MidPoint as CAS Client (Apache CAS Agent Method)

Unofficial functionality

This functionality requires modification of midPoint build, or even modification of midPoint source code. Therefore it is not officially supported - unless the support is explicitly negotiated in subscription.
The real solution to this problem would be Flexible Authentication. MidPoint platform subscription could be used to fund improvements in midPoint authentication mechanisms.

Thanks to Jason Everling for contributing this HOWTO

Assumptions

CAS Usernames must match midPoint user "name".

In this example I am using Apache with Tomcat 7, auth-cas and mod-jk

Apache installed and configured with SSL

Tomcat installed and configured working already with midPoint

Apache Configuration

```
sudo apt-get install libapache2-mod-jk libapache2-mod-auth-cas
```

Configure mod-jk

Create a workers.properties file in /etc/apache2

```
sudo vi /etc/apache2/workers.properties
```

Add the following

```
worker.list=worker1
worker.worker1.port=8009
worker.worker1.host=localhost
worker.worker1.type=ajp13
```

Configure apache2 sites

```
sudo vi /etc/apache2/sites-available/default-ssl.conf
```

Add the following below the first default DocumentRoot /var/www/html

```
<Location ~ "/midpoint*">  AuthType CAS
  AuthName "CAS"
  require valid-user
  CasAuthNHeader Cas-User
 </Location>

JkMount /midpoint* worker1
```

Configure auth-cas

```
sudo vi /etc/apache2/mods-available/auth_cas.conf
```

Add the following

```
CASCookiePath /var/cache/apache2/mod_auth_cas/
CASLoginURL https://SERVERURL/cas/login
CASValidateURL https://SERVERURL/cas/serviceValidate
CASDebug Off
CASValidateServer On
CASVersion 2
CASSSOEnabled On
#Below
 is needed, auth-cas will use the server hostname in the service URL
redirect so we will override that, do not add a trailing / or add
/midpoint!
CASRootProxiedAs https://MIDPOINTSERVERURL
```

Restart Apache2

```
sudo service apache2 restart
```

Tomcat Configuration

Confgure tomcat to use the AJP connector

```
sudo vi /var/lib/tomcat7/conf/server.xml
```

Uncomment the following so that it reads

```
<!-- Define an AJP 1.3 Connector on port 8009 -->
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
```

Midpoint Configuration

Edit ctx-web-security.xml

```
sudo vi /var/lib/tomcat7/webapps/midpoint/ctx-web-security.xml
```

Uncomment the following so that reads

```
<!-- For SSO integration use the following: -->
<custom-filter position="PRE_AUTH_FILTER" ref="requestHeaderAuthenticationFilter" />
```

Edit the following value "principalRequestHeader" in the bean "requestHeaderAuthenticationFilter" so that it reads

```
    <!-- Following bean is used with pre-authentication based on HTTP headers (e.g. for SSO integration) -->
    <beans:bean
 id="requestHeaderAuthenticationFilter"
class="org.springframework.security.web.authentication.preauth.RequestHeaderAuthenticationFilter">
     <beans:property name="principalRequestHeader" value="Cas-User"/>
     <beans:property name="authenticationManager" ref="authenticationManager" />
 </beans:bean>

 <beans:bean id="logoutHandler" class="com.evolveum.midpoint.web.security.AuditedLogoutHandler">       <beans:
property name="defaultTargetUrl" value="https://SERVERURL/cas/logout"/>
    </beans:bean>
```

Finally restart tomcat7

```
sudo service tomcat7 restart
```

User can now login to midPoint using CAS

See Also

- MidPoint and SSO HOWTO