CAS 系列

跨域 SSO 实例安装和配置指南

作者：Barry
QQ：394688159

版本	日期	作者	更改简述
0.1	2009. 5.1	Barry	初版
0.2	2009.5.2	Barry	增加配置 LDAP 验证部分
0.3	2009.5.3	Barry	增加集成 Oracle Berkeley DB 部分

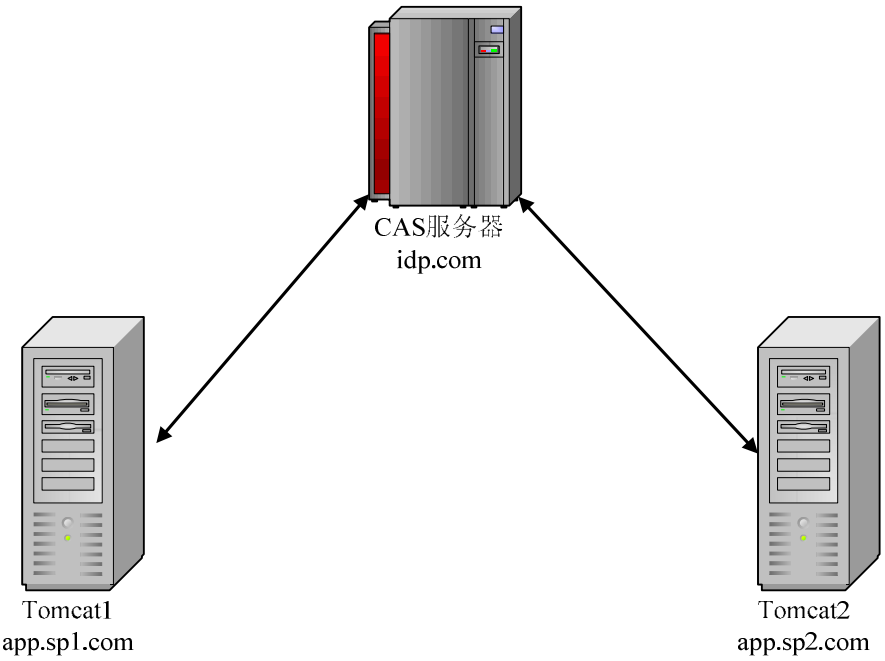
目录

前 言

此文档的目的就是为了帮助初涉 CAS 的人员提供一个真实测试实例，让大家很快了解 CAS 是如何工作，SSO 是怎么一回事。由于自己以前主要研究对象是商业产品，而对这个开源流行的 CAS SSO 解决方案知之甚少，因此抽点时间好好研究一下。

此文档之所以叫“CAS 跨域 SSO 实例安装和配置指南”，是常常有人会问 CAS 的 SSO 是否支持跨 Cookie 域的问题，其实只要在 SP(Service Provider, 就是部署应用程序的服务器)配置用同一个 IDP（Identity Provider, 就是提供用户验证的服务器，在此例里指的就是 CAS 服务器）来验证就会跨域。这是由 CAS 协议的实现机制所决定，并没有太多配置工作要做。

推荐在全新的操作系统上开始，就是没有 JDK/JRE 和 Tomcat。本文演示是在 Windows XP Professional。 为了方便讲解，也为了更好说明配置过程，这里需要准备三台机器（如果你理解好了，在一台机器上也可以），建议您把自己的主机当作 CAS 服务器（idp.com）。然后，用 VMWare 开了两个虚拟机（app.sp1.com 和 app.sp2.com）。基本部署结构如下图：



由于只是测试目的，一般没有必要配置 DNS，用修改 hosts 文件的方法添加域名最方便。请自行修改 hosts 文件。要保证在两个 SP (app.sp1.com 和 app.sp2.com) 能正常访问 IDP（idp.com）。

一. 安装 JDK

- 在 java.sun.com 网站下载 jdk-1_5_0_18-windows-i586-p.exe，然后分别在三台机器上运行这个 exe 文件，选择典型模式进行安装。
- 分别设置三个机器的环境变量 JAVA_HOME 指向 C:\Program Files\Java\jdk1.5.0_18。如果你的 JDK 没有安装在这个目录，请自己根据情况修改。

