How to install tomcatspnego module

How to install a tomcatspnego module for TCL authentication support for Apache Tomcat webserver

This document describes installation and configuration procedure for tomcatspnego authentication module for Apache Tomcat webserver on Windows platform. This module enables the thin client to use TCL authentication, i.e. the users in Windows domain (in intranet or connected to internal network via VPN) have access to D2000 application based on their login to domain, without a need to enter their username/password in a web browser.

This procedure presumes that the Apache Tomcat webserver is installed in directory *C:\Program Files\Apache Software Foundation\Tomcat 6.0.*

Installation procedure

1. Go to a web site of tomcatspnego project (http://tomcatspnego.codeplex.com) to section Download and download the current version (we used trunk-20022010.zip).
2. Extract a file which name begins with *trunk-dll* (we used *trunk-dll-20022010.zip*) from the downloaded file.
3. From this file extract dynamic libraries *dll\vc90\SSPAuthentification.dll* and *dll\vc90\SSPAuthentification64x.dll* and copy them to directory *C:\Program Files\Apache Software Foundation\Tomcat 6.0\bin*.

   Note: SSPAuthentification.dll will be used for 32-bit installation of Tomcat on 32 and 64 bit Windows, SSPAuthentification64x.dll will be used for 64-bit installation of Tomcat on 64 bit Windows. If there were problems with authentication, there are alternative versions *dll\PLatformSDK2003x64\SSPAuthentificationx64.dll* and *dll\vc2003\SSPAuthentification.dll* available.
4. Extract also the file *jar\tomcat6\frdoumesppitc6.jar* and copy it to directory *C:\Program Files\Apache Software Foundation\Tomcat 6.0\lib*.
5. If you didn't distribute your web application to a web server using the program TCLDeployer, please do it now. Further procedure presumes that the web application is located in a directory *C:\Program Files\Apache Software Foundation\Tomcat 6.0\webapps\mojaApp*.
6. Copy the extracted directory *example\authbysspi\META-INF* to the directory *C:\Program Files\Apache Software Foundation\Tomcat 6.0\webapps\mojaApp*. The extracted directory contains a file *context.xml* with text:

```
<?xml version='1.0' encoding='utf-8'?>
<Context>
  <Valve className="fr.doume.authenticator.SSPAuthenticator" />
  <Realm className="fr.doume.realm.WindowsRealm" />
</Context>
```

7. Insert a security section to a file *C:\Program Files\Apache Software Foundation\Tomcat 6.0\webapps\mojaApp\WEB-INF\web.xml*. The security section should be located at the end of the section <web-app>:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<web-app>

    ...
  <security-constraint>
<display-name>Example Security Constraint</display-name>
  <web-resource-collection>
    <web-resource-name>Protected Area</web-resource-name>
      <!-- Define the context-relative URL(s) to be protected -->
         <url-pattern>/idom.html</url-pattern>
         <url-pattern>/d2was_service2</url-pattern>
      <!-- If you list http methods, only those methods are protected -->
           <http-method>DELETE</http-method>
         <http-method>GET</http-method>
         <http-method>POST</http-method>
         <http-method>PUT</http-method>
      </web-resource-collection>
      <auth-constraint>
             <!-- Anyone with one of the listed roles may access this area -->
                  <role-name>MYDOMAIN\MYGRP</role-name>
      </auth-constraint>
   </security-constraint>

<!-- Default login configuration -->
   <login-config>
      <auth-method>BASIC</auth-method>
         <realm-name>Example Spnego</realm-name>
   </login-config>

   <security-role>
      <role-name>MYDOMAIN\MYGRP</role-name>
   </security-role>
</web-app>
```

   To logon to a domain use *idom.html* instead of *index.html* in URL.

   Example:
   `http://hostname[:port]/mojaApp/idom.html[?alias]`

Name of domain and user group *MYDOMAIN\MYGRP* replace by a name of your domain and your user group, which contains users who are permitted to access the web application using TCL authentication.

Note: It is possible to specify domain and user group *MYDOMAIN\everyone* or only *everyone*, so the authentication against the web server will be successful for all domain users . It may simplify the management of users and from the security point of view it is not a significant risk, because consequently the D2000 Server checks if the user exists in the configuration of application and if TCL authentication is permitted in user's Authentication methods.

8. If you didn't create a user group on your domain server (in our example *MYGRP*), you can do it now.
9. All users who should be able to use TCL authentication, must have this option permitted in the configuration of Authentication methods. The name of D2000 user and domain user must exactly match.
10. The name of Windows domain must be configured in parameter Domain.

Note 1: To enable TCL authentication in the browser Mozilla Firefox (starting with version 3.0), it needs to be configured to support Kerberos authentication for a specific web server:

1. Go to the URL address about:config. If a warning is displayed, confirm it.
2. In the filter specify the mask *network.negotiate*
3. Modify the settings *network.negotiate-auth.delegation-uris* and *network.negotiate-auth.trusted-uris* - add the name of your web server for which Firefox should use Kerberos authentication, which is a basis for TCL authentication. If your application is located on several web servers or you use several applications, the names of servers should be separated by comma.
   Example: *myweb1,myweb2,myapp3*

Note 2: After adding a domain user to a user group (in our example *MYGRP*) the user must log-out and log-in to Windows to be effectively in the group (as the credentials are obtained during logon process). In the meantime the user won't be successfully authenticated using TCL authentication and usual logon window prompting for user name and password will be displayed.

Related pages:

D2000 Thin Client
D2000 Thin Client installation