Creating a new SSL certificate on the First Node

Because the FQDN used to access the Client Access servers in our NLB cluster doesn't match the FQDN specified in the common name field nor the subject alternative names field in the default self-signed SSL certificate that automatically is installed on each Client Access server during Exchange 2007 setup (**Figure 3.5** and **3.6**), we must create a new certificate.

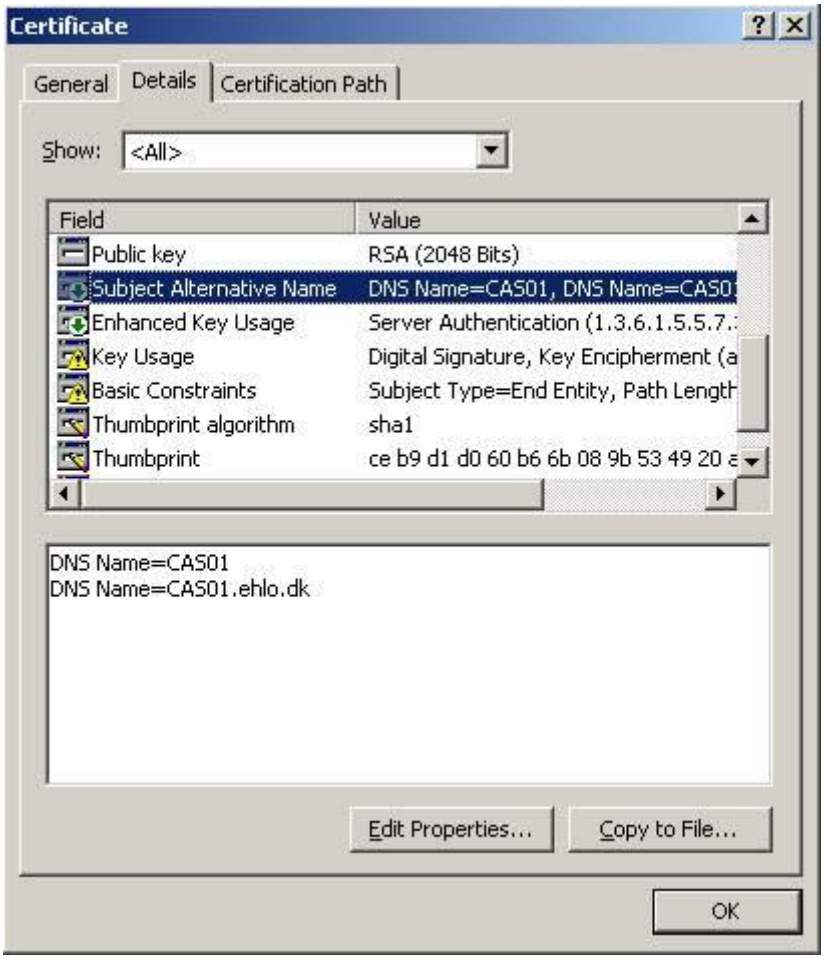

Figure 3.5: Subject Alternative Names on CAS01 Certificate Property page

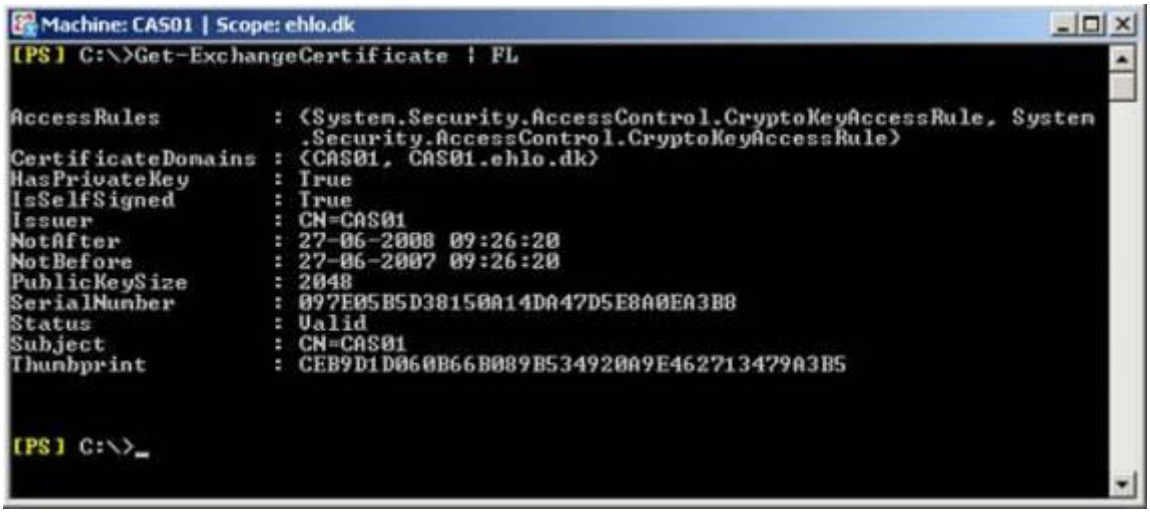

Figure 3.6: Subject Alternative Names on CAS01 via the Exchange Management Shell

For the purpose of this article series, we'll generate a new certificate using an internal Microsoft certificate authority server, but in a corporate production environment, you would in most situations want to submit the certificate request to a 3rd party certificate authority.

