To add an SAML IdP connector to the list, click **Add New Row**.
5. To bind only one IdP connector with this SP service, complete the configuration:
   a) Select a connector from the **SAML IdP Connectors** list in the new row.
      When you bind only one IdP connector to an SP service, you do not need to fill in the **Matching Source** and **Matching Value** fields.
   b) Click the **Update** button.
      The configuration is not saved until you click **OK**.
   c) Click **OK**.
      APM saves the configuration. The screen closes.
6. To bind multiple IdP connectors with this SP service, complete the configuration:
   a) Select a connector from the **SAML IdP Connectors** list in the new row.
   b) In the **Matching Source** column, select or type the name of a session variable.
      Use a session variable only if it is populated in the access policy before the SAML Auth action.
      For example, select `%{session.server.landinguri}` or type `%{session.logon.username}`.
   c) In the **Matching Value** column, type a value.
      The value can include the asterisk (*) wild card.
      For example, type `*hibb*` or `south*` .
   d) Click the **Update** button.
      The configuration is not saved until you click **OK**.
   e) To add other IdP connectors, start by clicking **Add New Row**, fill the new row, and end by clicking **Update**.
   f) Click **OK**.
      APM saves the configuration. The screen closes.

The SAML IdP connectors that you selected are bound the SAML SP service.

Exporting SAML SP metadata from APM

You need to convey the SP metadata from APM to the external SAML IdP that provides authentication service to this SP. Exporting the SAML SP metadata to a file provides you with the information that you need to do this.

1. On the Main tab, click **Access Policy** > **SAML** > **BIG-IP as IdP**.
   The BIG-IP as IdP screen opens and displays a list of local IdP services.
2. Select an SP service from the list and click **Export Metadata**.
   A popup window opens, displaying **No** on the **Sign Metadata** list.
3. For APM to sign the metadata, perform these steps:
   a) Select Yes from the **Sign Metadata** list.
   b) Select a key from the **Signing Key** list.
      APM uses the key to sign the metadata.
   c) Select a certificate from the **Signature Verification Certificate** list.
      APM exports the certificate to the metadata file. The system on which you import the metadata file can use the certificate to verify the metadata signature.

4. Select **OK**.
   APM downloads an XML file.

You must either import the XML file on the IdP system or use the information in the XML file to configure SP metadata on the IdP system .

Configuring an access policy to authenticate with an external SAML IdP

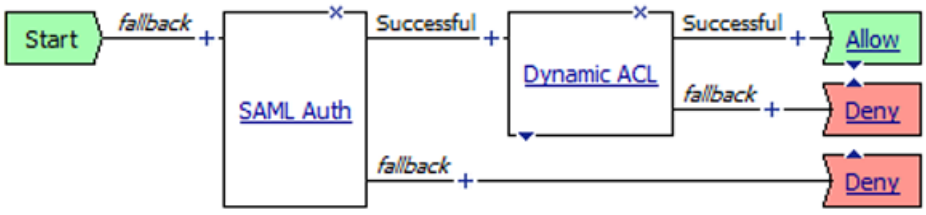
Before you start this task, configure an access profile.

When you use this BIG-IP® system as a SAML service provider (SP), configure an access policy to direct users to an external SAML Identity Provider (IdP) for authentication .

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. In the Access Policy column, click the **Edit** link for the access profile you want to configure.
   The visual policy editor opens the access policy in a separate screen.
3. Click the **(+)** icon anywhere in the access policy to add a new action item.
   A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
4. On the Authentication tab, select **SAML Auth** and click the **Add Item** button.
   The SAML Auth properties window opens.
5. In the SAML Authentication SP area from the **AAA Server** list, select a SAML SP service and click **Save**.
   The Access Policy window displays.
6. Add any additional actions that you require to complete the policy.
7. Change the Successful rule branch from **Deny** to **Allow** and click the **Save** button.
8. At the top of the window, click the **Apply Access Policy** link to apply and activate your changes to this access policy.
9. Click the **Close** button to close the visual policy editor.

You have an access policy that uses SAML authentication against an external SAML IdP and further qualifies the resources that a user can access.

Simple access policy to authenticate users against an external SAML IdP



To put the access policy into effect, you must attach it to a virtual server.

Adding the access profile to the virtual server

You associate the access profile with the virtual server so that Access Policy Manager® can apply the profile to incoming traffic.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the name of the virtual server you want to modify.
3. In the Access Policy area, from the **Access Profile** list, select the access profile.
4. Click **Update** to save the changes.

Your access policy is now associated with the virtual server.

Adding SAML SP metadata from APM to an external SAML IdP

To complete the agreement between APM® as the SAML service provider and a SAML IdP, you must configure service provider metadata at the IdP.

Note: *The method for configuring SAML service provider metadata at a SAML IdP will vary by vendor.*

Using the method that the vendor provides, either:

- Import the SAML SP metadata file that you exported from APM for a SAML SP service that is bound to the SAML IdP connector for this IdP.
- Or take information from the SAML SP metadata file that you exported from APM and add it using the vendor's interface. Pay particular attention to the values for entityID, AssertionConsumerService, and the certificate.

  Note: *Typically, the value of AssertionConsumerService is a URL that looks like this:* `https://bigip-sp-vs/saml/sp/profile/post/acs`.

Chapter

31

Using BIG-IP® IdP Automation

- *Overview: Automating SAML IdP connector creation*
- *Automating IdP connector creation for BIG-IP as SP*

Overview: Automating SAML IdP connector creation

When a BIG-IP® system is configured as a SAML service provider (SP), you can use SAML identity provider (IdP) automation to automatically create new SAML IdP connectors for SP services. Access Policy Manager® (APM®) polls a file or files that you supply; the files must contain cumulative IdP metadata. After polling, APM creates IdP connectors for any new IdPs and associates them with a specified SP service. APM uses matching criteria that you supply to send the user to the correct IdP.

When would I use SAML IdP automation?

Here is an example in which SAML Identity Provider (IdP) automation is especially useful. A large service provider (SP) supports a number of SAML identity providers. The service provider defines a SAML SP service on Access Policy Manager® (APM®) for access to that service. As IdPs come online, the service provider collects metadata from them and aggregates the IdP metadata into a file.

Note: *The process for collecting and aggregating IdP metadata into a file is up to the service provider.*

APM polls the metadata file, creates IdP connectors, associates new connectors to the specified SAML SP service, and ensures that clients performing SP-initiated access are sent to the correct IdP.

Automating IdP connector creation for BIG-IP as SP

To create a BIG-IP® Identity Provider (IdP) automation configuration, you need a BIG-IP® system that is configured to function as a SAML service provider (SP) and you need to have SAML SP services defined.

You create an IdP automation configuration to automatically create SAML IdP connectors and bind them to an SP service based on cumulative IdP metadata you maintain in a file or files. You specify matching criteria in the IdP automation for APM® to use, in order to send a user to the correct IdP.

1. On the Main tab, click **Access Policy** > **SAML** > **BIG-IP IdP Automation**.
   The BIG-IP IdP Automation screen opens and displays a table. Each row includes a configuration name, the URLs where IdP metadata files are stored for a particular SP service, and the name of the SP service to which automation applies.
2. Click **Create**.
   The Create New SAML IdP Automation popup screen opens.
3. In the **Name** field, type a name for the IdP automation configuration.
4. For the **SP Service** setting, select a service from the list.

   If the SP service you want has not already been defined, click **Create** to configure it and add it to the list.

   APM periodically creates SAML IdP connectors and binds them to the SP service you specify here.
5. From the **IdP Matching Source** list, select or type the name of a session variable.

   At the time of SP-initiated SAML single sign-on, APM (as a SAML SP) matches the value of this session variable to the value in the tag that you specify in the **Metadata Tag Match Value** field.
6. In the **Metadata Tag Match Value** field, type the name of a metadata tag.

   APM extracts the value in this tag from the IdP metadata and matches it with the value of the session variable specified in the **IdP Matching Source** field.

Note: *Do not include any wildcard in the value.*

7. In the **Metadata Tag For IdP Connector Name** field, type the name of a tag that is included in the IdP metadata.

   APM uses the value in the tag to name the IdP connector that it creates.

8. In the **Frequency** field, type a number of minutes.

   This specifies how often APM polls IdP metadata files.

9. Select **Metadata URLs** from the left pane.

   You specify URLs for one or more cumulative metadata files located on remote systems.

   A URL table displays in the right pane.

10. Specify a URL for each SAML IdP metadata file to be read. To add each URL, follow these steps:
    a) Click **Add**.
       A new field opens in the URL table.
    b) Type a URL.

       Begin the URL with `http` or `https`.

       For example, type `https://mywebsite.com/metdata/idp/idp_metadata.xml`.
    c) Click **Update**.
       The new URL displays in the top row of the table.

11. Click **OK**.

    The Create SAML IdP Automation screen closes. The new automation displays in the list.

For IdP automation to work, you must provide the metadata files as specified in the metadata URLs.

Chapter 32

BIG-IP System Federation for SP-Initiated Connections

- *Overview: Federating BIG-IP systems for SAML SSO (without an SSO portal)*
- *Task summary*

Overview: Federating BIG-IP systems for SAML SSO (without an SSO portal)

In a federation of BIG-IP® systems, one BIG-IP system acts as a SAML Identity Provider (IdP) and other BIG-IP systems act as SAML service providers (SPs).

This configuration supports:

- Only those connections that initiate at a service provider.
- Only service providers that accept assertions with similar subject type, attributes, and security settings.

About SAML IdP discovery

On a BIG-IP® system that you use as a SAML service provider (SP), you can bind an SP service to one or more SAML Identity Provider (IdP) connectors (each of which specifies an external IdP). When you bind an SP service to multiple IdP connectors, Access Policy Manager® chooses the correct IdP connector at run time through a filtering and matching process called IdP discovery.

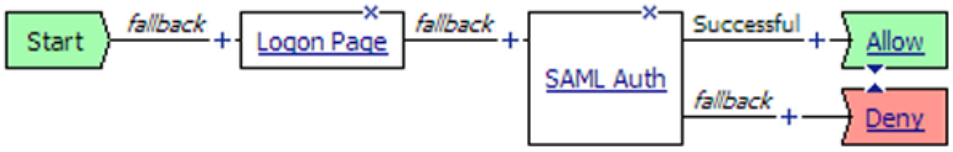
Scenario

You might bind multiple IdP connectors to an SP service on the BIG-IP system when you must provide services to different businesses and universities, each of which specifies an IdP to identify their users. When the user's information arrives at the SP service on the BIG-IP system, the SP service identifies the correct IdP and redirects the user to authenticate against that IdP before the SP service provides access to the service.

