Ministerie van Defensie

NEXTLABS

Server User Attribute Provider

User Guide

October 18, 2018

V1.0

Version Control

Date	Authors	Version/Comments
10/18/2018	Kent Lee	V1.0 – First release

THE INFORMATION CONTAINED IN THIS DOCUMENT IS PROVIDED"AS IS" WITHOUT ANY EXPRESS REPRESENTATIONS OR WARRANTIES. IN ADDITION, NEXTLABS, INC. DISCLAIMS ALL IMPLIED REPRESENTATIONS AND WARRANTIES, INCLUDING ANY WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

Table of Contents

1. Overview

1.1 Server User Attribute Provider

Server User Attribute Provider is a Dynamic User Attribute Provider plugin designed for Policy Controller/Java Policy Controller to query and cache user attributes from Active Directory (AD) Files and User AOR information by calling SAP Function. The plugin can provide user attribute data on-demand for Policy Controller evaluation when the attribute is not available in the enforcer request, by hitting its own cache or querying AD directly or querying SAP server.

2. Functionalities

2.1 Query user and group attributes from AD

- A user or a group can be identified by more than one identifier (ID). These IDs depend on the ID that the PEPs use to send request.
- The plugin should return the same user or group regardless of which ID the PEP passes. This means two PEPs using different attributes in AD for ID can work with one PDP.
- **All IDs must be unique across all domains configured.**
- More than one user attributes or group attributes can be configured to be retrieved. Each attribute is either single-valued or multi-valued and such information should be specified in the plugin's properties file.
- A LDAP user search filter and LDAP user search base can be configured in the plugin's properties file to limit the user returned. Search base is not applicable for LDIF and only simple filters like (objectclass=user) is applicable.
- A LDAP group search filter and LDAP group search base can be configured in the plugin's properties file to limit the group returned. Search base is not applicable for LDIF and only simple filters like (objectclass=group) is applicable.
- (Only applicable for LDAP domain type) Customer can choose whether to retrieve disabled accounts or not.
- The plugin would return EvalValue.NULL if user or group is not found.
- The plugin would return EValValue.NULL or MultiValue.EMPTY if the attribute is not found, depending on the attribute's cardinality.
- The plugin connects to AD using connection pool. The connection pool settings are configurable.
- The plugin loads all user information from the LDIF file in to the cache and also reloads the cache based on the cache_refresh_period property value. If domain type is LDIF, make sure that the cache_refresh_period value is not set to 0.
- The plugin supports connecting to AD using SSL. In order to configure SSL, a trust store containing the AD certificate needs to be specified in the plugin's properties.
- The plugin works in 2 way SSL communication, user will need to create the keystore to store the private key and distribute the certificate to AD server manually for 2 way SSL communication.

2.2 Query user AOR information from SAP AOR Master

- A user AOR information can be query by connecting to SAP server by RFC connection
- SAP side will provide a function call which will be utilize by this feature to fetch the data.
- The RFC connection is maintain by SAPJavaSDK plugin in Policy Controller, this feature will only retrieve the connection and make use of it.
- After the AOR information retrieved, it will be store inside the AOR cache region in the plugin.

2.3 Maintain an in-memory cache of user attributes.

- On PDP start/restart, the plugin would contain no data about user.
- Cache entry has a configurable expiration (time_to_live), which can be from 0 to **INFINITE**. This expiration is calculated from the time the entry register itself into cache. Supported units are: SECS, MINS,HRS,DAYS
- When time_to_live expires, cache will be missed on the next request and the plugin will query the user from AD and put back the user or group to cache.
- On PDP shutdown, the cache will be destroyed.

2.4 Maintain an in-memory cache of AOR attributes.

- The cache can be configure to work on live mode or purge mode.
- On PDP start/restart, the plugin would contain no data about AOR.
- When in live mode, cache entry has a configurable expiration (time_to_live), which can be from 0 to **INFINITE**. This expiration is calculated from the time the entry register itself into cache. Supported units are: SECS, MINS,HRS,DAYS
- When time_to_live expires, cache will be missed on the next request and the plugin will query the user from AD and put back the user or group to cache.
- When in purge mode, the cache refresh process will be started at a configurable fixed timestamp and subsequent refreshes will happen at a fixed rate calculated from the start timestamp. All the data in cache will be purge.
- In refresh process, if the refresh mode is turn on, the plugin will pull all the AOR information and put into cache after purging.
- When encounter exceptions, the cache refresh process will attempt a configurable number of retries between a configurable period
- On PDP shutdown, the cache will be destroyed

3. Component

3.1 Package

- ServerUserAttributeProvider.jar
- ServerUserAttributeProvider.properties
- suap-truststore.jks
- SAPJavaSDK-UAP.txt

3.2 Deploy

To deploy the plugin in the Policy Controller

- Copy **ServerUserAttributeProvider.properties and suap-truststore.jks** to **[PolicyController]/jservice/config (dpc/jservice/config in Java Policy Controller)**
- Copy **ServerUserAttributeProvider.jar** to a convenient folder. It's recommended to copy the jar to **[PolicyController]/jservice/jar (dpc/jservice/jar in Java Policy Controller)**
- Open **SAPJavaSDK-UAP.txt** and copy the content inside and append to **[PolicyController]/jservice/config/ SAPJavaSDKService.properties. (dpc/jservice/config in Java Policy Controller).**Edit the value accordingly to match SAP Sever information.