Note:
Because we need a certificate in which multiple FQDNs have to be specified, we must use a subject alternative name (SAN) certificate. At the time of this writing only a handful 3rd party CAs offer these types of certificates, most of which are listed in the following KB article: http://support.microsoft.com/kb/929395.

As we're going to generate a request for a new SAN certificate, we must use the New-ExchangeCertificate cmdlet for this purpose, as the IIS Manager isn't capable of creating requests for SAN certificates. To do this launch the Exchange Management Shell, then type the following command (replace the names with your own):

New-ExchangeCertificate –GenerateRequest –SubjectName "C=dk, O=EHLO organization, CN=mailehlo.dk" –DomainName mail.ehlo.dk, autodiscover.ehlo.dk, cas01.ehlo.dk, cas02.ehlo.dk –FriendlyName "CAS SAN Certificate" –KeySize 1024 –Path c:\CAS_SAN_cert.req –PrivateKeyExportable:$true

After hitting Enter, the thumbprint for the new certificate request will be listed as shown in **Figure 3.7**.

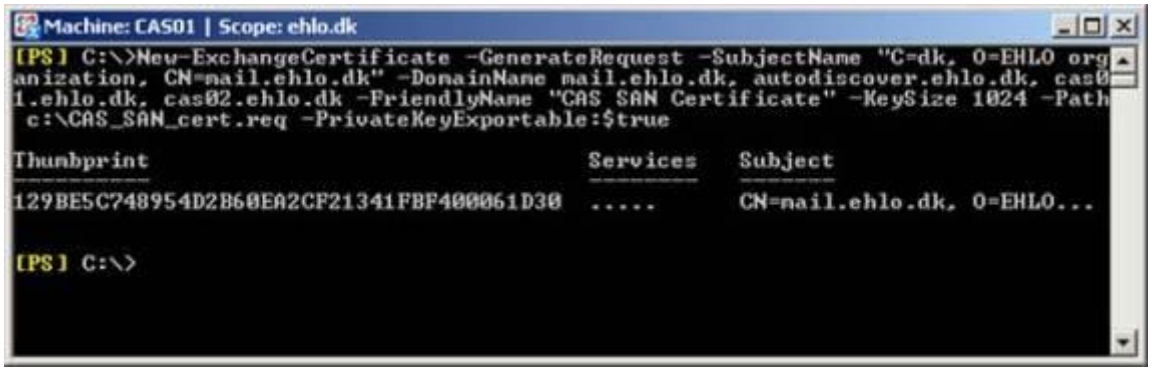

Figure 3.7: Generating a request for a new SAN Certificate

Submitting the SAN Certificate to a Microsoft Certificate Authority

With the SAN SSL certificate request generated, we can submit it to our Microsoft CA, or almost that is. The reason I why I say so, is because by default a Microsoft CA cannot handle certificates with the SAN field properly. To fix this issue log on to the Domain Controller and open a command prompt window, then type the following command:

Certutil –setreg policy\EditFlags +EDITF_ATTRIBUTESUBJECTALTNAME2

After hitting Enter, you should see the old and new value as in **Figure 3.8**.