二. 使用 keytool 工具为演示生成一个自签名证书

1. 在 idp.com 机器上，打开一个命令行窗口或者直接在运行里输入“cmd”，然后按照以下步骤进行操作：

```
C:\Documents and Settings\barry>cd "\Program Files"
C:\Program Files>cd java
C:\Program Files\Java>cd jdk1.5.0_18
C:\Program Files\Java\jdk1.5.0_18>cd bin
C:\Program Files\Java\jdk1.5.0_18\bin>
C:\Program Files\Java\jdk1.5.0_18\bin>keytool -genkey -alias idp
-keypass changeit -keyalg RSA
输入 keystore 密码:  changeit
您的名字与姓氏是什么?
  [Unknown]:  idp.com
您的组织单位名称是什么?
  [Unknown]:  develop department
您的组织名称是什么?
  [Unknown]:  my company
您所在的城市或区域名称是什么?
  [Unknown]:  beijing
您所在的州或省份名称是什么?
  [Unknown]:  changping
该单位的两字母国家代码是什么
  [Unknown]:  CN
CN=idp.com, OU=develop department, O=my company, L=beijing,
ST=changping, C=CN
正确吗?
  [否]:  y
```

```
C:\Program Files\Java\jdk1.5.0_18\bin>keytool -export -alias idp
-keypass changeit -file server.crt
输入 keystore 密码:  changeit
保存在文件中的认证 <server.crt>
```

这个操作目的是为 idp.com 生成一个自签名证书。**特别注意**：加重的部分，在这个例子中，必须是：idp.com，这个值必须要与 CAS 服务器的域名相匹配。

2. 把生成的 server.crt 文件分别拷贝到 app.sp1.com 和 app.sp2.com 机器的 C:\Program Files\Java\jdk1.5.0_18\bin 目录里。
3. 分别在 app.sp1.com 和 app.sp2.com 机器上，打开一个命令行窗口或者直接在运行里输入“cmd”，然后按照以下步骤进行操作：

```
C:\Documents and Settings\barry>cd "\Program Files"
C:\Program Files>cd java
C:\Program Files\Java>cd jdk1.5.0_18
C:\Program Files\Java\jdk1.5.0_18>cd bin
C:\Program Files\Java\jdk1.5.0_18\bin>
C:\Program Files\Java\jdk1.5.0_18\bin>keytool -import -file
server.crt -keypass changeit -keystore ..\jre\lib\security\cacerts
输入 keystore 密码:  changeit
Owner: CN=idp.com, OU=develop department, O=my company, L=beijing,
ST=changping,
 C=CN
发照者: CN=idp.com, OU=develop department, O=my company, L=beijing,
ST=changpin
g, C=CN
序号: 49f6c67a
有效期间: Tue Apr 28 17:03:54 CST 2009 至: Mon Jul 27 17:03:54 CST 2009
认证指纹:
        MD5:  1C:69:80:70:D3:E2:3E:EE:E4:6F:C8:18:A6:35:5E:1B
        SHA1:
96:D0:EF:3A:D4:5F:E7:C8:9E:B5:9C:46:85:19:6E:A3:E0:56:1E:76
信任这个认证? [否]:  y
认证已添加至 keystore 中
```

三. 安装 tomcat

1. 下载 tomcat5.5.23, 网址:
   http://archive.apache.org/dist/tomcat/tomcat-5/v5.5.23/bin/apache-tomcat-5.5.23.zip
2. 然后把下载的 zip 文件分别拷到 idp.com, app.sp1.com 和 app.sp2.com 三台机器上，并且分别解压到“C:\tomcat5.5.23”目录下。
   如果下载的是 exe 文件，在安装过程请安装到“C:\tomcat5.5.23”目录下, 并且指定使用的 JRE 为: “%JAVA_HOME%\jre”。

   注意: 指定 Tomcat 的 JRE 一定要指向正确。
3. 验证 tomcat 是否能正常运行。在三台机器分别做以下操作: 运行 C:\tomcat-5.5.23\bin\startup.bat 文件，查看日志有没有错误。然后在浏览器里输入: http://localhost:8080, 如果能看猫的标志，就表示安装 tomcat 成功。

四. 配置 idp.com 上的 tomcat server.xml

在 idp.com 机器上，打开 C:\tomcat-5.5.23\conf\server.xml， 然后找到

```
<!--
<Connector port="8443" maxHttpHeaderSize="8192"
        maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
        enableLookups="false" disableUploadTimeout="true"
        acceptCount="100" scheme="https" secure="true"
        clientAuth="false" sslProtocol="TLS" />
-->
```

把这部分替换成如下内容:

```
<Connector port="8443" maxHttpHeaderSize="8192"
        maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
        enableLookups="false" disableUploadTimeout="true"
        acceptCount="100" scheme="https" secure="true"
        clientAuth="false" sslProtocol="TLS"
        keystoreFile="C:/Documents and Settings/barry/.keystore"
        keystorePass="changeit"
        truststoreFile="C:/Program
Files/Java/jdk1.5.0_18/jre/lib/security/cacerts" />
```

