Common Access Card

On this page:

Basic Configuration

In order for Common Access Card (CAC) authentication to work, SSL must be enabled on the authentication server:

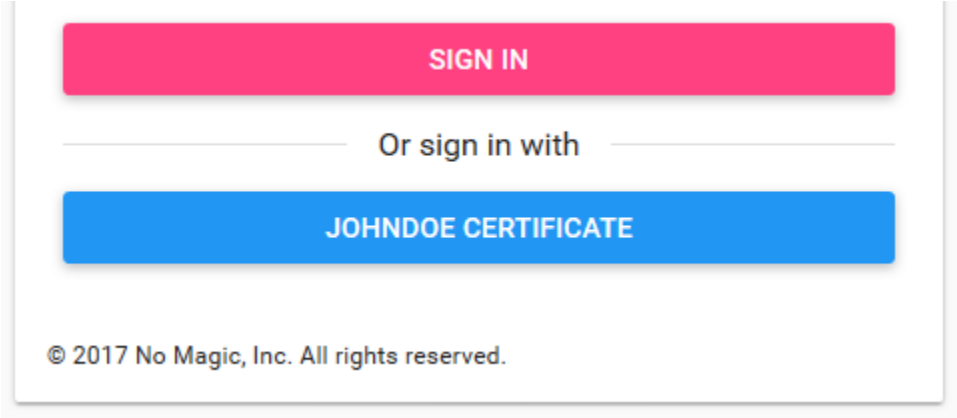
```
server.ssl.enabled=true
```

Next, you need to enable certificate authentication:

```
authentication.certificate.enabled=true
```

The next step is to configure which part of the subject DN (distinguished name) will be used as the username (**authentication.certificate.username.template**), and which part will be displayed in the login screen (**authentication.certificate.displayname.template**).

Both of these values default to using (CN)




```
authentication.certificate.username.template=(CN)
```

The template can contain ASCII characters as well as placeholders in parenthesis that are replaced with relative distinguished name (RDN) values from the DN.

For example, when the subject DN on the certificate is *CN=JohnDoe,O=MyCompany,C=GB*:

Template: *(CN)*, username: *JohnDoe*

Template: *(O)-(CN)*, username: *MyCompany-JohnDoe*

Template: *CERT_(CN)*, username: *CERT_JohnDoe*

To configure the value displayed in the login button, we must edit the **authentication.certificate.displayname.template** property.

```
authentication.certificate.displayname.template=(CN)
```

For example, as shown in the picture above, when the subject DN on the certificate is *CN=JohnDoe,O=MyCompany,C=GB*, and the display template is *(CN) CERTIFICATE*, the button will display „JOHNDOE CERTIFICATE“.

For a list of all the advanced properties available for configuration, please referrer to https://docs.nomagic.com/display/TWCloud190SP2/Authentication+by+certificate.

Trust Store Configuration

CAC integration requires that a trust store exist, containing the Certificate Authority (CA) certificates that issue the user's certificates.

The following parameter properties need to be configured:

```
server.ssl.trust-store=config/truststore.jks
server.ssl.trust-store-type=JKS
server.ssl.trust-store-password=secret
server.ssl.client-auth=want
```

There is no need to manually generate the truststore. Create a directory, named truststore, under AuthServer/config/ and place all of the CA certificates into it. Upon startup of the authentication server, if **truststore.jks** does not exist, it will be created by importing the CA certificates located in Authserver /config/truststore. If you make changes to the certificates in the truststore directory, delete **truststore.jks** and restart the authentication service. This will recreate the truststore with the current set of CA certificates.

Certificate Revocation List

The authentication server supports 2 methods of handling certificate revocation lists - via a URL, or via a local list stored in the file system. To enable this feature, uncomment either **authentication.certificate.revocation.list.url** or **authentication.certificate.revocation.list.file**, and point it to the location of the revocation list.

```
authentication.certificate.revocation.list.url=http://someserver.somedomain.com/revocation.lst
authentication.certificate.revocation.list.file=/opt/local/revovcation.lst
```