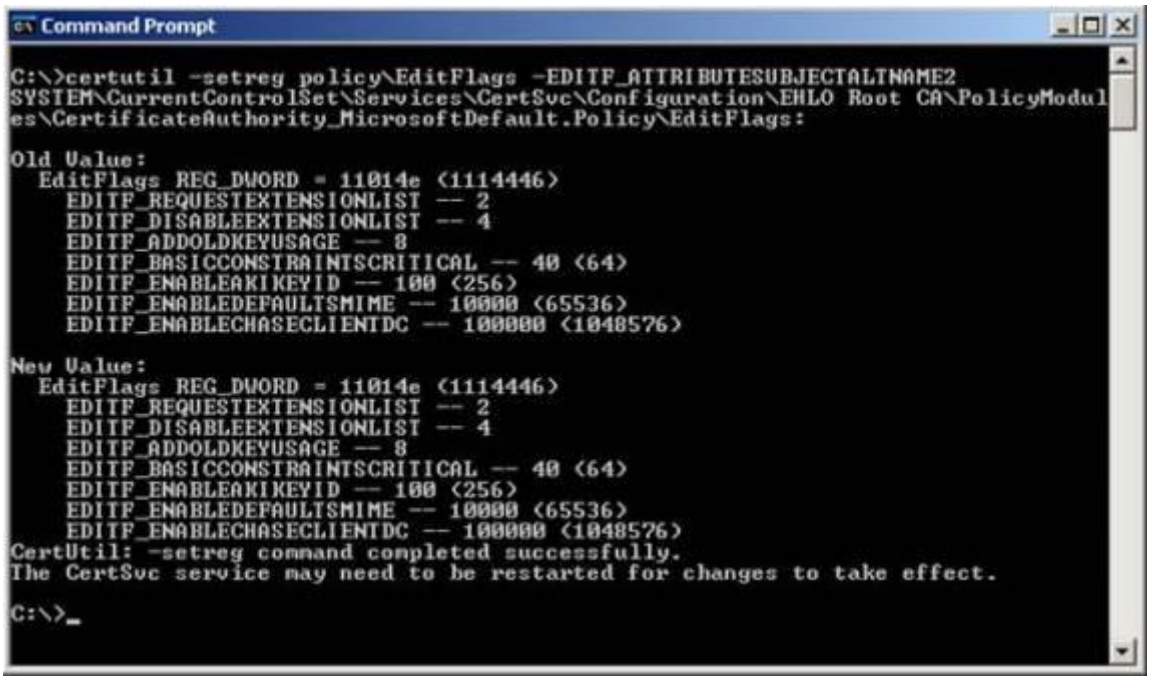

Figure 3.8: Changing the EditFlags on the Microsoft CA

Now restart Certificate Services (CertSVC) service on the Microsoft CA server (Domain Controller) in order to have the changes applied (**Figure 3.9**).

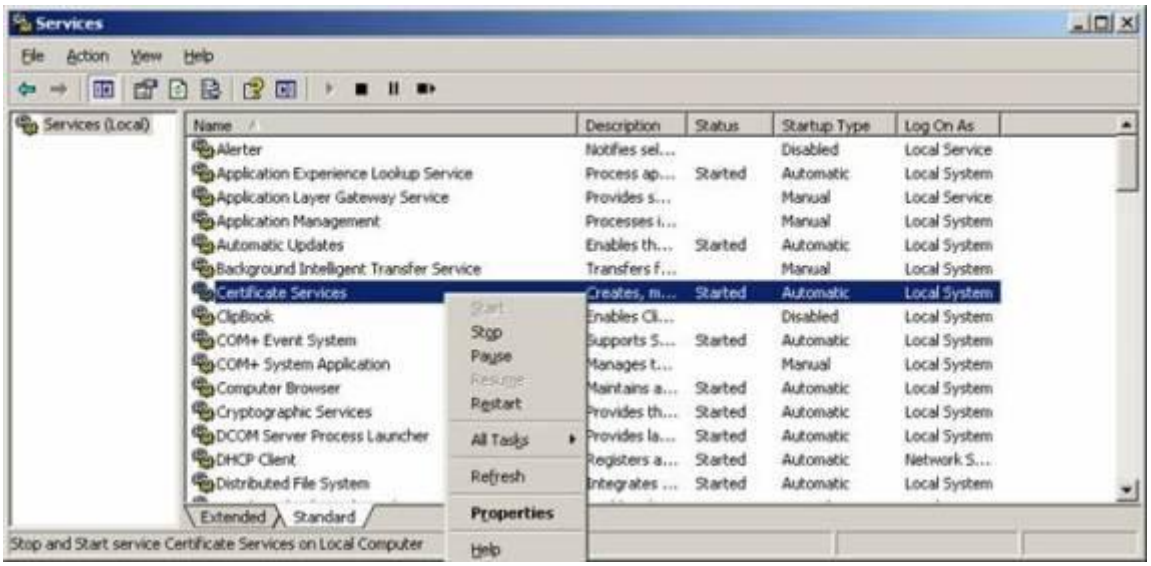

Figure 3.9: Restarting the Microsoft Certificate Service

We're now ready to submit the certificate request to the Microsoft CA. One way to do this is to open a browser and type http://dc_name/certsrv. On the Welcome page, click **Request a certificate** (**Figure 3.10**).

