

# Sun Access Manager CAC Authentication Deployment Configuration Guide

For

Access Manager 7.1 Single War Deployment into  
GlassFish Enterprise Application Server

**Author:** Jeff Nester  
Sun Microsystems  
*jeff.nester@sun.com*

**Version:** 1.0

**Date:** 9/15/2008



# *Table of Contents*

1	Introduction .....	3
1.1	Assumptions .....	3
2	Configuring CAC Authentication .....	3
2.1	Modify the AMConfig.properties file .....	3
2.2	Create the CAC Authentication Module and Chain .....	4
3	Configuring the Application Server .....	6
3.1	Modify the SSL Listener .....	6
3.2	Modify the domain.xml file and load the jss4.jar file.....	7
3.3	Load the OCSP Signing Certificate and DoD CA PKI Root Certificate Authorities Certificates .....	8
4	Troubleshooting.....	8
4.1	Troubleshooting thought 1 - Using telnet.....	8
4.2	Troubleshooting thought 2 - Using Open SSL.....	9

## Document Revisions

<b>Date</b>	<b>Editor</b>	<b>Description of Change</b>
9/15/08	Jeff Nester	Original Document Created

# 1 Introduction

In this series of documents I have been describing the entire start to finish process of install the software, deploying and configuring Access Manager and the necessary changes required to support CAC Authentication with Access Manager. This document assumes that you have an installed and working the GlassFish Enterprise Application Server and Sun Access Manager 7.1. **Note:** The GlassFish Enterprise Application Server is the renamed Sun Java Enterprise System Enterprise Application Server 9.1.

If your implementation involves the use of Access Manager Policy Agents it will be necessary to configure a second SSL listener without the Client Certificate “Required option” being set. The policy agent must be able to log into the Access Manager for it to function. Since the SSL listener configured in this document requires a certificate the policy agent will **NOT** be allowed to communicate with Access Manager. I will be developing another document later that will describe options to resolve this issue.

## 1.1 Assumptions

1. In this document the example assumes that the server being installed is **sedemo1identric.com**. This reference should be replaced with your specific server name.
2. It is also assumed that the GlassFish Enterprise Application Server was installed in `/opt/SUNWappserver`
3. This document assumes that Access Manager is already running and configured to use SSL.

# 2 Configuring CAC Authentication

In the following instructions it is assumed that the server that is being used is **sedemo1.identric.com**.

## 2.1 Modify the AMConfig.properties file

1. Login in as a super user to the Access Manager server.
2. Make a backup of the AMConfig.properties file. Do this by:
  - `cd /etc/opt/SUNWam`
  - `cp AMConfig.properties AMConfig.properties.bck`
3. The following two lines must be changed in order for Access Manager to locate the OCSP Signing Certificate and the OCSP server. Locate and modify the following 3 lines (the nickname attribute must be set to the name of the nickname used with the OCSP signing certificate is loaded into the certificate database):

```
com.sun.identity.authentication.ocspCheck=true
com.sun.identity.authentication.ocsp.responder.url=
com.sun.identity.authentication.ocsp.responder.nickname=
```

to look like: (**Note:** the nickname must match the value used when the OCSP signing certificate was loaded)

```
com.sun.identity.authentication.ocspCheck=true
com.sun.identity.authentication.ocsp.responder.url=http://ocsp.disa.mil
com.sun.identity.authentication.ocsp.responder.nickname=DoDocspCertificate
```

4. Change the following line:

```
com.iplanet.security.SecureRandomFactoryImpl=com.iplanet.am.util.SecureRandomFactoryImpl
```

to

```
com.iplanet.security.SecureRandomFactoryImpl=com.iplanet.am.util.JSSSecureRandomFactoryImpl
```

5. Change the following line:

```
com.iplanet.security.SSLSocketFactoryImpl=netscape.ldap.factory.JSSESocketFactory
```

to

```
com.iplanet.security.SSLSocketFactoryImpl=com.iplanet.services.ldap.JSSSocketFactory
```