3.3 Configure

Open **ServerUserAttributeProvider.properties** to configure the plugin

3.3.1 General information

Property	Meaning	Sample value
name	Plugin name. Must not be changed.	ServerUserAttributeProvider
jar-path	The path of the ServerUserAttributeProvider.jar.	[NextLabs]/Policy Controller/jservice/jar/ServerUserAttributeProvider.jar
description	Description of the plugin. Can leave as default.	User Attributes Plugin
friendly_name	Friendly name of the plugin. Can leave as default.	User Attributes Service

3.3.2 Cache information

Property	Meaning	Sample value
cache_heap_in_mb	Cache size used to store user information, adjust accordingly to user data size. Size is MB unit. If it is set to too low, cache missed will happened more frequent. Old	2048

	entry will be remove if cache is full based on first in first in first out.	
cache_max_object	Maximum objects in a cache element, if the maximum objects in element is more than setting, the old entry will be discarded based on first in first in first out.	50000
number_of_retries	Number of retries when an exception occurs during a refresh. Default value is 3	3
interval_between_retries	Time in second between retries when exception occurs. Default value is 30	30
user_time_to_live	The maximum period that user cache entry (a user) stays in the cache since last entry to the cache. The format of the property should be **<period>_<unit>.** Supported units are **SECS, MINS, HRS, DAYS**. When this period has passed for a user, the user will be removed from the cache. The plugin will query the AD for the user and put him back to the cache when the user is asked next time. Default value is 1_HRS	1_HRS
aor_expired_mode	Cache expired mode for AOR cache, live or purge	purge
aor_time_to_live	This setting take effect when aor_expired_mode is live. The maximum period that aor cache entry (a user) stays in the cache since last entry to the cache. The format of the property should be **<period>_<unit>.** Supported units are **SECS, MINS, HRS, DAYS**. When this period has passed for a user, the user will be removed from the cache. The plugin will query the AD for the user and put him back to the cache when the user is asked next time. Default value is 1_DAYS	1_DAYS
aor_purge_time	This setting take effect when aor_expired_mode is purge. The hour of the day, when purge triggered. When this triggered, the AOR cache will be flush. The format is hh:MM	01:00
aor_refresh	This setting take effect when aor_expired_mode is purge. Plugin will query all the AOR data from SAP and put it into the cache after the purge completed. The value can be true/false	true

aor_attributes_to_ pull	AOR attributes that the plugin needs to pull from SAP. This field must contain the exact attributes used in SAP. Attributes are separated by comma and prefixed by cardinality. Valid cardinality are "single:" and "multi:". The cardinality of the attributes should follow on the cardinality of such attributes in SAP.	multi:sloc,multi:whnum,multi:bizpn,multi:shippt,multi:fe
sap_server_prefix	The connection prefix which contain SAP connection information from SAPJavaSDKService.properties	SERV3_
sap_handler	SAP function name to retrieve AOR data	ZFM_NXL_GET_AOR_ATTRS

3.3.3 LDAP information

Property	Meaning	Sample value
pool_max_size	The maximum number of connections per connection identity that can be maintained concurrently in the pool. Default value is 20.	20
pool_pref_size	The preferred number of connections per connection identity that should be maintained concurrently in the pool. Default value is 10.	10
pool_init_size	The number of connections per connection identity to create when initially creating a connection for the identity. Default value is 1.	1
pool_time_out	The number of milliseconds that an idle connection may remain in the pool without being closed and removed from the pool. Default value is 30000.	360000
pool_debug	The level of debug output to produce. Valid values are "none", "fine" (trace connection creation and removal) and "all" (all debugging information).	false
key_store	The location of the keystore containing the private key of the plugin, which would be used for SSL connection. This is needed for 2 way SSL communication. The file will need to be created manually by user. The format should be in jks format. Optional for one way SSL communication	C:/Program Files/NextLabs/Policy Controller/jservice/config/suap-keystore.jks

key_store_pass	The password of the key store, in encrypted form. The password should be in encrypted format using NextLabs crypto tool which can be from in control center.	sa549f6ba05c840e5f43ef63e06a8ae1a
trust_store	The location of the trust store containing the certificate of the Active Directory, which would be used for SSL connection. The **suap-trustore.jks** in the package is an empty trust store that can be used.	C:/Program Files/NextLabs/Policy Controller/jservice/config/suap-truststore.jks
trust_store_pass	The password of the trust store, in encrypted form. The password for the suap-truststore.jks is **sa549f6ba05c840e5f43ef63e06a8ae1a**, which is the encrypted string for **123blue!**	
paging_size	The size of each batch of users returned in a LDAP search. By default LDAP restricts this number to be 1000 maximum. The number in this properties file cannot be larger than the setting in the AD. Default value is 1000.	1000
null_string	The string to return when the result is a NULL object. By commenting out this property, a NULL object will be returned	NO_DATA