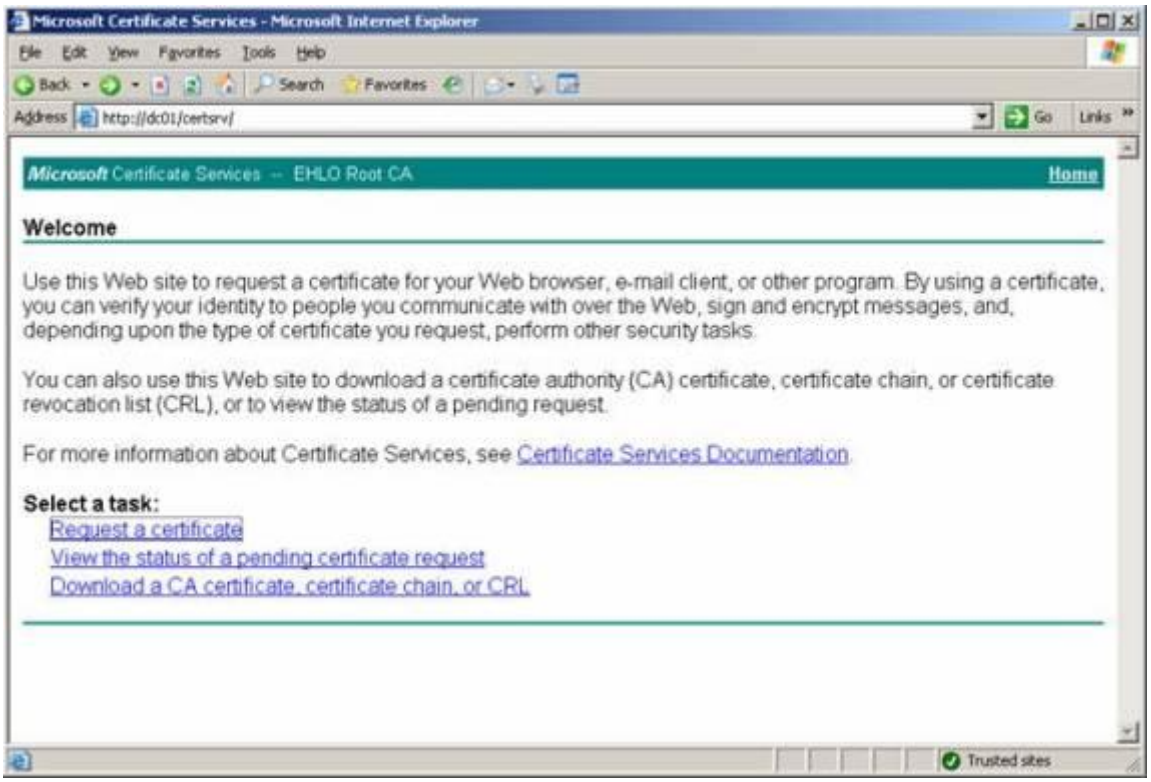

Figure 3.10: Microsoft Certificates Welcome page

On the Request a Certificate page, click **advanced certificate request** (**Figure 3.11**).

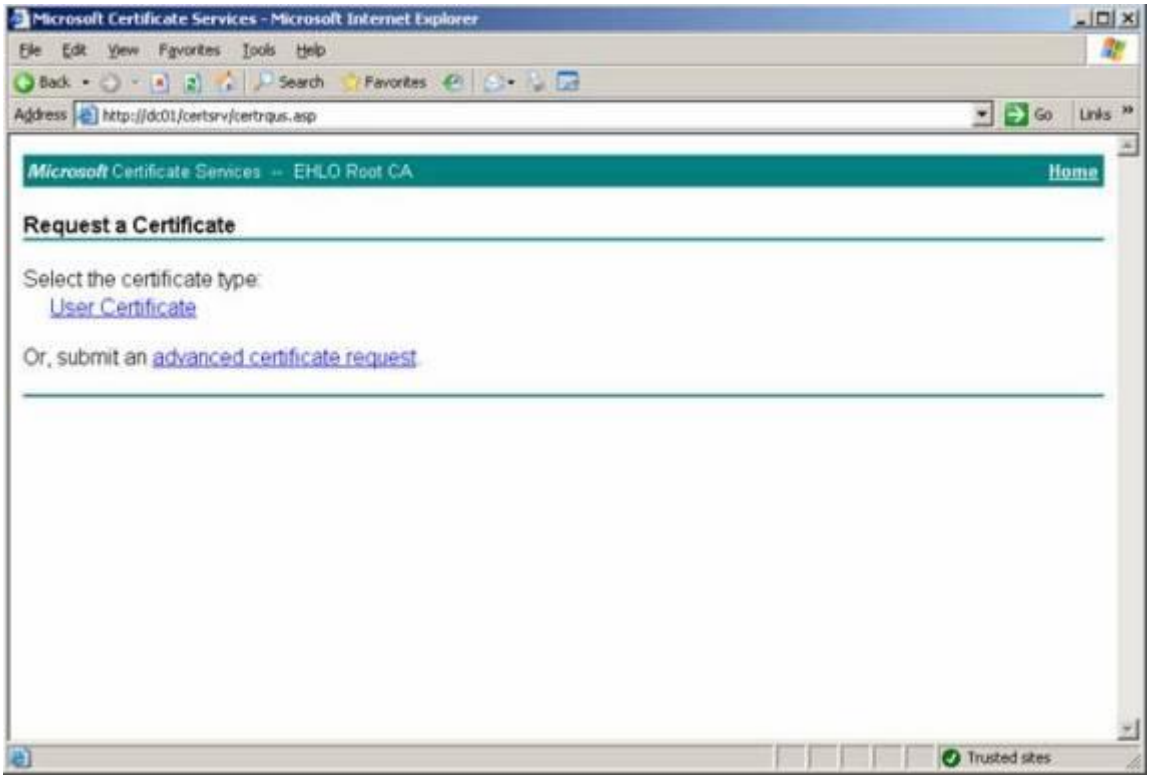

Figure 3.11: Requesting a Certificate

On the Advanced Certificate Request page, click **Submit a certificate request by using a base-64-encoded CMC or PKCS #10 file, or submit a renewal request by using a base-64-encoded PKCS #7 file** (**Figure 3.12**).