6. Change the following line:

```
com.iplanet.security.encryptor=com.iplanet.services.util.JCEEncryption
```

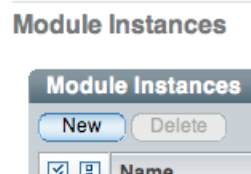
to

```
com.iplanet.security.encryptor=com.iplanet.services.util.JSSEncryption
```

7. Before these changes take affect the Application Server must be restarted. If you are continuing through the configuration you can restart the application server after the changes to the Application Server are completed.

## 2.2 Create the CAC Authentication Module and Chain

8. Login to the Access Manager console at <https://sedemo1.identric.com/amserver/console> using the user **amadmin** and the password configured during the installation.
9. Create the CAC Authentication Module by clicking on the **Access Control** Tab and then click on the hyperlink for the realm.
10. Click on the **Authentication** Tab and then click on the **New** button under **Module Instances**.



11. On the screen that is displayed specify the name of the module, **CAC**, and select certificate and then click on the **OK** button:

### New Module Instance

\* Name:

\* Type:  Active Directory  
 Anonymous  
 Certificate  
 Data Store  
 ...

12. Select the hyperlink of the newly created Module Instance. Once the page is displayed the only thing that must change is the **OCSP Validation** needs to be enabled. Once enabled click the **Save** button and then **Back to Authentication**:

### Certificate

#### Realm Attributes

Match Certificate in LDAP:	<input type="checkbox"/> Enabled
Subject DN Attribute Used to Search LDAP for Certificates:	<input type="text" value="CN"/>
Match Certificate to CRL:	<input type="checkbox"/> Enabled
Issuer DN Attribute Used to Search LDAP for CRLs:	<input type="text" value="CN"/>
HTTP Parameters for CRL Update:	<input type="text"/>
OCSP Validation:	<input checked="" type="checkbox"/> Enabled

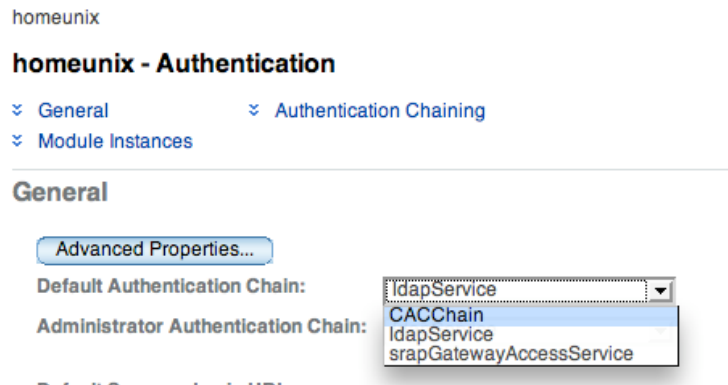
13. Create a new chain for the CAC Module. This is done by clicking on the **New** button under **Authentication Chaining**.
14. In the new window specify the Name **CACChain** and then click the **OK** button.
15. On the next screen click the **Add** button and then select the module **CAC** and mark it as **Required** and

### CACChain - Properties

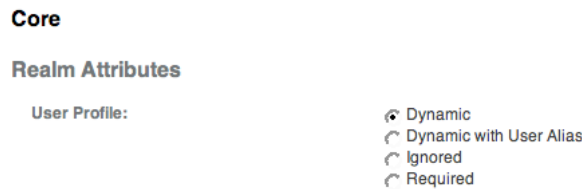
(1 Items)	
Instance	Criteria
<input checked="" type="checkbox"/> <input type="text" value="CAC"/>	REQUIRED

then click the **Save** button then **Back to Authentication**.