Note: *The SP service performs IdP discovery for a user only when the user initiates connection from an SP.*

Session variables and the typical access policy for BIG-IP system as SP

On a BIG-IP system configured as an SP, the typical access policy presents a logon page to the user. The Logon Page action populates session variables. You can customize the Logon Page action and affect session variable values. A SAML Auth action follows the logon page.



A SAML Auth action specifies an SP service. An SP service is an AAA service that requests authentication from an external IdP (specified in an IdP connector).

Session variables and SAML IdP discovery

Among multiple IdP connectors, the BIG-IP system must discover the correct external IdP with which to authenticate a user. For IdP discovery to work, you must specify matching criteria, a session variable name and value, for each IdP connector.

For example, users of a service might go to a particular landing page. When you bind the IdP connector, for the external IdP that serves those users, to the SP service, select the `%{session.server.landinguri}` session variable and supply a landing path value, such as, `/south*`. For users going to URLs such as

`https://sp-service/southwest` and `https://sp-service/southeast`, the SP service selects the same IdP to authenticate them.

Logon Page action customization

These are some common customization examples for the Logon Page action.

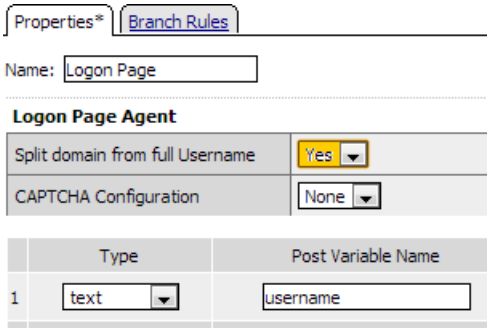


Figure 29: Setting the value of session.logon.last.domain variable to the domain name only

Select **Yes** for **Split domain from full Username**. The Logon Page agent takes the user name, such as joe@office.com, that was entered and creates the following session variables with these values.

Session Variable	Value
%{session.logon.last.username}	joe
%{session.logon.last.domain}	office.com
%{session.logon.last.logonname}	joe@office.com

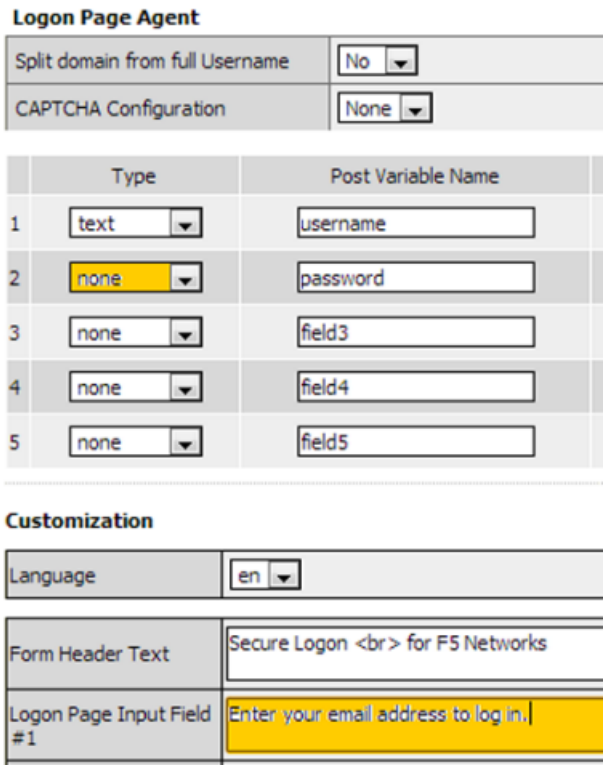


Figure 30: Obtaining and email address as the username

Change the prompt for the first field (username field). To omit the password, select Type none.

About local IdP service

A *SAML IdP service* is a type of single sign-on (SSO) authentication service in Access Policy Manager® (APM®). When you use a BIG-IP® system as a SAML identity provider (IdP), a SAML IdP service provides SSO authentication for external SAML service providers (SPs). You must bind a SAML IdP service to SAML SP connectors, each of which specifies an external SP. APM responds to authentication requests from the service providers and produces assertions for them.

About SP connectors

A SAML service provider connector (an SP connector) specifies how a BIG-IP® system, configured as a SAML Identity Provider (IdP), connects with an external service provider.

What are the available ways I can configure a SAML SP connector?

You can use one or more of these methods to configure SAML service provider (SP) connectors in Access Policy Manager®.

- From metadata - Obtain a metadata file from the vendor and import it into Access Policy Manager. The advantage to this method is that the vendor provides the majority of all required data, including certificates. You can complete the configuration by simply typing a unique name for the SP connector, a very few additional required fields, and browsing to and importing the file. Access Policy Manager then configures the SP connector.
- From template - Use templates that Access Policy Manager provides for some vendors; for example, Google. The advantages to this method are that:
  - Most required data is included in the template
  - Additional required data is minimal. You can obtain it and certificates from the vendor

  After you select a template and type data into a few fields, Access Policy Manager configures the SP connector.
- Custom - Obtain information from the vendor and type the settings into the Configuration utility. To use this method, you must also obtain certificates from the vendor and import them into the BIG-IP® system. Use this method when a metadata file or a template for an SP connector is not available.

About local SP service

A *SAML SP service* is a type of AAA service in Access Policy Manager® (APM®). It requests authentication from an external SAML Identity Provider (IdP) that is specified on APM in a SAML IdP connector. (You bind a SAML service provider (SP) service to one or more SAML IdP connectors.) APM requests authentication from an IdP and consumes assertions from it to allow access to resources behind APM.

About IdP connectors

An IdP connector specifies how a BIG-IP® system, configured as a SAML service provider (SP), connects with an external SAML identity provider (IdP).

About methods for configuring SAML IdP connectors in APM

You can use one or more of these methods to configure SAML identity provider (IdP) connectors in Access Policy Manager® (APM®).

- From metadata - Obtain a metadata file from the vendor and import it into APM. The advantage to this method is that the vendor provides all required data, including the certificate. You can complete the configuration by simply typing a unique name for the identity provider, and browsing to and importing the file. APM imports the certificate to the BIG-IP® system and configures the SAML IdP connector.
- From template - Use templates that APM provides for some vendors. The advantages to this method are that:
  - Most required data is included in the template. (Note that the certificate is not included.)
  - Additional required data is minimal and is available from the vendor.

  APM configures the SAML IdP connector. You must obtain a certificate from the vendor and import it into the BIG-IP system.
- Custom - Research the identity provider requirements and type all settings into the Configuration utility. Use this method when a metadata file or a template for an identity provider is not available. APM configures the SAML IdP connector. You must obtain a certificate from the vendor and import it into the BIG-IP system.
- IdP Automation - Provide files with cumulative IdP metadata on remote systems, then configure BIG-IP IdP automation to poll the files periodically and create IdP connectors and bind them to a specific service provider (SP) service.

Task summary

Setting up SAML federation for BIG-IP® systems involves three major activities:

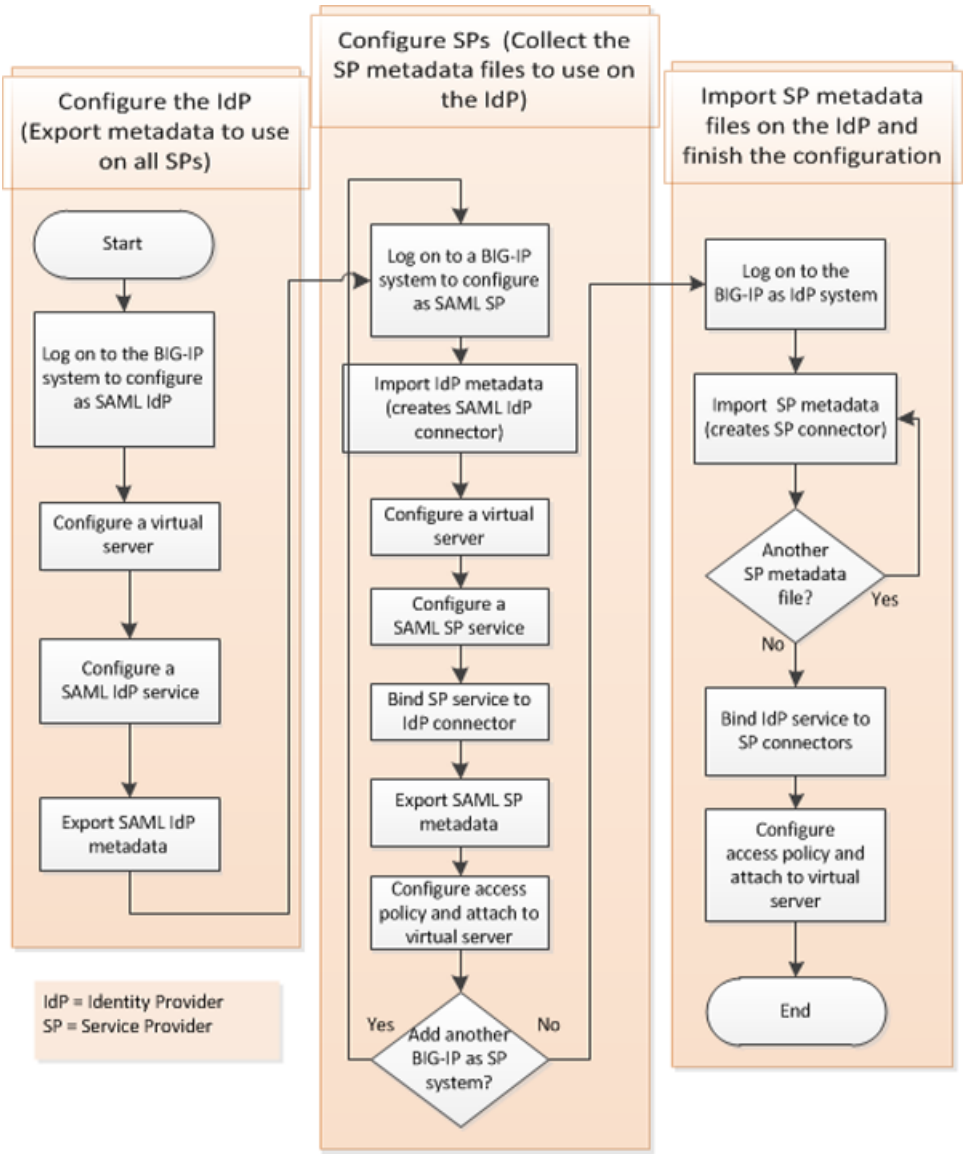
- First, you set up one BIG-IP system as a SAML identity provider (IdP) system
- Next, you set up one or more BIG-IP systems as a SAML service provider (SP)
- Last, you go back to the IdP system and set up connectivity to the SP systems

Task list

Setting up a BIG-IP system as a SAML IdP
Setting up a BIG-IP system as a SAML service provider system
Setting up connectivity from the IdP system to the SP systems

Flowchart: BIG-IP system federation configuration

This flowchart illustrates the process for configuring BIG-IP® systems in federation without providing an SSO portal.