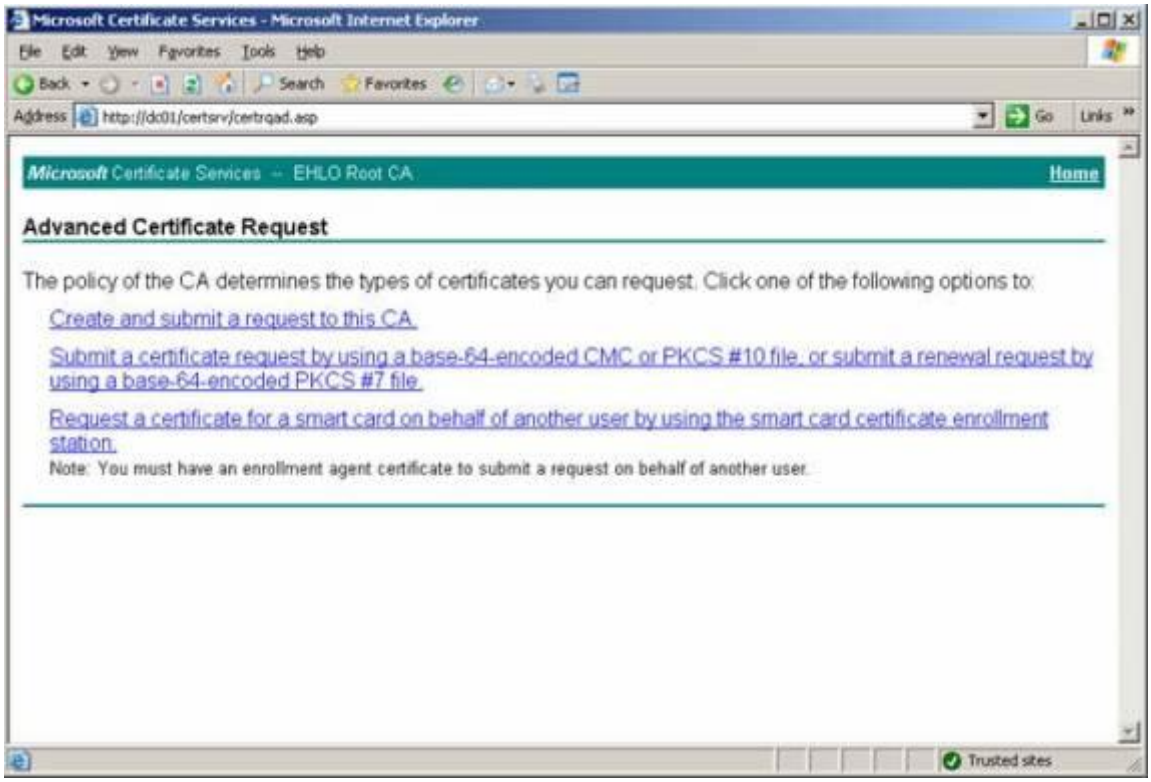

Figure 3.12: Selecting the second option on the Advanced Certificate Request page

Now paste the content of the certificate request file into the Base-64-encoded window as shown in **Figure 3.13**. Then select **Web Server** in the certificate template drop-down menu and click **Submit**.