16. Next make the newly created chain the Default Authentication Chain by selected **CACChain** in the **Default Authentication Chain** and then click **Save**:



17. Finally click on the **Advanced Properties** button and change **User Profile** to **Dyamic**. Then click the **Save** button:



18. Restart the Web Server and when the web server restarts CAC Authentication should be enabled with OCSP verification. When a user authenticates with their CAC a new profile will be added to Access Manager

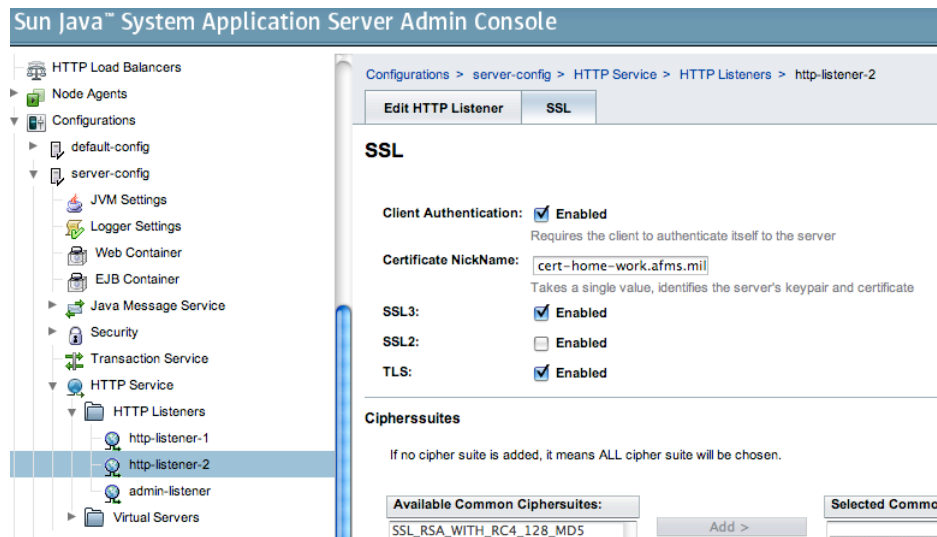
## 3 Configuring the Application Server

### 3.1 Modify the SSL Listener

In order for the Application Server to request the certificate from the CAC card, the SSL Listener hosting Access Manager must be configured to obtain the certificate. Use the application server admin console to configure the SSL Listener for "Client Certificate Required." This is done by doing the following:

1. Login to the application server admin console at <https://sedemo1.identric.com:4848> using the user admin and password established during installation

- Drill down to the **http-listener-2**. This is done by going to Configuration → HTTP Service → HTTP Listeners → http-listener-2. Once there select the SSL tab in the right hand pane and check the **Client Authentication Enabled** box and then click **Save**. For example:



### 3.2 Modify the domain.xml file and load the jss4.jar file

A jvm-option must be added to the domain.xml file for the Application Server. This can be done by doing the following:

- `cd /opt/SUNWappserver/domains/domain1/config`
- `cp domain.xml domain.xml.beforeCAC`
- vi the domain.xml file and search for the following line:

```
<jvm-options>-Djava.security.auth.login.config=${com.sun.aas.instanceRoot}/config/login.conf</jvm-options>
```

immediately after this line add the following:

```
<jvm-options>-Djava.protocol.handler.pkgs=com.ipplanet.services.comm</jvm-options>
```

- Search for the line in step 3 again. If it is found then place the jvm-option value immediately after it. Repeat this until every occurrence is followed with the jvm-option.