Setting up a BIG-IP system as a SAML IdP

You log in to the BIG-IP® system that you have selected to act as the SAML Identity Provider (IdP) so that you can configure elements that are required for SAML federation.

Log on to the BIG-IP system that you have selected to act as the SAML IdP in a SAML federation of BIG-IP systems.

Creating a virtual server for a BIG-IP (as SAML IdP) system

Before you start this task, configure a client SSL profile and a server SSL profile if you are going to create an SSL virtual server.

Note: *Access Policy Manager® supports using a non-SSL virtual server for the BIG-IP® system configured as a SAML Identity Provider (IdP). However, we recommend using an SSL virtual server for security*

reasons. The following procedures include steps that are required for configuring an SSL virtual server, such as selecting client and server SSL profiles, and setting the service port to HTTPS.

Specify a host virtual server to use as the SAML IdP.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, select **Host** and in the **Address** field, type the IP address for the virtual server.
5. In the **Service Port** field, type `443` or select **HTTPS** from the list.
6. For the **HTTP Profile** setting, verify that the default HTTP profile, **http**, is selected.
7. For the **SSL Profile (Client)** setting, from the **Available** list, select the name of the Client SSL profile you previously created, and using the Move button, move the name to the **Selected** list.
8. For the **SSL Profile (Server)** setting, from the **Available** list, select the name of the Server SSL profile you previously created, and using the Move button, move the name to the **Selected** list.
9. Click **Finished**.

The virtual server for the BIG-IP system configured as an IdP now appears on the Virtual Server List. The virtual server destination is available for use in one or more SAML IdP service configurations.

Configuring a SAML IdP service

Configure a SAML Identity Provider (IdP) service for the BIG-IP® system, configured as a SAML IdP, to provide authentication service for SAML service providers (SPs).

Note: *Configure this IdP service to meet the requirements of all SAML service providers that you bind with it.*

1. On the Main tab, click **Access Policy** > **SAML** > **BIG-IP as IdP**.
   The BIG-IP as IdP screen displays a list of SAML IdP services.
2. Click **Create**.
   The Create New IdP Service popup screen displays.
3. In the **IdP Service Name** field, type a unique name for the SAML IdP service.
4. In the **IdP Entity ID** field, type a unique identifier for the IdP (this BIG-IP® system). Include the URI that points to the virtual server with the BIG-IP system and a unique path.
   For example, if you type `https://bigip-idp/idp`, `https://bigip-idp` should point to the virtual server you use for the BIG-IP system as a SAML IdP and `/idp` is a string that distinguishes one IdP from another when this BIG-IP system supports multiple SAML IdP services.

   The path portion on the IdP Entity ID is not a physical location on the BIG-IP system.
5. Click **Assertion Settings** from the left pane.

   The applicable settings display.
   a) From the **Assertion Subject Type** list, select the type of subject for the IdP to authenticate.
   b) From the **Assertion Subject Value** list, select the name of a session variable.

      This variable, `%{session.logon.last.username}`, is generally applicable. Some session variables are applicable depending on the type of authentication that you use for your site.
   c) Select the **Enable encryption of Subject** check box to encrypt the subject.

The **Encryption Strength** list becomes available.

d) From the **Encryption Strength** list, select a value.

Supported values are AES128, AES192, and AES256.

6. Click **SAML Attributes** from the left pane.

The SAML Attributes list displays. For each attribute that you want to include in the attribute statement, repeat these substeps.

a) Click **Add**.

b) Type a name and a value in the new row.

Usually, the name is a fixed string; it can be a session variable. You can use a session variable for the value. This example shows using a fixed string for the name and a session variable for the value. Name: `user_telephonenumber` and value: `%{session.ad.last.attr.telephoneNumber}`.

c) Select the **Encrypt** check box and select a value from the **Type** list.

Select the check box to encrypt the attribute. Supported values for type are AES128, AES192, and AES256.

d) Click **Update**.

7. Click **Security Settings** from the left pane.

a) From the **This device's Assertion Signing Key** list, select the key from the BIG-IP system store.

None is selected by default.

b) From the **This device's Public Certificate** list, select the certificate from the BIG-IP system store.

When selected, the IdP (the BIG-IP system) publishes this certificate to the service provider so that the service provider can verify the assertion. **None** is selected by default.

8. Click **OK**.
The popup screen closes. The new IdP service appears on the list.

APM creates a SAML IdP service. It is available to bind to SAML SP connectors. This service works with external service providers that share the same requirements for assertion settings and SAML attribute settings.

Exporting SAML IdP metadata from APM

You need to convey the SAML Identity Provider (IdP) metadata from APM to the external service providers that use the SAML IdP service. Exporting the IdP metadata for a SAML IdP service to a file provides you with the information that you need to do this.

1. On the Main tab, click **Access Policy** > **SAML** > **BIG-IP as IdP**.
The BIG-IP as IdP screen displays a list of SAML IdP services.
2. Select a SAML IdP service from the table and click **Export Metadata**.
A popup screen opens, with **No** selected on the **Sign Metadata** list.
3. For APM to sign the metadata, perform these steps:

a) Select Yes from the **Sign Metadata** list.

b) Select a key from the **Signing Key** list.

APM uses the key to sign the metadata.

c) Select a certificate from the **Signature Verification Certificate** list.

APM exports the certificate to the metadata file. The system on which you import the metadata file can use the certificate to verify the metadata signature.

4. Select **OK**.
   APM downloads an XML file.

An XML file that contains IdP metadata is available.

Setting up a BIG-IP system as a SAML service provider system

You log in once to each BIG-IP® system that you have selected to act as a SAML service provider so that you can configure the elements on it that are required for federation with other BIG-IP systems, one of which functions as an SAML IdP.

Log on to a BIG-IP system that you have selected to act as a SAML SP in a federation of BIG-IP systems.

Configuring an IdP connector from IdP metadata

Locate the SAML IdP metadata file that you exported from the BIG-IP® system (as IdP). If the metadata file is signed, obtain the certificate also; import it into the BIG-IP system store on this device.

Import IdP metadata to create a SAML IdP connector on this BIG-IP system. The SAML IdP connector enables this BIG-IP system to connect and exchange information with the external BIG-IP system that acts as the IdP in the SAML federation.

1. On the Main tab, click **Access Policy** > **SAML** > **BIG-IP as IdP**.
   The BIG-IP as IdP screen opens and displays a list of local IdP services.
2. On the menu bar, click **External IdP Connectors**.
   A list of SAML IdP connectors displays.
3. Select **Create** > **From Metadata**.
   The Create New SAML IdP Connector screen opens.
4. In the **Select File** field, browse to and select the metadata file for the IdP.
5. In the **Identity Provider Name** field, type a unique name for the IdP.
6. If the metadata is signed, select a certificate from the **Select Signing Certificate** list.
7. Click **OK**.
   The file is uploaded, the SAML IdP connector is created, and the screen closes.

The SAML IdP connector is displayed on the SAML IdP Connectors list.

Creating a virtual server for a BIG-IP (as SAML SP) system

Before you start this task, configure a client SSL profile and a server SSL profile.

Note: *Access Policy Manager® supports using a non-SSL virtual server for the BIG-IP system (as SP). However, we highly recommend using an SSL virtual server for security reasons. The following procedure includes steps that are required for configuring an SSL virtual server. These are: selecting client and server SSL profiles and setting the service port to HTTPS.*

Specify a host virtual server to use as the SAML SP.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.

3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, select **Host** and in the **Address** field, type the IP address for the virtual server.
5. In the **Service Port** field, type `443` or select **HTTPS** from the list.
6. For the **HTTP Profile** setting, verify that the default HTTP profile, **http**, is selected.
7. For the **SSL Profile (Client)** setting, from the **Available** list, select the name of the Client SSL profile you previously created, and using the Move button, move the name to the **Selected** list.
8. For the **SSL Profile (Server)** setting, from the **Available** list, select the name of the Server SSL profile you previously created, and using the Move button, move the name to the **Selected** list.
9. Click **Finished**.

The virtual server for the BIG-IP system configured as an SP now appears on the Virtual Server List. The virtual server destination is available for use in a SAML SP service configuration.

Configuring a SAML SP service for federation

Configure a SAML SP service for Access Policy Manager® to provide AAA authentication, requesting authentication and receiving assertions from a SAML IdP.

1. On the Main tab, click **Access Policy** > **SAML** > **BIG-IP as IdP**.
   The BIG-IP as IdP screen opens and displays a list of local IdP services.
2. Click **Create**.
   The Create New SAML SP Service screen opens.
3. In the **Name** field, type a unique name for the SAML SP service.
4. In the **Entity ID** field, type a unique identifier for the service provider that includes the URI that points to the virtual server you created for this BIG-IP system and a unique path.

   For example, if you type `https://bigip-sp/sp`, then https:/bigip-sp points to the virtual server on this BIG-IP system and "/sp" is a unique string.

   Note: *The path is not a physical path on the BIG-IP system, but a string that distinguishes one SAML SP service from another when multiple SAML SP services are configured on this BIG-IP system.*

   ---

5. In the **Relay State** field, type a scheme, host, and path.

   This is a path is where this BIG-IP system redirects users after they are authenticated.

6. From the left pane, select **Security Settings**.

   The screen changes to display the applicable settings.

   a) Select **Signed Authentication Request** if you want this BIG-IP system to send signed authentication requests to the SAML IdP.
   b) Select **Want Encrypted Assertion** if this BIG-IP system requires encrypted assertions from the SAML IdP.
   c) Select **Want Signed Assertion** if the BIG-IP service provider system requires signed assertions from the SAML IdP.

      This is selected by default. It is recommended that it be selected.

   d) From **SP's Authentication Signing/Assertion Decryption Private Key**, select a key from the BIG-IP system store on this device.

      You can select a private key only when you select at least one of these check boxes: **Signed Authentication Request** and **Want Encrypted Assertion**. APM uses this private key to sign the authentication request to the IdP and to decrypt an encrypted assertion from the IdP.

   e) From **SP Certificate**, select a certificate.

APM includes this certificate in the SAML SP metadata that you export. After the SAML SP metadata is imported on the IdP, the IdP can use this certificate to verify a signed authentication request and to encrypt an assertion.

7. Click **OK**.
   The screen closes.

APM creates the SAML SP service. It is available to bind to SAML IdP connectors and to export to a metadata file.