此操作是为了启动 idp 上的 https。

五. 为两个 SP 上 HelloWorld Servlet 配置 CAS Client 过滤器

1. 分别在两个 SP 机器上打开

   C:\tomcat-5.5.23\webapps\servlets-examples\WEB-INF\web.xml 文件。在此文件里添加如下内容：

```
<filter>
    <filter-name>CAS Filter</filter-name>
    <filter-class>edu.yale.its.tp.cas.client.filter.CASFilter</filter-class>
    <init-param>
        <param-name>edu.yale.its.tp.cas.client.filter.loginUrl</param-name>
        <param-value>https://idp.com:8443/cas/login</param-value>
    </init-param>
    <init-param>
        <param-name>edu.yale.its.tp.cas.client.filter.validateUrl</param-name>
        <param-value>https://idp.com:8443/cas/serviceValidate</param-value>
    </init-param>
    <init-param>
        <param-name>edu.yale.its.tp.cas.client.filter.serverName</param-name>
        <param-value>app.sp.com:8080</param-value>
    </init-param>
</filter>

<filter-mapping>
    <filter-name>CAS Filter</filter-name>
    <url-pattern>/servlet/HelloWorldExample</url-pattern>
</filter-mapping>
```

   注意：加重部分在不同的两个 SP 机器上是不一样的。app.sp1.com 机器上应该为:app.sp1.com，而在 app.sp2.com 机器上应该为:app.sp2.com。

2. 下载 CAS 客户端 jar 文件。网址：

   http://www.ibiblio.org/maven/cas/jars/casclient-2.1.1.jar。把这个文件分别放在 app.sp1.com 和 app.sp2.com 机器上的 C:\tomcat5.5.23\webapps\servlets-examples\WEB-INF\lib 目录下，如果这个目录不存在，你可以自己手动创建一个。

六. 在 IDP 上下载和部署 CAS 服务器

1. 在 idp.com 机器上下载 CAS Server 3.3.2。网址：

   http://www.ja-sig.org/downloads/cas/cas-server-3.3.2-release.zip。

2. 解压到一个目录里。解压之后可以在 modules 目录里找到 cas-server-webapp-3.3.2.war 文件， 把这个文件拷贝到 C:\tomcat-5.5.23-compA\webapps 目录里，并且重命名为 cas.war。

   注意： 这里重命名是必要的，至于叫什么名字，与第五步里网址 (https://idp.com:8443/**cas**/login)里的加重部分一致就可以。

3. 重新启动 IDP 上的 tomcat,查看日志，日志里不应该有错误报告。

七. 测试 SSO

1. 测试 SSO 之前，把三台机器上的 tomcat 都启动。并且保证在 app.sp1.com 和 app.sp2.com 上可以正常访问到 idp.com（这个域名必须能够工作）机器的 8080 端口。可以分别在两个 SP 的机器上用如下命令来测试：

```
ping idp.com
```

   如果这个命令都执行正常，没有超时现象，说明这个域名正常工作，并且都可达。

2. 再找一个机器（可以是三台机器的任一台，或者其他机器）。但要先做如下测试：

```
ping idp.com
ping app.sp1.com
ping app.sp2.com
```

   如果三个命令都能执行正常，没有超时现象，说明这三个域名都正常工作，并且都可达。

3. 打开一个新的浏览器，输入：http://app.sp1.com:8080/servlets-examples/servlet/HelloWorldExample，你将被要求输入用户名和密码，输入 barry/barry 或者其他的两个相同的值。如果能正常显示“Hello World”信息就表示你已经成功登录。再尝试访问：http://app.sp2.com:8080/servlets-examples/servlet/HelloWorldExample，应该会有所不同，不再要求登录。表示你的 SSO 已经工作正常。

4. 关闭浏览器，把上一步的两个网址的先后顺序换一下，再测试一次。

八. 配置 LDAP

1. 安装和配置 LDAP 服务器。本人喜欢使用 Sun Directory Server.5.2.P4。如果没有请自行从 Sun 的网站上下载并且安装。

2. 下载 Spring-LDAP 的支持包，并自行解压。CAS3 以后把自己支持 LDAP 的类已经删除，所以这个 Sping-LDAP 是必须的。

3. 在第五步里 modules 目录里找到 cas-server-support-ldap-3.3.2.jar 文件，并且把这个文件拷贝到 IDP 机器的 C:\tomcat-5.5.23\webapps\cas\WEB-INF\lib 里，如果 CAS 服务器没有部署在这个目录里，请自行修改。
4. 在刚刚下载的 Sping-LDAP 里找到 spring-ldap-1.3.0.RELEASE-all.jar 文件，并且把这个文件拷贝到 IDP 机器的 C:\tomcat-5.5.23\webapps\cas\WEB-INF\lib 里，如果 CAS 服务器没有部署在这个目录里，请自行修改。
5. 在 IDP 服务器上，打 C:\tomcat-5.5.23\webapps\cas\WEB-INF、deployerConfigContext.xml 文件， 添加如下内容：

```
 <bean id="contextSource"
class="org.springframework.ldap.core.support.LdapContextSource">
      <property name="pooled" value="true"/>
      <property name="urls">
          <list>
              <value>ldap://localhost:389/</value>
          </list>
      </property>
      <property name="userDn" value="cn=Directory Manager"/>
      <property name="password" value="password"/>
      <property name="baseEnvironmentProperties">
          <map>
              <entry>
                  <key>
                      <value>java.naming.security.authentication</value>
                  </key>
                  <value>simple</value>
              </entry>
          </map>
      </property>
 </bean>
```