Next we need to place the jss4.jar file in the amserver lib directory. The jss4.jar file is available from <http://jeffnester.com/downloads/jss4.jar> Do the following to load the library:

- `cd /opt/SUNWappserver/domains/domain1/applications/j2ee-modules/amserver/WEB-INF/lib`
- `cp {sourceaction}/jss4.jar .`

### 3.3 Load the OCSP Signing Certificate and DoD CA PKI Root Certificate Authorities Certificates

1. The OCSP server that is used to vet CAC cards required a signing signature. This signature must be loaded in the certificate database. Obtain this certificate from your source of certificates To load the OCSP signing certificate do the following:

- a. `cd /opt/SUNWappserver/domains/domain1/config`
- b. `certutil -A -n "DoDocspCertificate" -d . -i certificate.cer -t "CT,CT,CT"`

2. In order for the Application Server to request the necessary certificates from the CAC card the DoD CA PKI Root Certificates must be loaded into the certificate database. Obtain these appropriate certificates from your security resource. You now should install the DoD CA-11, DoD CA-12, DoD CA-13, DoD CA-14, DoD CA-15, DoD CA-16, DoD CA-17, DoD CA-18 and DoD CA-19 certificates. Do the following to load the certificates:

- c. `/opt/SUNWappserver/domains/domain1/config`
- d. `certutil -A -n DoDCA_11 -d . -i DoDCA_11.crl -t "CT,CT,CT"`
- e. repeat the above step for all of the necessary DoD CA Roots.

3. The Application Server must be restarted:

```
/opt/SUNWappserver/bin/asadmin stop-domain  
/opt/SUNWappserver/bin/asadmin start-domain -user admin domain1
```

## 4 Troubleshooting

The most difficult part of this configuration is getting the OCSP check to work properly. DISA requires a signing certificate be configured. This is described in the instructions above. For the OCSP check to work the Access Manager server must be able to access the **ocsp.disa.mil** server via port **80**.

### 4.1 Troubleshooting thought 1 - Using telnet

This can be tested by logging into the Access Manager server a type the following from the command line:

```
telnet ocsp.disa.mil 80
```

if the following occurs type **GET**:

```
Trying 164.235.15.70... (Note: maybe a different IP Address)  
Connected to ocsp.csd.disa.mil.  
Escape character is '^'
```

The response to the GET should look something like:

```
HTTP/1.1 400 Bad Request  
Cache-Control: no-cache  
Pragma: no-cache  
Content-Type: text/html; charset=utf-8  
Proxy-Connection: close  
Connection: close  
Content-Length: 690  
  
<HTML><HEAD>  
<TITLE>Request Error</TITLE>
```

```
</HEAD>
<BODY>
<FONT face="Helvetica">
<big><strong></strong></big><BR>
</FONT>
<blockquote>
<TABLE border=0 cellpadding=1 width="80%">
<TR><TD>
<FONT face="Helvetica">
<big>Request Error (invalid_request)</big>
<BR>
<BR>
</FONT>
</TD></TR>
<TR><TD>
<FONT face="Helvetica">
Your request could not be processed.
</FONT>
</TD></TR>
<TR><TD>
<FONT face="Helvetica">
This could be caused by a misconfiguration, or possibly a malformed request.
</FONT>
</TD></TR>
<TR><TD>
<FONT face="Helvetica" SIZE=2>
<BR>
For assistance, contact your network support team. Reference device SATX-DISA1
</FONT>
</TD></TR>
</TABLE>
</blockquote>
</FONT>
</BODY></HTML>
```

If you see this type of response then you have a good connection to the OCSP server. If you do NOT see this type of response you must determine why your network will not let you access this host and port.

## 4.2 Troubleshooting thought 2 - Using Open SSL

Another valuable tool for debugging the process is the use of the OpenSSL utility. To use this command you must have the DoD Root CA certificate for your CAC card (for example DoD-16), the OCSP signing certificate and the user certificate from the CAC card being tested. Run the command as shown below:

```
openssl ocsp -host ocsp.disa.mil:80 -issuer dod-16.cer -VAfile dod_ocsp_ss.cer -cert jeff.cer
```

If the certificates are all correct the Response from the OCPS server will be similar to:

```
Response verify OK
jeff.cer: good
This Update: Jul 23 05:36:53 2008 GMT
Next Update: Jul 30 05:36:53 2008 GMT
```