Binding the BIG-IP system (as IdP) with the SP service on this device

Bind the SAML SP service for this device (BIG-IP system) to the SAML IdP connector for the external BIG-IP system that acts as the IdP, so that this device requests authentication service from the IdP.

1. On the Main tab, click **Access Policy** > **SAML** > **BIG-IP as IdP**.
   The BIG-IP as IdP screen opens and displays a list of local IdP services.
2. Select a SAML SP service from the list.
3. Click **Bind/Unbind IdP Connectors**.
   A pop-up screen displays a list of any IdP connectors that are associated with this SP service.
4. Click **Add New Row**.
5. Select the SAML IdP connector for the BIG-IP system that acts as the IdP in the federation.
   Because you are binding only one IdP connector to the SP service, you do not need to fill in the **Matching Source** and **Matching Value** fields.
6. Click **Update**.
   The configuration is not saved until you click **OK**.
7. Click **OK**.
   APM saves the configuration. The screen closes.

The SAML IdP connector that you selected is bound to the SAML SP service.

Exporting SAML SP metadata from APM

You need to convey the SP metadata from APM to the external SAML IdP that provides authentication service to this SP. Exporting the SAML SP metadata to a file provides you with the information that you need to do this.

1. On the Main tab, click **Access Policy** > **SAML** > **BIG-IP as IdP**.
   The BIG-IP as IdP screen opens and displays a list of local IdP services.
2. Select an SP service from the list and click **Export Metadata**.
   A popup window opens, displaying **No** on the **Sign Metadata** list.
3. For APM to sign the metadata, perform these steps:
   a) Select Yes from the **Sign Metadata** list.
   b) Select a key from the **Signing Key** list.
      APM uses the key to sign the metadata.
   c) Select a certificate from the **Signature Verification Certificate** list.
      APM exports the certificate to the metadata file. The system on which you import the metadata file can use the certificate to verify the metadata signature.

4. Select **OK**.
   APM downloads an XML file.

You must either import the XML file on the IdP system or use the information in the XML file to configure SP metadata on the IdP system .

Configuring an access policy to authenticate with an external SAML IdP

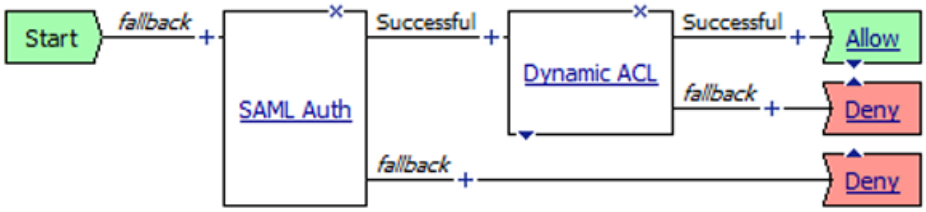
Before you start this task, configure an access profile.

When you use this BIG-IP® system as a SAML service provider (SP), configure an access policy to direct users to an external SAML Identity Provider (IdP) for authentication .

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. In the Access Policy column, click the **Edit** link for the access profile you want to configure.
   The visual policy editor opens the access policy in a separate screen.
3. Click the **(+)** icon anywhere in the access policy to add a new action item.
   A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
4. On the Authentication tab, select **SAML Auth** and click the **Add Item** button.
   The SAML Auth properties window opens.
5. In the SAML Authentication SP area from the **AAA Server** list, select a SAML SP service and click **Save**.
   The Access Policy window displays.
6. Add any additional actions that you require to complete the policy.
7. Change the Successful rule branch from **Deny** to **Allow** and click the **Save** button.
8. At the top of the window, click the **Apply Access Policy** link to apply and activate your changes to this access policy.
9. Click the **Close** button to close the visual policy editor.

You have an access policy that uses SAML authentication against an external SAML IdP and further qualifies the resources that a user can access.

Simple access policy to authenticate users against an external SAML IdP



To put the access policy into effect, you must attach it to a virtual server.

Adding the access profile to the virtual server

You associate the access profile with the virtual server so that Access Policy Manager® can apply the profile to incoming traffic.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the name of the virtual server you want to modify.
3. In the Access Policy area, from the **Access Profile** list, select the access profile.
4. Click **Update** to save the changes.

Your access policy is now associated with the virtual server.

Setting up connectivity from the IdP system to the SP systems

You log in to the BIG-IP® system that you configured as the SAML Identity Provider (IdP) so that you can set up connectivity to the BIG-IP systems you configured as SAML service providers (SPs).

Log on to the BIG-IP system that you have selected to act as the SAML IdP in a SAML federation of BIG-IP systems.

Configuring SAML SP connectors from SAML SP metadata files

Import SP metadata into this BIG-IP system from each BIG-IP system that is configured as an SP to create SP connectors in this system that you can use to create a federation of BIG-IP systems.

1. On the Main tab, click **Access Policy** > **SAML** > **BIG-IP as IdP**.
   The BIG-IP as IdP screen displays a list of SAML IdP services.
2. On the menu bar, click **External SP Connectors**.
   A list of SAML SP connectors displays.
3. Select **Create** > **From Metadata**
   The Create New SAML Service Provider window opens.
4. In the **Select File** field, browse to and select the metadata file for the service provider.
5. In the **Service Provider Name** field, type a unique name for the service provider.
6. If the metadata is signed, select the certificate from the **Select Signing Certificate** list.
7. Click **OK**.
   The file is uploaded, the SAML SP connector is created, and the window closes.

The SAML SP connector is displayed on the External SP Connectors list.

Binding IdP service and SP connectors for federation

Select a SAML Identity Provider (IdP) service and the SAML service provider (SP) connectors that use the service so that this BIG-IP system can provide authentication (SAML IdP service) to external SAML service providers.

1. On the Main tab, click **Access Policy** > **SAML** > **BIG-IP as IdP**.
   The BIG-IP as IdP screen displays a list of SAML IdP services.
2. Select a SAML IdP service from the list.

   A SAML IdP service provides authentication service.
3. Click **Bind/Unbind SP Connectors**.
   The screen displays a list of available SAML SP connectors.

4. Select the SAML SP connectors for the external BIG-IP systems that are configured as SPs and that you want to use this service.
5. Click **OK**.
   The screen closes.

The SAML IdP service is bound to the external SAML service providers specified in the SAML SP connectors.

Creating an access profile associated with the SAML IdP service

Use this procedure when this BIG-IP system, as a SAML Identity Provider (IdP), supports service provider-initiated connections only.

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. Click **Create**.
   The New Profile screen opens.
3. Type a name for the access profile.
4. In the SSO Across Authentication Domains (Single Domain mode) area, from the **SSO Configuration** list, select the name of the local SAML IdP service.
5. In the Language Settings area, add and remove accepted languages, and set the default language.

   A browser uses the highest priority accepted language. If no browser language matches the accepted languages list, the browser uses the default language.
6. Click **Finished**.

The access profile now shows up in the Access Profiles List.

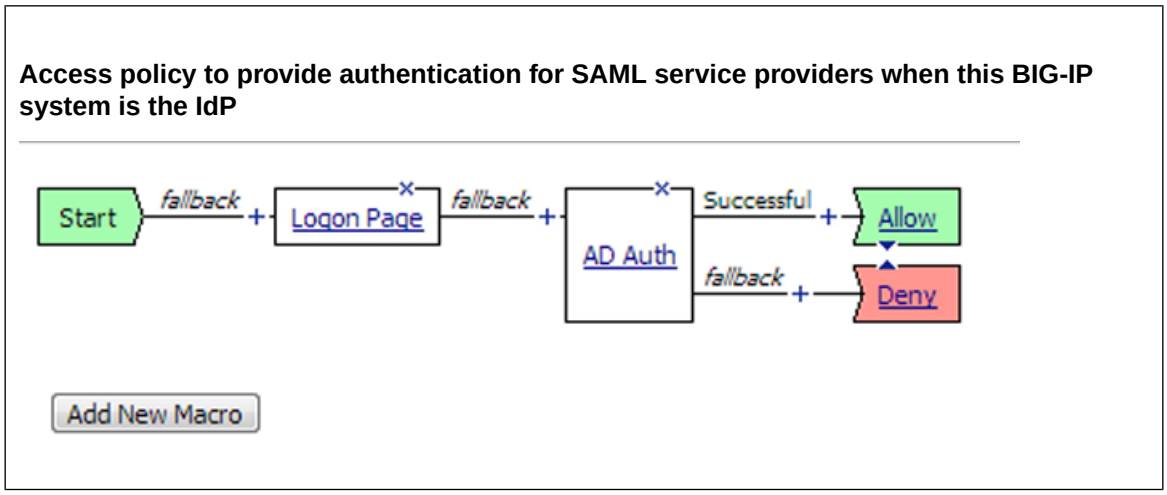
Configuring an access policy to provide authentication from the local IdP

Configure an access policy so that this BIG-IP® system, as a SAML Identity Provider (IdP) can provide authentication for SAML service providers.

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. In the Access Policy column, click the **Edit** link for the access profile you want to configure.
   The visual policy editor opens the access policy in a separate screen.
3. Click the **(+)** icon anywhere in the access policy to add a new action item.
   A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
4. On the Logon tab, select **Logon Page** and click the **Add Item** button.
   The Logon Page Agent properties screen opens.
5. Make any changes that you require to the logon page properties and click **Save**.
   The properties screen closes and the visual policy editor displays.
6. Add one or more authentication checks on the fallback branch after the **Logon Page** action.

   Select the authentication checks that are appropriate for application access at your site.
7. (Optional) Add any other branches and actions that you need to complete the access policy.
8. Change the Successful rule branch from **Deny** to **Allow** and click the **Save** button.
9. Click the **Apply Access Policy** link to apply and activate the changes to the access policy.
10. Click the **Close** button to close the visual policy editor.

You have an access policy that presents a logon page and authenticates the user..



To put the access policy into effect, you must attach it to a virtual server.

Adding the access profile to the virtual server

You associate the access profile with the virtual server so that Access Policy Manager® can apply the profile to incoming traffic.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the name of the virtual server you want to modify.
3. In the Access Policy area, from the **Access Profile** list, select the access profile.
4. Click **Update** to save the changes.

Your access policy is now associated with the virtual server.

Chapter

33

BIG-IP System Federation for SP- and IdP-Initiated Connections

- *Overview: Federating BIG-IP systems for SAML SSO (with an SSO portal)*
- *Task summary*

Overview: Federating BIG-IP systems for SAML SSO (with an SSO portal)

In a federation of BIG-IP® systems, one BIG-IP system acts as a SAML Identity Provider (IdP) and other BIG-IP systems act as SAML service providers.