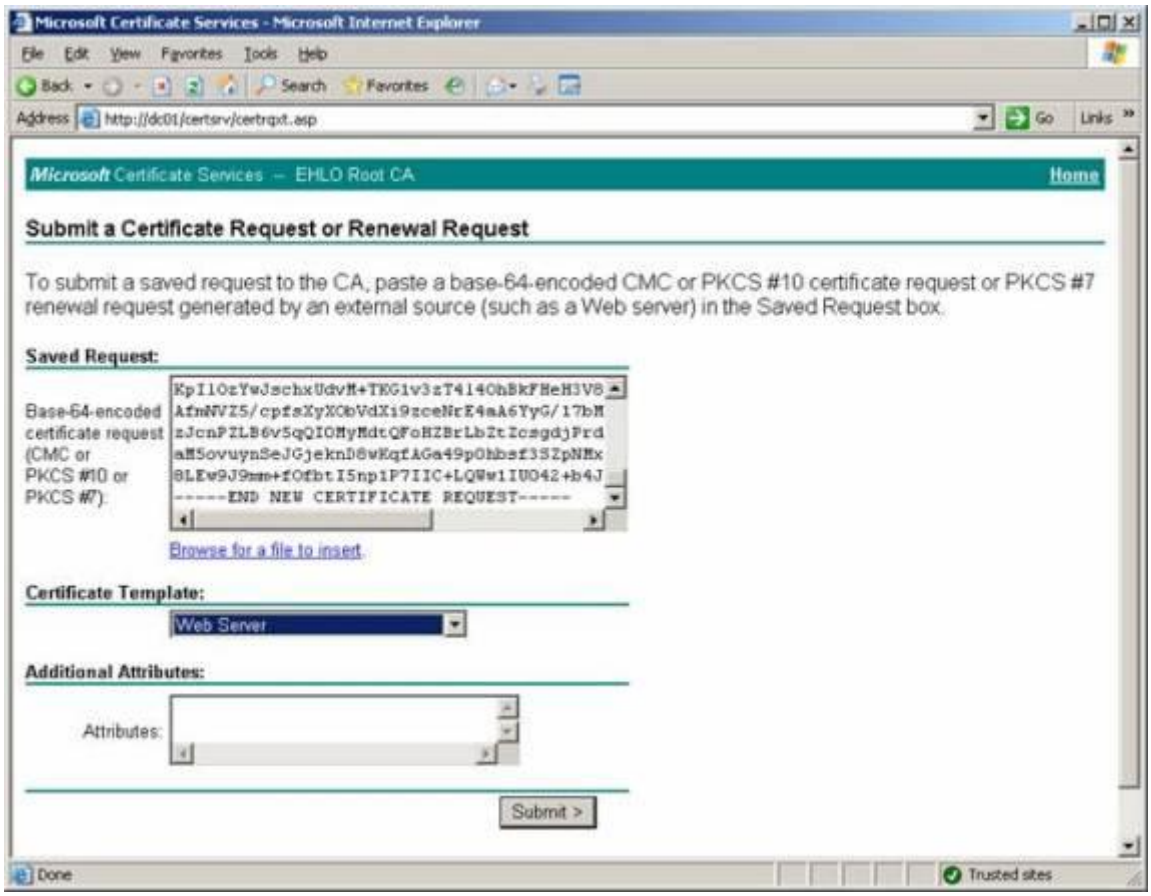

Figure 3.13: Submitting the Certificate Request

The certificate has been issued and you can download a DER or Base 64 encoded version by clicking Download certificate or Download certificate chain. Let's select **Base 64 encoded** followed by clicking **Download certificate chain** (**Figure 3.14**).

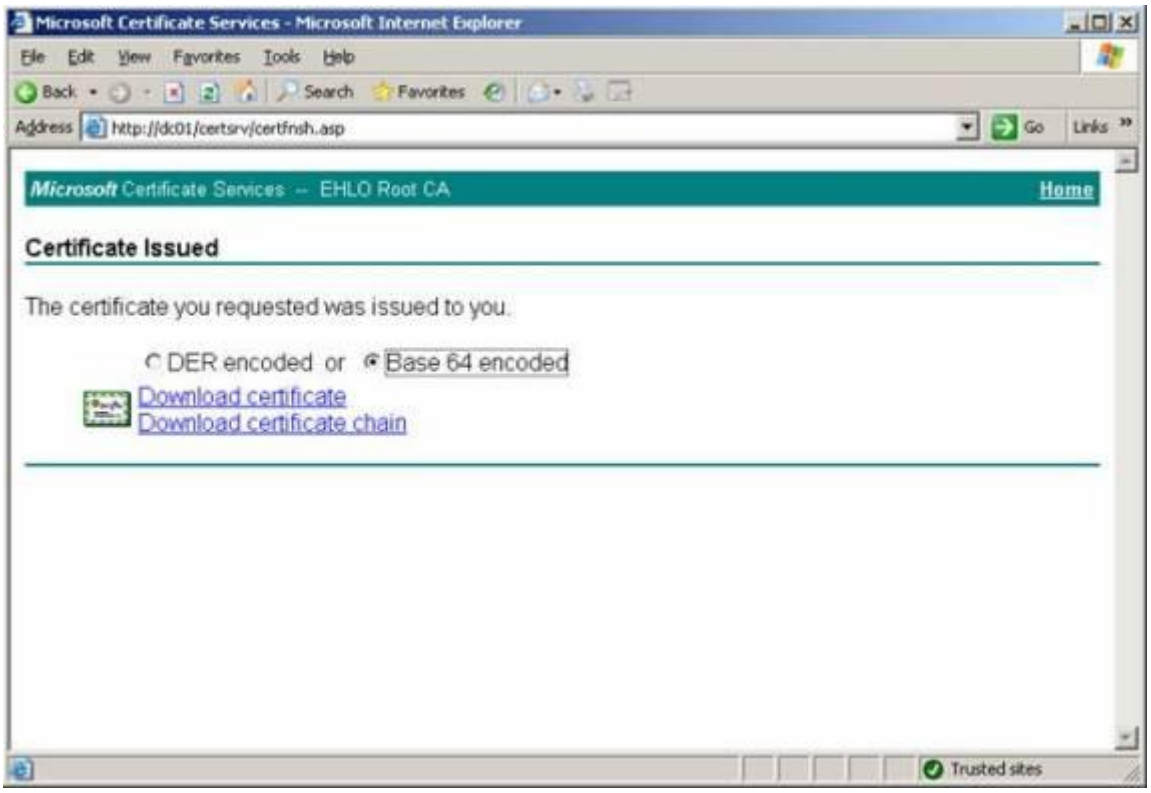

Figure 3.14: Downloading the issued Certificate

It's time to import the issued certificate using the Import-ExchangeCertificate cmdlet. We do this by typing the following command:

Import-ExchangeCertificate –Path c:\certnew.p7b

The certificate has now been imported to the personal certificate store.

