Guacamole LDAP Authentication with Active Directory

Guacamole supports LDAP authentication via an extension available from the main project website. This extension allows users and connections to be stored directly within an LDAP directory including Active Directory

The LDAP authentication module will need an Active Directory implementation as storage for all authentication data, and the instructions here assume you already have AD installed and working.

The user name entered on the Guacamole authentication page will be their common name(cn) or display name (i.e. John Smith), not their login name (i.e. jsmith). The LDAP Authentication extension binds to LDAP using a users DN, which identifies the individual user via the cn attribute not the sAMAccountName attribute.

Disclaimer: The procedures contained in this document PERMANENTLY modify the Active Directory schema. Please be carefull.

Installing LDAP authentication

The LDAP authentication module is not included in the main Guacamole bundle nor is it enabled by default. You must use the download link provided in the downloads section of the main Guacamole site.

The downloaded .tar.gz file will contain several directories:

lib/
: Contains all .jar files required for the LDAP authentication module to work, including the module itself and the LDAP library driving it.

schema/
: Contains OpenLDAP specific files that are not used with Active Directory.

The contents of lib/ must be copied into the classpath of Guacamole, which is the directory specified by the lib-directory property in guacamole.properties. If this property is not specified, simply add it. On Linux servers, /var/lib/guacamole/classpath is a good choice, but it can be whatever you like.

After copying the files in place, check to make sure all files are present, and there are no conflicts in between multiple versions ofguacamole-auth-ldap. The contents should match at least the files shown here:

```
$ ls /var/lib/guacamole/classpath
```

```
guacamole-auth-ldap-0.8.0.jar  jldap-4.3.jar
$
```

Each of the .jar files above is either the LDAP authentication module itself guacamole-auth-ldap-0.8.0.jar) or a dependency. They must all be placed in Guacamole's lib-directory for the LDAP authentication to work.

Configuring Guacamole

Additional properties must be added to guacamole.properties for Guacamole to load the LDAP support and for the LDAP support to properly connect to your AD server:

```
# Auth provider class
auth-provider:
net.sourceforge.guacamole.net.auth.ldap.LDAPAuthenticationProvider

# LDAP properties
ldap-hostname:           ad1.contoso.com
ldap-port:               389
ldap-user-base-dn:       CN=Users,DC=contoso,DC=com
ldap-username-attribute: CN
ldap-config-base-dn:     CN=Users,DC=contoso,DC=com
```

The LDAP support depends on the following properties, as shown in the example above:

ldap-hostname
: The hostname of your AD domain controller.

ldap-port
: The port your LDAP server listens on.

ldap-user-base-dn
: The base of the DN (Distinguished Name) for all Guacamole users. This will be appended to the username when a user logs in. The example above assumes your users are located in the default Users container. All Guacamole users must be located in the same container/OU.

ldap-username-attribute
: The attribute which contains the username which is part of the DN. With active Directory this should be set to CN

ldap-config-base-dn
: The base of the DN for all Guacamole configurations. Each configuration is analogous to a connection. Within Guacamole's LDAP support, each configuration functions as a group, having user members. A user which is a member of a particular configuration group will have access to that configuration.
  This base DN will be used when querying all configurations accessible by a user once they have successfully logged in.

With the above properties properly set, Guacamole will connect to your AD server after you

restart Tomcat (or whatever servlet container you are using). You will still need to install the schema modifications to your AD server such that you can create new configurations and associated them with users.

Preparing Active Directory

This procedure permanently modifies the Active Directory schema. Please see this article for best practices when making schema changes.

Copy the guacSchema.ldif file from bellow to the AD Domain controller. Modify the ldif file replacing “DC=contoso,DC=com” with the correct distinguished name for your domain.