This configuration supports:

- Connections that initiate at the IdP or at SAML service providers.
- Service providers that require different types of subject, attributes, and security settings for assertions.

About local IdP service

A *SAML IdP service* is a type of single sign-on (SSO) authentication service in Access Policy Manager® (APM®). When you use a BIG-IP® system as a SAML identity provider (IdP), a SAML IdP service provides SSO authentication for external SAML service providers (SPs). You must bind a SAML IdP service to SAML SP connectors, each of which specifies an external SP. APM responds to authentication requests from the service providers and produces assertions for them.

About SP connectors

A SAML service provider connector (an SP connector) specifies how a BIG-IP® system, configured as a SAML Identity Provider (IdP), connects with an external service provider.

What are the available ways I can configure a SAML SP connector?

You can use one or more of these methods to configure SAML service provider (SP) connectors in Access Policy Manager®.

- From metadata - Obtain a metadata file from the vendor and import it into Access Policy Manager. The advantage to this method is that the vendor provides the majority of all required data, including certificates. You can complete the configuration by simply typing a unique name for the SP connector, a very few additional required fields, and browsing to and importing the file. Access Policy Manager then configures the SP connector.
- From template - Use templates that Access Policy Manager provides for some vendors; for example, Google. The advantages to this method are that:
  - Most required data is included in the template
  - Additional required data is minimal. You can obtain it and certificates from the vendor

  After you select a template and type data into a few fields, Access Policy Manager configures the SP connector.
- Custom - Obtain information from the vendor and type the settings into the Configuration utility. To use this method, you must also obtain certificates from the vendor and import them into the BIG-IP® system. Use this method when a metadata file or a template for an SP connector is not available.

About local SP service

A *SAML SP service* is a type of AAA service in Access Policy Manager® (APM®). It requests authentication from an external SAML Identity Provider (IdP) that is specified on APM in a SAML IdP connector. (You

bind a SAML service provider (SP) service to one or more SAML IdP connectors.) APM requests authentication from an IdP and consumes assertions from it to allow access to resources behind APM.

About IdP connectors

An IdP connector specifies how a BIG-IP® system, configured as a SAML service provider (SP), connects with an external SAML identity provider (IdP).

About methods for configuring SAML IdP connectors in APM

You can use one or more of these methods to configure SAML identity provider (IdP) connectors in Access Policy Manager® (APM®).

- From metadata - Obtain a metadata file from the vendor and import it into APM. The advantage to this method is that the vendor provides all required data, including the certificate. You can complete the configuration by simply typing a unique name for the identity provider, and browsing to and importing the file. APM imports the certificate to the BIG-IP® system and configures the SAML IdP connector.
- From template - Use templates that APM provides for some vendors. The advantages to this method are that:
  - Most required data is included in the template. (Note that the certificate is not included.)
  - Additional required data is minimal and is available from the vendor.

  APM configures the SAML IdP connector. You must obtain a certificate from the vendor and import it into the BIG-IP system.
- Custom - Research the identity provider requirements and type all settings into the Configuration utility. Use this method when a metadata file or a template for an identity provider is not available. APM configures the SAML IdP connector. You must obtain a certificate from the vendor and import it into the BIG-IP system.
- IdP Automation - Provide files with cumulative IdP metadata on remote systems, then configure BIG-IP IdP automation to poll the files periodically and create IdP connectors and bind them to a specific service provider (SP) service.

Task summary

Setting up SAML federation for BIG-IP® systems involves three major activities:

- First, you set up one BIG-IP system as a SAML identity provider (IdP) system
- Next, you set up one or more BIG-IP systems as a SAML service provider (SP)
- Last, you go back to the IdP system and set up connectivity to the SP systems

Task list

Setting up a BIG-IP system as a SAML IdP
Setting up a BIG-IP system as a SAML service provider system
Setting up connectivity from the IdP system to the SP systems

Flowchart: BIG-IP system federation configuration with SSO portal

This flowchart illustrates the process for configuring BIG-IP® systems in federation and providing an SSO portal.

Setting up a BIG-IP system as a SAML IdP

You log in to the BIG-IP® system that you have selected to act as the SAML Identity Provider (IdP) so that you can configure elements that are required for SAML federation.

Log on to the BIG-IP system that you have selected to act as the SAML IdP in a SAML federation of BIG-IP systems.

Creating a virtual server for a BIG-IP (as SAML IdP) system

Before you start this task, configure a client SSL profile and a server SSL profile if you are going to create an SSL virtual server.

Note: *Access Policy Manager® supports using a non-SSL virtual server for the BIG-IP® system configured as a SAML Identity Provider (IdP). However, we recommend using an SSL virtual server for security*

reasons. The following procedures include steps that are required for configuring an SSL virtual server, such as selecting client and server SSL profiles, and setting the service port to HTTPS.

Specify a host virtual server to use as the SAML IdP.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, select **Host** and in the **Address** field, type the IP address for the virtual server.
5. In the **Service Port** field, type `443` or select **HTTPS** from the list.
6. For the **HTTP Profile** setting, verify that the default HTTP profile, **http**, is selected.
7. For the **SSL Profile (Client)** setting, from the **Available** list, select the name of the Client SSL profile you previously created, and using the Move button, move the name to the **Selected** list.
8. For the **SSL Profile (Server)** setting, from the **Available** list, select the name of the Server SSL profile you previously created, and using the Move button, move the name to the **Selected** list.
9. Click **Finished**.

The virtual server for the BIG-IP system configured as an IdP now appears on the Virtual Server List. The virtual server destination is available for use in one or more SAML IdP service configurations.

Configuring a SAML IdP service for one SP connector

Configure a SAML Identity Provider (IdP) service for Access Policy Manager®, as a SAML IdP, to provide single sign-on authentication for one SAML service provider (SP).

Note: *Configure one IdP service for each SAML service provider.*

1. On the Main tab, click **Access Policy** > **SAML** > **BIG-IP as IdP**.
   The BIG-IP as IdP screen displays a list of SAML IdP services.
2. Click **Create**.
   The Create New IdP Service popup screen displays.
3. In the **IdP Service Name** field, type a unique name for the SAML IdP service.
4. In the **IdP Entity ID** field, type a unique identifier for the IdP (this BIG-IP® system). Include the URI that points to the virtual server with the BIG-IP system and a unique path.
   For example, if you type `https://bigip-idp/idp`, `https://bigip-idp` should point to the virtual server you use for the BIG-IP system as a SAML IdP and `/idp` is a string that distinguishes one IdP from another when this BIG-IP system supports multiple SAML IdP services.
   The path portion on the IdP Entity ID is not a physical location on the BIG-IP system.
5. Click **Assertion Settings** from the left pane.
   The applicable settings display.
   a) From the **Assertion Subject Type** list, select the type of subject for the IdP to authenticate.
   b) From the **Assertion Subject Value** list, select the name of a session variable.
      This variable, `%{session.logon.last.username}`, is generally applicable. Some session variables are applicable depending on the type of authentication that you use for your site.
   c) Select the **Enable encryption of Subject** check box to encrypt the subject.
      The **Encryption Strength** list becomes available.

d) From the **Encryption Strength** list, select a value.
Supported values are AES128, AES192, and AES256.

6. Click **SAML Attributes** from the left pane.
The SAML Attributes list displays. For each attribute that you want to include in the attribute statement, repeat these substeps.
   a) Click **Add**.
   b) Type a name and a value in the new row.
   Usually, the name is a fixed string; it can be a session variable. You can use a session variable for the value. This example shows using a fixed string for the name and a session variable for the value. Name: `user_telephonenumber` and value: `%{session.ad.last.attr.telephoneNumber}`.
   c) Select the **Encrypt** check box and select a value from the **Type** list.
   Select the check box to encrypt the attribute. Supported values for type are AES128, AES192, and AES256.
   d) Click **Update**.

7. Click **Security Settings** from the left pane.
   a) From the **This device's Assertion Signing Key** list, select the key from the BIG-IP system store.
   None is selected by default.
   b) From the **This device's Public Certificate** list, select the certificate from the BIG-IP system store.
   When selected, the IdP (the BIG-IP system) publishes this certificate to the service provider so that the service provider can verify the assertion. **None** is selected by default.

8. Click **OK**.
The popup screen closes. The new IdP service appears on the list.

APM creates a SAML IdP service. It is available to bind to an SP connector.

Exporting SAML IdP metadata from APM

You need to convey the SAML Identity Provider (IdP) metadata from APM to the external service providers that use the SAML IdP service. Exporting the IdP metadata for a SAML IdP service to a file provides you with the information that you need to do this.

1. On the Main tab, click **Access Policy** > **SAML** > **BIG-IP as IdP**.
The BIG-IP as IdP screen displays a list of SAML IdP services.
2. Select a SAML IdP service from the table and click **Export Metadata**.
A popup screen opens, with **No** selected on the **Sign Metadata** list.
3. For APM to sign the metadata, perform these steps:
   a) Select Yes from the **Sign Metadata** list.
   b) Select a key from the **Signing Key** list.
   APM uses the key to sign the metadata.
   c) Select a certificate from the **Signature Verification Certificate** list.
   APM exports the certificate to the metadata file. The system on which you import the metadata file can use the certificate to verify the metadata signature.

4. Select **OK**.
APM downloads an XML file.

An XML file that contains IdP metadata is available.

Configuring a SAML resource and attaching a SAML IdP service

Configure a SAML resource to provide access to services on a SAML service provider when using APM as a SAML IdP.

Note: *Configure one SAML resource for each SAML IdP service that you have configured.*

1. On the Main tab, click **Access Policy** > **SAML** > **SAML Resources**.
   The SAML Resource list screen opens.
2. Click the **Create** button.
   The SAML Resource New Resource window opens
3. In the **Name** field, type a unique name for the SAML resource.
4. Do not clear the **Publish on Webtop** check box unless when you want to remove this resource from the webtop.

   When **Publish on Webtop** is selected, the SAML resource is displayed on a webtop where a user can initiate connection to an SP by clicking the icon. If you want users to initiate connection to this resource from an external SAML service provider only and do not want to show this resource on a webtop, clear the check box.
5. In the Configuration area from the **SSO Configuration** list, select the SAML IdP service that is bound to the SAML SP connector with the resources you want.
6. In the **Customization Settings for English** area in the **Caption** field, type a caption for this SAML resource.
7. Click **Finished**.
   The SAML resource is created and associated with a SAML IdP service that is bound to one external service provider.

Setting up a BIG-IP system as a SAML service provider system

You log in once to each BIG-IP® system that you have selected to act as a SAML service provider so that you can configure the elements on it that are required for federation with other BIG-IP systems, one of which functions as an SAML IdP.

Log on to a BIG-IP system that you have selected to act as a SAML SP in a federation of BIG-IP systems.