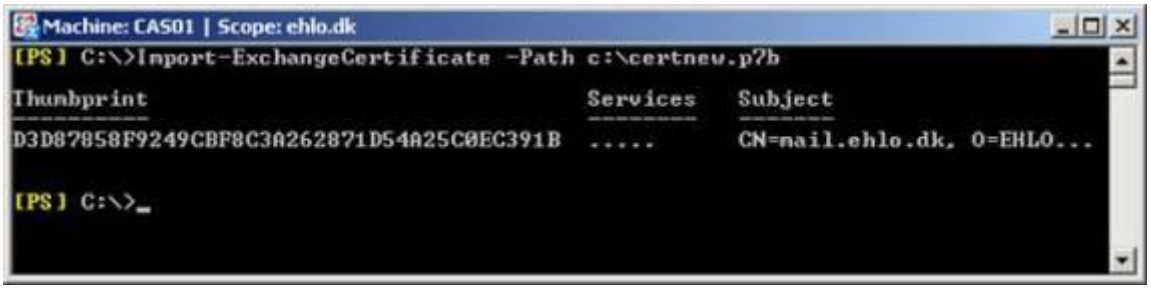

Figure 3.15

To verify the certificate looks like expected, let's now type the following command:

Get-ExchangeCertificate -Thumbprint <thumbprint> | FL

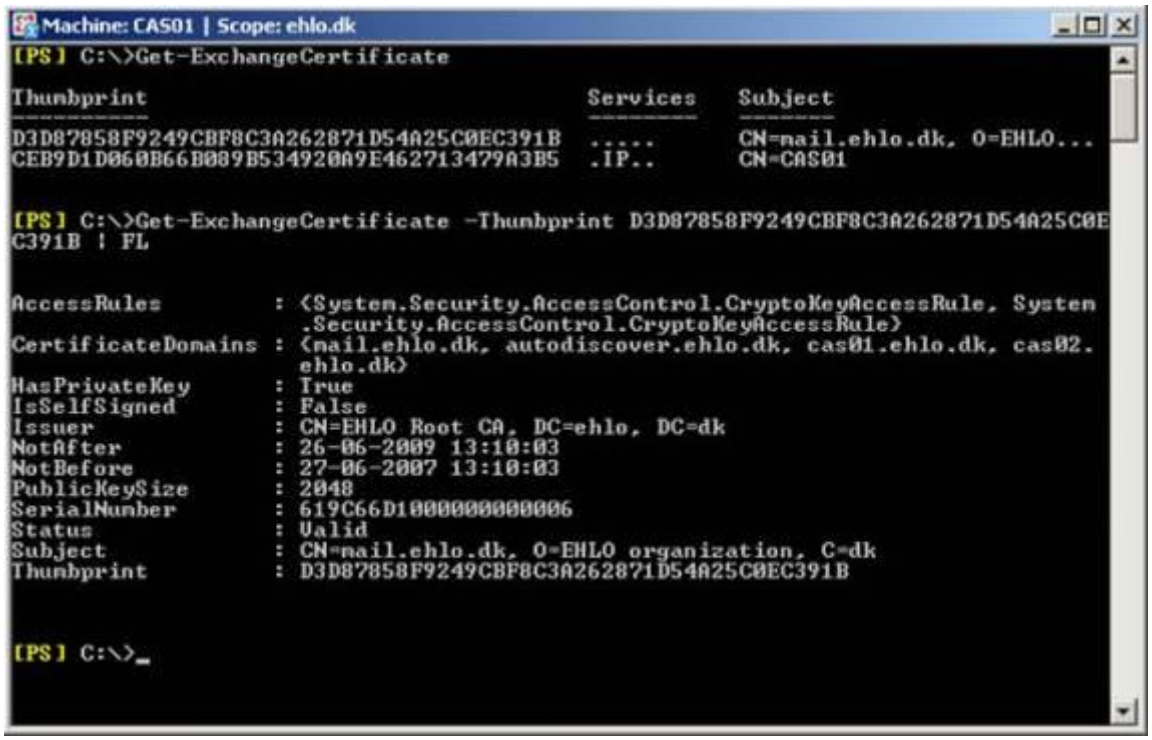

Figure 3.16: SAN Certificate - Detailed Information

Finally we need to enable the certificate for the client services, our end-users will use to connect to their mailboxes. In this setup I'll enable the certificate for OWA, EAS, Outlook Anywhere, POP3 and IMAP4. To do so we need to type:

Enable-ExchangeCertificate –Thumbprint <thumbprint> -Services "IIS, POP, IMAP"