Property	Meaning	Sample value
DOMAIN_1_host	(Only for LDAP) The host name or IP address of the Active Directory.	anvm105
DOMAIN_1_port	(Only for LDAP) The port of the Active Directory.	389 for non SSL, 636 for SSL
DOMAIN_1_ssl	(Only for LDAP) Boolean flag to indicate whether to connect to the AD using SSL channel or not. When set to true, the certificate of the AD must be imported into the trust store used by the plugin.	true
DOMAIN_1_authentication	(Only for LDAP) Authentication mode for the AD, it can be simple or none, for none the username and password will be optional.	simple
DOMAIN_1_username	(Only for LDAP) The account user name used to query users This account should have read permission on all user objects in the Active Directory.	uid=admin,ou=system

DOMAIN_1_password	(Only for LDAP) The password of the account in encrypted format.	s819587290b046f43e1a1910728a7b7a9
DOMAIN_1_get_disabled_accounts	(Only for LDAP) The Boolean flag to decide if the plugin should retrieve disabled accounts in AD or not.	false
DOMAIN_1_user_search_base	(Only for LDAP) The bases of the search for users in Active Directory. Multiple search bases can be specified, separated by semicolon.	o=mindef,c=nl
DOMAIN_1_user_filter	The search filter which would be applied to all user searches executed by the plugin.	(|(objectClass=user)(objectClass=person))
DOMAIN_1_user_key_attributes	The string listing the ID attribute(s) of users in AD. Multiple attributes can be listed, separated by comma, prefixed by case sensitivity. If the attribute is prefixed by 'cs:', the plugin will do a case-sensitive query for the IDs of that attribute. If the attribute is prefixed by 'ci:'. the plugin will do a case-insensitive query for the IDs of that attribute. Case-insensitive query is useful when the user ID is case-converted by the enforcer or the enforced application, such as Exchange Server, which makes the ID value passed by the enforcer into the Policy Controller different from the value provided by the AD.	ci:employeeNumber
DOMAIN_1_user_attributes_to_pull	The string listing the user attributes which need to be pulled by the plugin. Attributes are separated by comma and prefixed by cardinality. Valid cardinality are "single:" and "multi:". The cardinality of the attributes should follow on the cardinality of such attributes in AD.	multi:ammoArea,single:clearanceCode,single:sapAccessLevel

4. Troubleshooting

Open **logging.properties** in **[PolicyController]/config**

Find a line containing **com.bluejungle.level** and add **com.nextlabs.level = FINEST** in the bellow line (change the level to **FINEST** if the **com.nextlabs.level** is already present)

Restart the Policy Controller.

5. Appendix

5.1 Server User Attribute Provider, Enforcer and Enrollment

A User Attribute Provider will only be triggered for an attribute if that attribute value is not provided by the enforcer.

With enrollment, any condition in components involving enrolled attributes will be precompiled by the Control Center. As such, policy is matched based on the ID of the user, not by any condition on any attribute. Hence, the Server User Attribute Provider will only be triggered in the advanced condition block, or when the policy asks for an unenrolled attribute in components.

Without enrollment, the Server User Attribute Provider will always be triggered in both components and advanced condition block when the policy asks for an attribute that is not provided by the enforcer.

5.2 Key store

A key store of an application (A) is a container holding all private key of all applications/servers (A). From the private key in this store, a certificate can be extract and import to application (B) trust store so that application(B) can treated application(A) as trusted client.

The **suap-keystore.jks** will need to created manually by user to enable 2 way SSL communicaition. In order to create the key store, the **keytool** utility of Java or the application **KeyStore Explorer** (http://keystore-explorer.org/) can be used. It is not compulsory to use the suap-keystore.jks. It is needed when 2 way SSL communication is in place.

Comment out this portion if 2 way SSL is not in-use.

5.3 Trust store

A trust store of an application (A) is a container holding all certificates of all applications/servers (B) that this application (A) should trust. By trusting an application or a server (B), this application (A) can send requests to (B) using one-way SSL channel.

The **suap-truststore.jks** is an empty trust store packaged with the plugin. To use this trust store, all certificates used for SSL of the Active Directory needs to be imported to this trust store. In order to import a certificate to a trust store, the **keytool** utility of Java or the application **KeyStore Explorer** (http://keystore-explorer.org/) can be used. The password for the suap-truststore.jks is **123blue!**

It is not compulsory to use the suap-truststore.jks. An existing trust store with the necessary certificates can be used as well.

5.4 Password Encryption

To create an encrypted password, a tool located at **[PolicyServer]/tools/crypt/mkpassword.bat** can be used. A sample command is: **mkpassword.bat –password password** with the result of **scc1d86db33184b96278786ce58a47957**