Configuring an IdP connector from IdP metadata

Locate the SAML IdP metadata file that you exported from the BIG-IP® system (as IdP). If the metadata file is signed, obtain the certificate also; import it into the BIG-IP system store on this device.

Import IdP metadata to create a SAML IdP connector on this BIG-IP system. The SAML IdP connector enables this BIG-IP system to connect and exchange information with the external BIG-IP system that acts as the IdP in the SAML federation.

1. On the Main tab, click **Access Policy** > **SAML** > **BIG-IP as IdP**.
   The BIG-IP as IdP screen opens and displays a list of local IdP services.
2. On the menu bar, click **External IdP Connectors**.
   A list of SAML IdP connectors displays.
3. Select **Create** > **From Metadata**.
   The Create New SAML IdP Connector screen opens.
4. In the **Select File** field, browse to and select the metadata file for the IdP.

5. In the **Identity Provider Name** field, type a unique name for the IdP.
6. If the metadata is signed, select a certificate from the **Select Signing Certificate** list.
7. Click **OK**.
   The file is uploaded, the SAML IdP connector is created, and the screen closes.

The SAML IdP connector is displayed on the SAML IdP Connectors list.

Creating a virtual server for a BIG-IP (as SAML SP) system

Before you start this task, configure a client SSL profile and a server SSL profile.

Note: *Access Policy Manager® supports using a non-SSL virtual server for the BIG-IP system (as SP). However, we highly recommend using an SSL virtual server for security reasons. The following procedure includes steps that are required for configuring an SSL virtual server. These are: selecting client and server SSL profiles and setting the service port to HTTPS.*

Specify a host virtual server to use as the SAML SP.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, select **Host** and in the **Address** field, type the IP address for the virtual server.
5. In the **Service Port** field, type `443` or select **HTTPS** from the list.
6. For the **HTTP Profile** setting, verify that the default HTTP profile, **http**, is selected.
7. For the **SSL Profile (Client)** setting, from the **Available** list, select the name of the Client SSL profile you previously created, and using the Move button, move the name to the **Selected** list.
8. For the **SSL Profile (Server)** setting, from the **Available** list, select the name of the Server SSL profile you previously created, and using the Move button, move the name to the **Selected** list.
9. Click **Finished**.

The virtual server for the BIG-IP system configured as an SP now appears on the Virtual Server List. The virtual server destination is available for use in a SAML SP service configuration.

Configuring a SAML SP service for federation

Configure a SAML SP service for Access Policy Manager® to provide AAA authentication, requesting authentication and receiving assertions from a SAML IdP.

1. On the Main tab, click **Access Policy** > **SAML** > **BIG-IP as IdP**.
   The BIG-IP as IdP screen opens and displays a list of local IdP services.
2. Click **Create**.
   The Create New SAML SP Service screen opens.
3. In the **Name** field, type a unique name for the SAML SP service.
4. In the **Entity ID** field, type a unique identifier for the service provider that includes the URI that points to the virtual server you created for this BIG-IP system and a unique path.

   For example, if you type `https://bigip-sp/sp`, then https:/bigip-sp points to the virtual server on this BIG-IP system and "/sp" is a unique string.

Note: *The path is not a physical path on the BIG-IP system, but a string that distinguishes one SAML SP service from another when multiple SAML SP services are configured on this BIG-IP system.*

5. In the **Relay State** field, type a scheme, host, and path.

   This is a path is where this BIG-IP system redirects users after they are authenticated.

6. From the left pane, select **Security Settings**.

   The screen changes to display the applicable settings.

   a) Select **Signed Authentication Request** if you want this BIG-IP system to send signed authentication requests to the SAML IdP.

   b) Select **Want Encrypted Assertion** if this BIG-IP system requires encrypted assertions from the SAML IdP.

   c) Select **Want Signed Assertion** if the BIG-IP service provider system requires signed assertions from the SAML IdP.

      This is selected by default. It is recommended that it be selected.

   d) From **SP's Authentication Signing/Assertion Decryption Private Key**, select a key from the BIG-IP system store on this device.

      You can select a private key only when you select at least one of these check boxes: **Signed Authentication Request** and **Want Encrypted Assertion**. APM uses this private key to sign the authentication request to the IdP and to decrypt an encrypted assertion from the IdP.

   e) From **SP Certificate**, select a certificate.

      APM includes this certificate in the SAML SP metadata that you export. After the SAML SP metadata is imported on the IdP, the IdP can use this certificate to verify a signed authentication request and to encrypt an assertion.

7. Click **OK**.
   The screen closes.

APM creates the SAML SP service. It is available to bind to SAML IdP connectors and to export to a metadata file.

Binding the BIG-IP system (as IdP) with the SP service on this device

Bind the SAML SP service for this device (BIG-IP system) to the SAML IdP connector for the external BIG-IP system that acts as the IdP, so that this device requests authentication service from the IdP.

1. On the Main tab, click **Access Policy** > **SAML** > **BIG-IP as IdP**.
   The BIG-IP as IdP screen opens and displays a list of local IdP services.
2. Select a SAML SP service from the list.
3. Click **Bind/Unbind IdP Connectors**.
   A pop-up screen displays a list of any IdP connectors that are associated with this SP service.
4. Click **Add New Row**.
5. Select the SAML IdP connector for the BIG-IP system that acts as the IdP in the federation.

   Because you are binding only one IdP connector to the SP service, you do not need to fill in the **Matching Source** and **Matching Value** fields.

6. Click **Update**.

   The configuration is not saved until you click **OK**.

7. Click **OK**.

   APM saves the configuration. The screen closes.

The SAML IdP connector that you selected is bound to the SAML SP service.

Exporting SAML SP metadata from APM

You need to convey the SP metadata from APM to the external SAML IdP that provides authentication service to this SP. Exporting the SAML SP metadata to a file provides you with the information that you need to do this.

1. On the Main tab, click **Access Policy** > **SAML** > **BIG-IP as IdP**.
   The BIG-IP as IdP screen opens and displays a list of local IdP services.
2. Select an SP service from the list and click **Export Metadata**.
   A popup window opens, displaying **No** on the **Sign Metadata** list.
3. For APM to sign the metadata, perform these steps:
   a) Select Yes from the **Sign Metadata** list.
   b) Select a key from the **Signing Key** list.
      APM uses the key to sign the metadata.
   c) Select a certificate from the **Signature Verification Certificate** list.
      APM exports the certificate to the metadata file. The system on which you import the metadata file can use the certificate to verify the metadata signature.
4. Select **OK**.
   APM downloads an XML file.

You must either import the XML file on the IdP system or use the information in the XML file to configure SP metadata on the IdP system .

Configuring an access policy to authenticate with an external SAML IdP

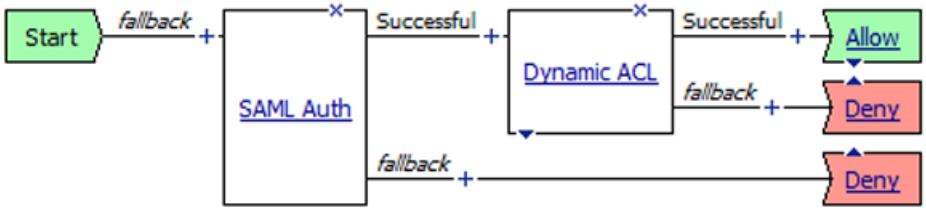
Before you start this task, configure an access profile.

When you use this BIG-IP® system as a SAML service provider (SP), configure an access policy to direct users to an external SAML Identity Provider (IdP) for authentication .

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. In the Access Policy column, click the **Edit** link for the access profile you want to configure.
   The visual policy editor opens the access policy in a separate screen.
3. Click the **(+)** icon anywhere in the access policy to add a new action item.
   A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
4. On the Authentication tab, select **SAML Auth** and click the **Add Item** button.
   The SAML Auth properties window opens.
5. In the SAML Authentication SP area from the **AAA Server** list, select a SAML SP service and click **Save**.
   The Access Policy window displays.
6. Add any additional actions that you require to complete the policy.
7. Change the Successful rule branch from **Deny** to **Allow** and click the **Save** button.
8. At the top of the window, click the **Apply Access Policy** link to apply and activate your changes to this access policy.
9. Click the **Close** button to close the visual policy editor.

You have an access policy that uses SAML authentication against an external SAML IdP and further qualifies the resources that a user can access.

Simple access policy to authenticate users against an external SAML IdP



To put the access policy into effect, you must attach it to a virtual server.

Adding the access profile to the virtual server

You associate the access profile with the virtual server so that Access Policy Manager® can apply the profile to incoming traffic.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the name of the virtual server you want to modify.
3. In the Access Policy area, from the **Access Profile** list, select the access profile.
4. Click **Update** to save the changes.

Your access policy is now associated with the virtual server.

Setting up connectivity from the IdP system to the SP systems

You log in to the BIG-IP® system that you configured as the SAML Identity Provider (IdP) so that you can set up connectivity to the BIG-IP systems you configured as SAML service providers (SPs).

Log on to the BIG-IP system that you have selected to act as the SAML IdP in a SAML federation of BIG-IP systems.

Configuring SAML SP connectors from SAML SP metadata files

Import SP metadata into this BIG-IP system from each BIG-IP system that is configured as an SP to create SP connectors in this system that you can use to create a federation of BIG-IP systems.

1. On the Main tab, click **Access Policy** > **SAML** > **BIG-IP as IdP**.
   The BIG-IP as IdP screen displays a list of SAML IdP services.
2. On the menu bar, click **External SP Connectors**.
   A list of SAML SP connectors displays.
3. Select **Create** > **From Metadata**
   The Create New SAML Service Provider window opens.
4. In the **Select File** field, browse to and select the metadata file for the service provider.

5. In the **Service Provider Name** field, type a unique name for the service provider.
6. If the metadata is signed, select the certificate from the **Select Signing Certificate** list.
7. Click **OK**.
   The file is uploaded, the SAML SP connector is created, and the window closes.

The SAML SP connector is displayed on the External SP Connectors list.

Binding a SAML IdP service to one SP connector

Bind a SAML Identity Provider (IdP) service and a SAML service provider (SP) connector so that the BIG-IP® system can provide authentication (SAML IdP service) to the external SAML service provider.

1. On the Main tab, click **Access Policy** > **SAML** > **BIG-IP as IdP**.
   The BIG-IP as IdP screen displays a list of SAML IdP services.
2. Select a SAML IdP service from the list.
   Select an IdP service that you configured for use with one particular SP connector only.
3. Click **Bind/Unbind SP Connectors**.
   The screen displays a list of available SAML SP connectors.
4. Select the one SAML SP connector that you want to pair with this IdP service.
5. Select **OK**.
   The screen closes.