guacSchema.ldif

```
#Attribute definitions

dn: CN=guacConfigParameter,CN=Schema,CN=Configuration,DC=contoso,DC=com
changetype: ntdsschemaadd
objectClass: top
objectClass: attributeSchema
cn: guacConfigParameter
attributeID: 1.3.6.1.4.1.38971.1.1.2
attributeSyntax: 2.5.5.12
isSingleValued: FALSE
adminDisplayName: guacConfigParameter
adminDescription: guacConfigParameter
oMSyntax: 64
searchFlags: 1
lDAPDisplayName: guacConfigParameter
systemOnly: FALSE

dn:
changetype: modify
add: schemaUpdateNow
schemaUpdateNow: 1
-

dn: CN=guacConfigProtocol,CN=Schema,CN=Configuration,DC=contoso,DC=com
changetype: ntdsschemaadd
objectClass: top
objectClass: attributeSchema
cn: guacConfigProtocol
attributeID: 1.3.6.1.4.1.38971.1.1.1
```

```
attributeSyntax: 2.5.5.12
isSingleValued: FALSE
adminDisplayName: guacConfigProtocol
adminDescription: guacConfigProtocol
oMSyntax: 64
searchFlags: 1
lDAPDisplayName: guacConfigProtocol
systemOnly: FALSE

dn:
changetype: modify
add: schemaUpdateNow
schemaUpdateNow: 1
-

# Classes

dn: CN=guacConfigGroup,CN=Schema,CN=Configuration,DC=contoso,DC=com
changetype: ntdsschemaadd
objectClass: top
objectClass: classSchema
cn: guacConfigGroup
governsID: 1.3.6.1.4.1.38971.1.2.1
rDNAttID: cn
adminDisplayName: guacConfigGroup
adminDescription: guacConfigGroup
objectClassCategory: 1
lDAPDisplayName: guacConfigGroup
name: guacConfigGroup
systemOnly: FALSE
subClassOf: groupOfNames
mayContain: guacConfigParameter
mustContain: guacConfigProtocol


dn:
changetype: modify
add: schemaUpdateNow
schemaUpdateNow: 1
-
```

Open a command prompt and run this command to make the schema changes Replace DC=contoso,DC=com with the distinguished name for your domain. Also replace the <username>, <domain>, and <password> fields with the auth information for a user in the schema admins group, typically administrator.

```
ldifde -i -f <Path>\guacSchema.ldif -b <username> <domain> <password> -k -j .
-c "CN=Schema,CN=Configuration,DC=contoso,DC=com" #schemaNamingContext
```

Example:

```
C:\>ldifde -i -f guacSchema.ldif -b administrator contoso secret -k -j . -c
"CN=Schema,CN=Configuration, DC=contoso,DC=com" #schemaNamingContext
Connecting to "ad1.contoso.com"
Logging in as current user using SSPI
Importing directory from file "guacSchema.ldif"
Loading entries.......
6 entries modified successfully.
The command has completed successfully
C:\>
```

Adding Config Groups to Active Directory

Guacamole's LDAP support allows users and connections to be managed purely within Active Directory. This is accomplished with a minimum of changes to the standard LDAP schema - all Guacamole users are traditional AD users. The only new type of object required is a representation for Guacamole connections, guacConfigGroup, which was added to your server's schema during the install process above.

Users

All Guacamole users, as far as the LDAP support is concerned, are AD users with standard AD credentials. When a user signs in to Guacamole, their Common Name and password will be used to bind to the AD server. If this bind operation is successful, the available connections are queried from the directory and the user is allowed in.

Connections and parameters

Each connection is represented by an instance of the guacConfigGroup object class, which is simply an extended version of the standard LDAP groupOfNames which provides a protocol and set of parameters. Only members of the guacConfigGroup will have access to the corresponding connection.

The guacConfigGroup object class provides two new attributes in addition to those provided by groupOfNames:

guacConfigProtocol
: The protocol associated with the connection, such as "vnc" or "rdp". This attribute is required for every guacConfigGroup and can be given only once.

guacConfigParameter
: The name and value of a parameter for the specified protocol. This is given as name=value, where "name" is the name of the parameter as defined by the documentation for the protocol specified, and "value" is any allowed value for that parameter.
: This attribute can be given multiple times for the same connection.

For example, to create a new RDP connection which connects to host1 at port 3389, while granting access to John Smith, you could create an .ldif file like the following:

```
DN: CN=TestGuacGroup,CN=Users,DC=contoso,DC=com
changetype: add
CN: TestGuacGroup
objectClass: guacConfigGroup
guacConfigProtocol: rdp
guacConfigParameter: hostname=host1
guacConfigParameter: port=3389
member: CN=John Smith,CN=Users,DC=contoso,DC=com
```

The new connection can then be added to AD using the ldifde utility:

```
C:\>ldifde -i -f guacUser.ldif
Connecting to "AD1.contoso.com"
Logging in as current user using SSPI
Importing directory from file "guacUser.ldif"
Loading entries..
1 entry modified successfully.

The command has completed successfully
```

Note: Much of the content in this page is based off the [LDAP Authenticaiton] page included with the Guacamole documentation.