Configuring SPNEGO based SSO with Tomcat and Active Directory

Table of contents

1.Introduction....... 1
2.Compilation....... 1
3.Installation and Configuration....... 1
1.General....... 1
2.Writing Kerberos configuration....... 2
3.Configuring a JAAS Login Module for Kerberos Authentication....... 2
4.Configuring SpnegoRealm in Tomcat....... 2
5.Configuring SpnegoValve in Tomcat....... 3

1.Introduction

Tomcat provides a low level http request interception mode much more general than servlet filters : *valves*. Such objects allows one to work on a http request before it reaches the servlet container. To enable automatic authentication, jaaslounge provides you a Valve that realizes the same job than jCifs-ext Authentication Filter, but at a more general level. With this valve, you can use the declarative security, at the application deployment descriptor level.

However, simply using a valve do not allows you to check for roles. This is why we provide a custom realm, which works with Active Directory to obtain groups information associated with the user authenticated by the valve.

2.Compilation

The SPNEGO based Valve and Realm provided consists in several classes. To compile these classes you need the following libraries :

- jCifs-ext : version 0.9.4
- jCifs : version 1.1.11 , later versions must not be used as jCifs-ext does not work with.
- catalina.jar : found in directory “server/lib” of tomcat installation
- servlet-api.jar : found in directory “common/lib” of tomcat installation
- commons-logging-api.jar : found in directory “bin” of tomcat installation

3.Installation and Configuration

1. General

First of all, ensure that your active directory is properly configured. Steps required to configure Active Directory are detailed in the document “AD_Spnego_Configuration”.

Once you have generated a jar containing all the required classes (the package org.jaaslounge.sso.tomcat), copy it to the “server/lib” directory of your Tomcat installation. You then need to configure Tomcat.

2. Writing Kerberos configuration

To make use of Kerberos, you need to write an configuration file. This file specifies settings like the Kerberos Realm, address of the server... Here is a sample of such a file, considering that our realm is named MY.DOMAIN.COM, and that our Active Directory server is named adserver.my.domain.com :

```
[libdefaults]
default_realm = MY.DOMAIN.COM
default_tkt_enctypes = rc4-hmac des-cbc-md5 des-cbc-crc des3-cbc-sha1
default_tgs_enctypes = rc4-hmac des-cbc-md5 des-cbc-crc des3-cbc-sha1

[realms]
 MY.DOMAIN.COM = {
  kdc = adserver.my.domain.com:88
  admin_server = adserver.my.domain.com
  default_domain = MY.DOMAIN.COM
 }

[domain_realm]
 .my.domain.com = MY.DOMAIN.COM
 MY.DOMAIN.COM = MY.DOMAIN.COM
```

Either save this file as C:\winnt\krb5.ini (on windows), or add an option in CATALINA_OPTS environment variable to point to krb5 configuration file; add the following entry to CATALINA_OPTS : -Djava.security.krb5.conf=*path* , where "*path*" points to your Kerberos configuration file.

3. Configuring a JAAS Login Module for Kerberos Authentication

By default, jCifs-ext comes with 2 pre-configured aliases to sun's Kerberos Login Modules : jcifs.spnego.initiate and jcifs.spnego.accept. There is no need to change these settings except when you use a Kerberos Login module different from sun's one. You can then specify the path to the login configuration by modifying the CATALINA_OPTS environment variable; add the following entry : -Djava.security.auth.login.config=*path*, where "*path*" points to your login configuration file.

4. Configuring SpnegoRealm in Tomcat

Open the file *server.xml* located in the directory "*conf*" of tomcat installation. Under the section "*Engine*", add a "*Realm*" entry. Suppose that we work with an Active Directory server which dns is adserver.my.domain.com; and the active directory domain is MY.DOMAIN.COM. Then you will add the following lines in your configuration file :

```
      <Realm className="org.jaaslounge.sso.tomcat.spnego.SpnegoRealm"
            domainController="adserver.my.domain.com"
            servicePrincipalName="HTTP/webserver.my.domain.com"
            servicePassword="mypassword"
            loginModule="jcifs.spnego.accept"
            ldapSearchContext="DC=my,DC=domain,DC=com"
            contextFactory="com.sun.jndi.ldap.LdapCtxFactory"
            stripGroupNames="true"
       />
```

The available options for SpnegoRealm are :

- domainController : address of the domain controller.
- servicePrincipalName : principal name identifying the service to allow tomcat to connect to active directory using Kerberos.
- servicePassword : password to be used with the principal name to allow tomcat to connect to active directory using Kerberos.
- loginModule : alias identifying the login module to use to connect to active directory using Kerberos (usually : jcifs.spnego.accept).
- ldapSearchContext : base distinguished name identifying the root context for ldap to search for groups and users.
- contextFactory : fully qualified name of factory class allowing to create the jndi initial context used for ldap search (usually : com.sun.jndi.ldap.LdapCtxFactory).
- stripGroupNames : boolean specifying if the groups obtained from Active Directory must be in short form (stripped) or long form (LDAP full name). For example, on the domain MY.DOMAIN.COM, a group placed in the organization unit “groups” and named “mygroup” will have :
    - LDAP full name : *CN=mygroup, OU=groups, DC=my, DC=domain, DC=com*
    - stripped name : *mygroup*

5. Configuring SpnegoValve in Tomcat

Open the file *context.xml* located in the directory “*conf*” of tomcat installation. Under the section “*Context*”, add a “*Valve”* entry. Taking in account the same sample as in the previous point, you will add the following lines to the file :

```
<Valve className="org.jaaslounge.sso.tomcat.spnego.SpnegoValve"
      domainController="adserver.my.domain.com"
      domainName="MY.DOMAIN.COM"
/>
```

The available options for SpnegoValve are :

- domainController : address of the domain controller.
- domainName : name of the domain.