The SAML SP connector that you selected is bound to the SAML IdP service.

Configuring a full webtop

A full webtop allows your users to connect and disconnect from a network access connection, portal access resources, SAML resources, app tunnels, remote desktops, and administrator-defined links.

1. On the Main tab, click **Access Policy** > **Webtops**.
2. Click **Create** to create a new webtop.
3. Type a name for the webtop you are creating.
4. From the **Type** list, select **Full**.
5. Click **Finished**.

The webtop is now configured, and appears in the list. You can edit the webtop further, or assign it to an access policy.

To use this webtop, it must be assigned to an access policy with an advanced resource assign action or with a webtop and links assign action. All resources assigned to the full webtop are displayed on the full webtop.

Configuring an access policy for a SAML SSO portal

Before you configure this access policy, configure an access profile without selecting an SSO configuration for it.

Configure an access policy so that the BIG-IP® system, as a SAML Identity Provider (IdP) can authenticate users using any non-SAML authentication type, and assign SAML resources and a webtop to the session.

Note: *This access policy supports users that initiate a connection at a SAML service provider or at the SAML IdP.*

---

1. On the Main tab, click **Access Policy** > **Access Profiles**.

The Access Profiles List screen opens.

2. In the Access Policy column, click the **Edit** link for the access profile you want to configure.
   The visual policy editor opens the access policy in a separate screen.
3. Click the **(+)** icon anywhere in the access policy to add a new action item.
   A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
4. On the Logon tab, select **Logon Page** and click the **Add Item** button.
   The Logon Page Agent properties screen opens.
5. Make any changes that you require to the logon page properties and click **Save**.
   The properties screen closes and the visual policy editor displays.
6. Add one or more authentication checks on the fallback branch after the **Logon Page** action.

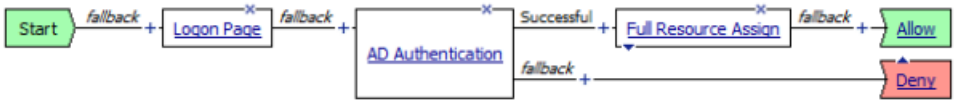
   Select the authentication checks that are appropriate for application access at your site.
7. On a successful branch after an authentication check, assign SAML resources and a full webtop to the session.
   a) Click plus **[+]** on a successful branch after an authentication check.
      The Add Item window opens.
   b) On the Assignment tab, select the **Advanced Resource Assign** agent, and click **Add Item**.
      The Resource Assignment window opens.
   c) Click **Add new entry**.
      An **Empty** entry displays.
   d) Click the **Add/Delete** link below the entry.
      The screen changes to display resources on multiple tabs.
   e) Select the SAML tab, then from it select the SAML resources that represent the service providers that authorized users can access.
   f) Click **Update**.
      The window changes to display the Properties screen, where the selected SAML resources are displayed.
   g) Click the **Add/Delete** link below the entry.
      The screen changes to display resources on multiple tabs.
   h) Select the Webtop tab, then select a full webtop on which to present the selected resources.

      You must assign a full webtop to the session even if you have configured all SAML resources to not publish on a webtop.
   i) Click **Update**.
      The window changes to display the Properties screen. The selected webtop and SAML resources are displayed.
   j) Click **Save**.
      The Properties window closes and the Access Policy window is displayed.

   You have configured a webtop to display resources that are available from service providers and that an authorized user can access.
8. (Optional) Add any other branches and actions that you need to complete the access policy.
9. Change the Successful rule branch from **Deny** to **Allow** and click the **Save** button.
10. Click the **Apply Access Policy** link to apply and activate the changes to the access policy.
11. Click the **Close** button to close the visual policy editor.

You have an access policy that presents a logon page, authenticates the user, and assigns SAML resources and a full webtop on which to present them to the user.

Simple access policy for access to services on SAML service providers



To put the access policy into effect, you must attach it to a virtual server.

Adding the access profile to the virtual server

You associate the access profile with the virtual server so that Access Policy Manager® can apply the profile to incoming traffic.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the name of the virtual server you want to modify.
3. In the Access Policy area, from the **Access Profile** list, select the access profile.
4. Click **Update** to save the changes.

Your access policy is now associated with the virtual server.

Index

A

AAA authentication
  local SP service *250*, *268*, *274*, *282*, *288*
AAA high availability
  about *20*
  configuring *20*
  testing for supported authentication servers *31*, *51*, *76*, *142*, *164*
AAA server
  configuring for Active Directory *28*
  RADIUS authentication *74*
  SecurID *63*
  server pool *74*
AAA servers
  about *20*
  and Active Directory *34*
  configuring *20*
access policy
  accounting using TACACS+ *163*
  authenticating with external IdP *236*, *247*, *258*, *276*, *278*, *290*, *292*
  authenticating with TACACS+ *163*
  demanding an SSL certificate *146*
  for Active Directory AAA server *29*, *39*
  for LDAPS *49*
  LDAP query *57*
  one-time password configuration *158*
  one-time -password configuration *154*
  supporting SAML SSO portal *236*, *292*
  using SAML Auth agent *258*, *276*, *290*
access profile
  adding to virtual server *102*, *106*, *117*, *134*, *142*, *147*, *237*, *247*, *259*, *276*, *279*, *291*, *294*
  and SAML IdP service *246*, *278*
  configuring for SSO *217*
  creating *29*, *48*, *64*, *74*, *91*, *100*, *105*, *112*, *116*, *133*, *139*
  managing for SSO *217*
Active Directory
  and authentication *28*
  configuring an AAA server *28*
  configuring a Trusted Domain *34*
  overview of trusted domains *34*
Active Directory AAA server
  access policy for *29*
Active Directory attributes
  binary *27*, *39*
Active Directory cross-domain rules *32*
Active Directory password management
  about *26*, *38*
  password management
    for Active Directory *26*, *38*
Active Directory query
  access policy for *39*
Active Directory session variables
  descriptions *42*
Active Directory troubleshooting tips
  list of *33*, *43*
Active Directory Trusted Domains *34*
AD auth and LocalDB lockout macro
  example *123*
  loop terminal, about *123*
AD query
  about *38*
  using in addition to authentication *38*
  using IPv6 *40*
attributes
  for RADIUS *77*
authentication
  adding to access policy *22*
  and logon item *21*
  configuring for external server *110*
  configuring for LDAP *47*, *63*
  configuring for Microsoft Exchange clients *22*
  multi-factor *21*
  two-factor *64*
  using a one-time password *22*
  using RSA SecurID *64*
authentication: configuring end-user login *90*
authentication and authorization
  configuring HTTP Basic/NTLM *110*
  configuring LDAPS *48*
authentication method
  configuring for HTTP form-based method *111*
authentication methods
  for Active Directory *26*
  for HTTP *110*
  for LDAP *46*
  for LDAPS *46*
  for OCSP *132*
  for RADIUS *72*
  for RADIUS accounting *82*
  for TACACS+ *162*
authentication servers
  testing support for AAA high availability *31*, *51*, *76*, *142*, *164*

B

basic authentication and Kerberos end-user logon
  about *88*
BIG-IP Access Policy ManagerOAM AAA serverSAML AAA server
  finding documentation for *22*
BIG-IP system and SAML
  benefits of *222*
BIG-IP system as IdP configuration
  configuration flowchart *230*, *241*
  single BIG-IP system configuration *223*
  SSO portal *230*
  supporting SP- and IdP-initiated connections *228*
  supporting SP-initiated connections only *240*
BIG-IP system as SP configuration
  and SAML IdP discovery *250*, *266*
  configuration flowchart *253*
  configuration requirements *250*

BIG-IP system as SP configuration *(continued)*
single BIG-IP system configuration *224*
virtual server, configuring *255*, *273*, *288*
BIG-IP system federation
binding local SP *275*, *289*
configuration flowchart: multiple IdP services *284*
configuration flowchart for one IdP service *269*
configuring an IdP *270*, *277*, *284*, *291*
configuring an SP *273*, *287*
importing SP metadata *277*, *291*
local IdP service, binding *277*
SP- and IdP-initiated connection support *282*
SP-initiated connection support only only *266*
BIG-IP systems
machine account, sharing *96*
Bugzilla
and SSO configuration example *186*

C

Ceridian
and SSO configuration example *187*
certificate
exporting SSL *225*
obtaining from external system *224*
obtaining through metadata *224*
providing for external system *224*
requirement *224*
certificate inspection
at session start *21*
On-Demand *21*
certificate revocation list
retrieving *138*
certificate revocation status
using client SSL profile *21*
using CRLDP *21*
using OCSP *21*
Citrix 4.5 and Citrix 5
and SSO configuration example *189*
client certificate inspection
task summary *150*
using client SSL profile *150*
verifying CRL *150*
clientssl profile
configuring for CRLDP *141*
client SSL profile
adding to virtual server *134*, *142*, *147*
client SSL profiles
creating *147*
Client SSL profiles
creating *146*
common session variables
descriptions *32*, *42*, *51*, *58*, *77*, *165*
common session variablesActive Directory session variables
descriptions *32*, *42*, *51*, *58*, *77*, *165*
common session variablesLDAP session variables
descriptions *32*, *42*, *51*, *58*, *77*, *165*
configuration flowchart
BIG-IP system as IdP *241*
BIG-IP system as IdP with SSO portal *230*
BIG-IP system as SP *253*
BIG-IP system federation *282*
configuration flowchart *(continued)*
BIG-IP system federation (multiple IdP services) *284*
BIG-IP system federation (one IdP service) *269*
supporting SP-initiated connection only *241*
configuring *244*, *271*
CRL distribution point
directory path (dirName) *138*
HTTP URL *138*
LDAP URI *138*
CRLDP
about *138*
configuring clientssl profile *141*
CRLDP access policy
example *135*, *143*
CRLDP Auth item
including in access policy *140*
CRLDP session variables
descriptions *143*
CRLDP troubleshooting tips
list of *144*
cross-domain rules for Active Directory *32*

D

delegation account
Kerberos SSO *96*
setting up *98*, *201*
documentation, finding *22*
domain join *97*
Domino Web Access configuration example *186*

E

emails
sending through SMTP server *154*
end-user logon
about *88*
end-user logon access policy
example *92*
end-user logon support, configuring *91*
Exchange profile
access profile, property of *96*
access profile setting *100*, *105*
configuring *99*, *104*
Microsoft Exchange services, for *96*
external IdP connector
about *252*, *268*, *283*
creating from template *252*, *269*, *283*
custom configuration *252*, *269*, *283*
importing from file *252*, *269*, *283*
external SP connector
about *229*, *240*, *268*, *282*
and BIG-IP system federation *277*, *291*
creating from template *229*, *241*, *268*, *282*
custom configuration *229*, *232*, *241*, *243*, *268*, *282*
importing from metadata *229*, *241*, *268*, *282*
importing metadata *277*, *291*