注意：上面内容中加重部分，请根据自己的情况修改。

6. 在此 xml 文件里找到如下内容：

```
<bean
 class="org.jasig.cas.authentication.handler.support.SimpleTestUsernamePasswordAut
henticationHandler" />
```

用以下内容替换：

```
<bean
    class="org.jasig.cas.adaptors.ldap.BindLdapAuthenticationHandler">
    <property name="filter" value="uid=%u" />
    <property name="searchBase" value="ou=people,dc=ttg,dc=cc" />
```

```
    <property name="contextSource"  ref="contextSource" />
</bean>
```

注意：上面内容中加重部分，请根据自己的情况修改。

7. 重新启动 IDP 上的 Tomcat，如果日志没有错误报告，就说明 LDAP 配置完毕。
8. 重复以上第七步里测试 SSO 步骤。

九. 与 Oracle Berkeley DB 集成

默认的 TicketRegistry 是在内存里建一个映射表，这样在大量访问的时候，Ticket 比较多。这样就要更多的内存，从而影响其他的应用程序。所以大量访问的网站，最好把 TicketRegistry 不要使用默认的实现。这种情况，Oracle Berkeley DB 嵌入式数据库就是一个不错的选择。

1. 在 Oracle 的网站下载 Berkeley JE。网址：http://www.oracle.com/technology/software/products/berkeley-db/htdocs/popup/je/3.3.75/je-zip.html。
2. 下载之后，解压。在里面能够找一个文件 je-3.3.75.jar。把些文件拷贝到 IDP 机器的 C:\tomcat-5.5.23\webapps\cas\WEB-INF\lib 目录下。
3. 在第五步里 modules 目录里找到 cas-server-integration-berkeleydb-3.3.2.jar 文件，并且把这个文件拷贝到 IDP 机器的 C:\tomcat-5.5.23\webapps\cas\WEB-INF\lib 里
4. 创建一个目录 C:\tomcat-5.5.23-compA\webapps\cas\ticketDB 用来存储 DB 文件。
5. 打开文件 C:\tomcat-5.5.23-compA\webapps\cas\WEB-INF\spring-configuration\ticketRegistry.xml，修改之后看起来内容如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:p="http://www.springframework.org/schema/p"
      xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-2.0.xsd">
   <description>
      Configuration for the default TicketRegistry which stores the tickets in-memory
and cleans them out as specified intervals.
   </description>

   <!-- Ticket Registry -->
 <!--bean id="ticketRegistry"
class="org.jasig.cas.ticket.registry.DefaultTicketRegistry" /-->
 <bean id="ticketRegistry"
class="org.jasig.cas.ticket.registry.BerkeleyDbTicketRegistry">
      <property name="dbHome" value="ticketDB"/>
 </bean>
```

```
 <!--Quartz -->
 <!-- TICKET REGISTRY CLEANER -->
 <bean id="ticketRegistryCleaner"
class="org.jasig.cas.ticket.registry.support.DefaultTicketRegistryCleaner"
      p:ticketRegistry-ref="ticketRegistry" />

 <bean id="jobDetailTicketRegistryCleaner"
class="org.springframework.scheduling.quartz.MethodInvokingJobDetailFactoryBean"
      p:targetObject-ref="ticketRegistryCleaner"
      p:targetMethod="clean" />

 <bean id="triggerJobDetailTicketRegistryCleaner"
class="org.springframework.scheduling.quartz.SimpleTriggerBean"
      p:jobDetail-ref="jobDetailTicketRegistryCleaner"
      p:startDelay="20000"
      p:repeatInterval="5000000" />

 <bean id="scheduler"
class="org.springframework.scheduling.quartz.SchedulerFactoryBean">
      <property name="triggers">
           <list>
                <ref
                     local="triggerJobDetailTicketRegistryCleaner" />
           </list>
      </property>
 </bean>
</beans>
```

重点注意加重部分。我看资料显示 Berkeley DB 嵌入式数据库是高性能，并且能备份，能恢复，我看了一下源码，好像没有提供这些功能。估计是用不到这些特性。主要是解决 Ticket 存在内在的问题。

6. 重新启动 IDP 上的 Tomcat, 如果日志没有错误报告，就说明集成 Berkeley DB 的配置完毕。
7. 重复以上第七步里测试 SSO 步骤。

恭喜，SSO 工作正常。接下来，我会仔细研究一下 CAS Server 的源码并且随时更新此文档。