F

form-based client-initiated SSO
about *180*

form-based client-initiated SSO *(continued)*
and default behavior *180*
configuring basic *180*
form-based client-initiated SSO authentication
using *182*
form-based client-initiated SSO configuration
and minimum requirements *180*
examples *186*
for Bugzilla *186*
for Ceridian *187*
for Citrix 4.5 and Citrix 5 *189*
for DevCentral *189*
for Google *190*
for Oracle Application Server 10g *191*
for OWA 2003 *192*
for OWA 2010 and OWA 2007 *191*
for Perforce *192*
for Reviewboard *193*
for Salesforce *194*
for SAP *193*
for Sharepoint 2010 *195*
for Weblogin *196*
for Yahoo *196*
matching criteria *180*
form-based client-initiated SSO configuration example
for Domino Web Access *186*
form-based client-initiated SSO settings
described *183*
form request
changing default properties *181*
form submittal
changing default properties *181*
full webtop
configuring *236*, *292*

H

high availability
Active Directory AAA server *26*, *47*, *72*, *138*, *162*
CRLDP AAA server *26*, *47*, *72*, *138*, *162*
LDAP AAA server *26*, *47*, *72*, *138*, *162*
RADIUS AAA server *26*, *47*, *72*, *138*, *162*
RADIUS guidelines *72*
TACACS+ AAA server *26*, *47*, *72*, *138*, *162*
testing for supported authentication servers *31*, *51*, *76*, *142*, *164*
HTTP authentication
with HTTPS authentication server *115*
HTTP Basic/NTLM
configuring for authentication *110*
HTTP Basic SSO configuration settings *173*
HTTP form-based
configuring authentication *111*
HTTP form-based authentication
hidden form parameter example *157*
using to send a one-time password *156*
HTTP Forms SSO configuration settings *174*
HTTPS authentication
configuring *114*

I

Identity Provider, See IdP
IdP
about *222*
IdP connector
configuring *254*
creating from metadata *273*, *287*
creating IdP connectors automatically *262*
IdP connectors
creating automatically *262*
IdP metadata
exporting *235*
importing to APM *273*, *287*
IdP service
SSO authentication *229*, *240*, *268*, *282*
IPv6
workaround for AD Query *40*

J

JavaScript insertion *181*

K

Kerberos
Kerberos
access profile for *203*
virtual server for *203*
Kerberos authentication method
supporting *99*, *201*
Kerberos authentication requirements
about *89*
Kerberos authentication troubleshooting tips *93*
Kerberos configuration
domain, joining a *90*
user account, creating *90*
Kerberos SSO
about *200*
configuring *99*, *201*
editing an access policy *202*
overview *200*
setting up a delegation account *98*, *201*
supporting *99*, *200–201*
task summary *200*
Kerberos SSO configuration settings *203*
Kerberos SSO object *203*
Kerberos SSO session variables
descriptions *206*
Kerberos SSO troubleshooting tips
list of *206*

L

layered virtual servers
configuring for SSO *218*
LDAP attributes
handling binary values *46*, *56*
LDAP auth and query default rules
example *31*, *51*, *57*
LDAP query
memberOf attribute *56*

LDAP query *(continued)*
primary group attribute *56*
LDAP QueryAD Query
nested group *38*, *56*
LDAP Query session variable
in LDAP Query *58*
session variable *58*
LDAPS
configuring for authentication *48*
virtual server *50*
LDAP troubleshooting tips
list of *52*, *59*
local database users
dynamic *120*
exporting and importing from file *120*
local database write action
Common partition *120*, *124*
locked out status, updating *120*, *124*
login failures, updating *120*, *124*
local IdP metadata
configuring at service provider *238*, *248*
importing to service provider *238*, *248*
local IdP service
244, *271*
and SP connector *234*, *292*
and SP connectors *245*
binding *234*, *245*, *292*
configuring *233*, *285*
exporting *246*, *272*, *286*
in access profile *246*, *278*
XML file *246*, *272*, *286*
local SP metadata
configuring at external IdP *259*
importing into external IdP *259*
local SP service
about *250*, *268*, *282*
and IdP connectors *256*
binding *256*
configuring *255*, *274*, *288*
exporting *257*, *275*, *290*
XML file *257*, *275*, *290*
local user
password change, forcing *122*
local user database
123
authenticating users *125*
configuring *22*, *120*
creating *121*
in an access policy *123*
locking a user account *127*
lockout interval *121*
lockout threshold *121*
read action, example of *123*
unlocking user account *127*
user lockout *125*
local user database HA
ConfigSync, compared to *120*
error message reporting *120*
schedule, understanding *120*
sync status, viewing *120*
local users
adding *122*
Logon Page
customization *64*
Logon Page action
collecting two passwords *67*
customization example *67*

M

machine account
renewing password for *102*
replication *96*
machine trust account
configuring in Access Policy Manager *97*
macro
AD auth query OTP by email and resources *154*
AD query auth OTP by HTTP and resources *156*
memberOf
session variable *38*, *56*
metadata
about *222*
exporting local IdP service *246*, *272*, *286*
exporting local SP service *257*, *275*, *290*
external IdP connector, importing *252*, *269*, *283*
external SP connector, importing *229*, *241*, *268*, *282*
Microsoft Exchange applications
service settings for *104*
Microsoft Exchange client
configuring authentication for *22*
HTTP basic authentication, supporting *106*
Microsoft Exchange clients
and NTLM authentication *96*
providing full proxy for *96*, *104*
Microsoft Exchange client traffic
APM configuration for *99*, *104*
multi-domain support
overview *210*
multi-domain support for SSO
about *210*
attaching an access policy *213*
configuring *212*

N

NAS *74*
network access
configuring *216*
for SSO with web applications *216*
network access properties
configuring *217*
for SSO with web applications *217*
network access server *74*
NTLM authentication
about using *96*
accessing domain-joined Microsoft Exchange clients *97*
delegation account *96*
determining NTLM *100*
Kerberos SSO requirement *96*
machine account *96*
NTLMV1 SSO configuration settings *176*
NTLMV2 SSO configuration settings *177*

O

OCSP
  about *132*
  authenticating with *132*
  configuring clientssl profile *134*
OCSP session variables
  descriptions *135*
OCSP troubleshooting tips *136*
one-time password
  configuring in an access policy *22*

P

pool
  high availability *20*
  load-balancing *20*
portal access resource
  configuring *218*
  for SSO *218*
profile
  creating for client-side SSL *146–147*
profiles
  creating CRLDP *150*

R

RADIUS
  authentication *74*
  server pool *74*
RADIUS attributes
  descriptions *77*
  handling binary values *73*, *82*
  list of *77*
RADIUS high availability guidelines *72*
RADIUS session variables
  descriptions *77*
RADIUS troubleshooting tips *78*, *84*
routing domain *20*
RSA SecurID
  authentication agent *62*
  RADIUS client *62*
  token policy *62*
RSA SecurIDauthentication
  about *62*
  token *62*
  two-factor with SecurID *62*
RSA SecurID on Windows troubleshooting tips *68*
RSA SecurID session variables
  descriptions *68*

S

SAML
  about *222*
SAML and BIG-IP system
  benefits of *222*
SAML Auth agent *258*, *276*, *290*
SAML IdP automation
  for BIG-IP as SP *262*
SAML IdP discovery
  about *250*, *266*
SAML resource
  configuring *235*, *287*
  publishing on webtop *235*, *287*
SAML single logout *222*
service provider, See SP
service settings
  ActiveSync *104*
  Autodiscover *104*
  Exchange Web Service *104*
  NTLM authentication *104*
  Offline Address Book *104*
  Outlook Anywhere *104*
session variable
  session.user.otp.pwd *158*
session variables
  Active Directory *32*, *42*
  common *32*, *42*, *51*, *58*, *68*, *77*, *135*, *143*, *165*
  CRLDP *143*
  LDAP *51*, *58*
  OCSP *135*
  RADIUS *77*
  RSA SecurID *68*
  TACACS-plus *165*
single domain support for SSO
  configuring *212*
Single Sign-On
  about *170*
  with credential caching and proxying *170*
Single Sign-On configuration objects
  about *172*
SMS
  and HTTP authentication *156*
SMTP server
  configuring *154*
SP
  about *222*
SSL certificate
  importing *224*
  importing from external SP *232*, *243*
SSL certificate and key
  importing from external IdP *250*
SSL certificates
  defined *21*
SSO
  configuring *173–174*
  configuring form-based client-initiated authentication *182*
  configuring for NTLM v1 authentication method *176*
  configuring for NTLM v2 authentication method *177*
  configuring multi-domain support *212*
  configuring single domain support *212*
  configuring with Kerberos *99*, *201*
  supporting *99*, *201*
  using form-based authentication method *174*
  using HTTP basic authentication method *173*
SSO authentication
  local IdP service *229*, *240*, *268*, *282*
SSO common use cases
  deploying *216*
SSO domain support
  configuring *212*
  for SSO *212*
SSO methods supported *172*

SSO portal configuration *228*

T

TACACS+
accounting *163*
authentication *163*
task summary *27*, *73*, *162*
TACACS+ access policy
example *164*
TACACS+ troubleshooting tips *165*
TACACS-plus session variables
descriptions *165*
template
external IdP connector, configuring from *252*, *269*, *283*
external SP connector, configuring from *229*, *241*, *268*, *282*
token code
collecting on logon page *64*
troubleshooting tips
for Active Directory *33*, *43*
for CRLDP *144*
for Kerberos authentication *93*
for LDAP *52*, *59*
for OCSP *136*
for RADIUS *78*, *84*
for RSA SecurID on Windows *68*
for TACACS+ *165*
trusted CA certificates *21*
trusted domains
and Active Directory *34*

U

UserDN settings
for LDAP *52*
user group
fetching the memberOf attribute *38*
fetching the primary group *38*
user lockout
from a local database *123*
from an external system *123*

V

virtual server
associating with access profile *30*, *66*, *76*, *114*
BIG-IP system as IdP configuration *231*, *242*, *270*, *284*
BIG-IP system as SP configuration *255*, *273*, *288*
configuring for LDAPS *50*
creating *115*
defining for *30*, *66*, *76*, *114*
for HTTPS authentication *115*
for web applications *217*
over network access *217*
SSL configuration *231*, *242*, *255*, *270*, *273*, *284*, *288*

W

web application over network access
configuring *216*
for OAM *216*
webtops
configuring full *236*, *292*
write action, example of *123*