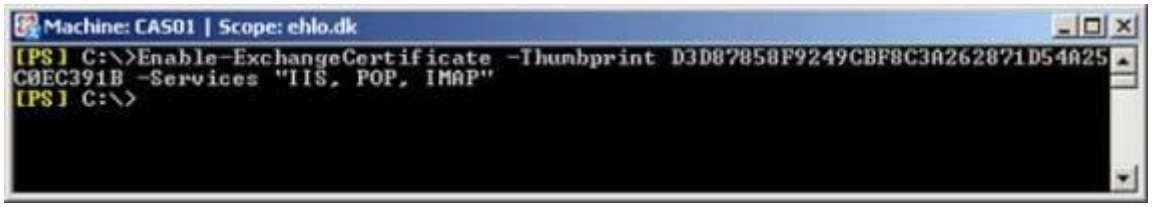

Figure 3.17: Enabling the SAN certificate

The certificate has now been enabled for these services but only on the first Client Access server in our NLB cluster.

Importing and Enabling the SAN SSL certificate on the Second Client Access Server in the NLB Cluster

To import the SAN certificate on the second Client Access server in the NLB cluster, we first need to export it from the first Client Access server. When doing so, we need to make sure we export the certificate with its private key. This is done by opening the Certificates snap-in. To open the Certicates snap-in, click **Start** > **Run** and type **mmc.exe** to first open an empty MMC window. Now click **File** > **Add/Remove Snap-in** > **Add** > Select **Certificates** > Click **Add** > Select **Computer Account** > Click **Next** > **Finish** > **Close** and finally **OK**. Expand **Certificates (Local Computer)** > **Personal,** then right-click on the certificate that should be exported. On the context appearing menu, select **All Tasks** > **Export** (**Figure 3.18**).

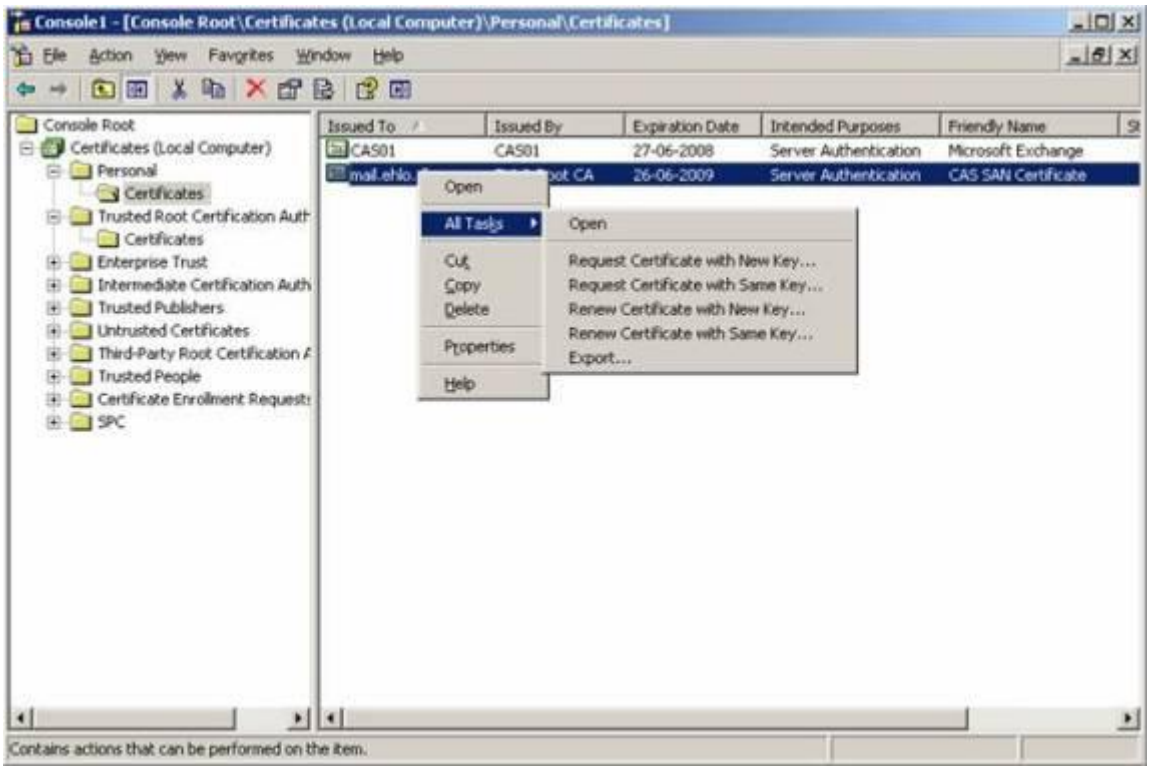

Figure 3.18: Selecting Export on the Context Menu

In the Certificate Export Wizard, click **Next**. On the Export Private Key page, select **Yes, export the private key** as shown in **Figure 3.19** then click **Next**.

Figure 3.19: Exporting the private key

On the **Export File Format** page, select **Personal Information Exchange – PKCS #12 (.PFX)** and tick **Include all certificates in the certificates path if possible** as shown in **Figure 3.20**. Click **Next**.

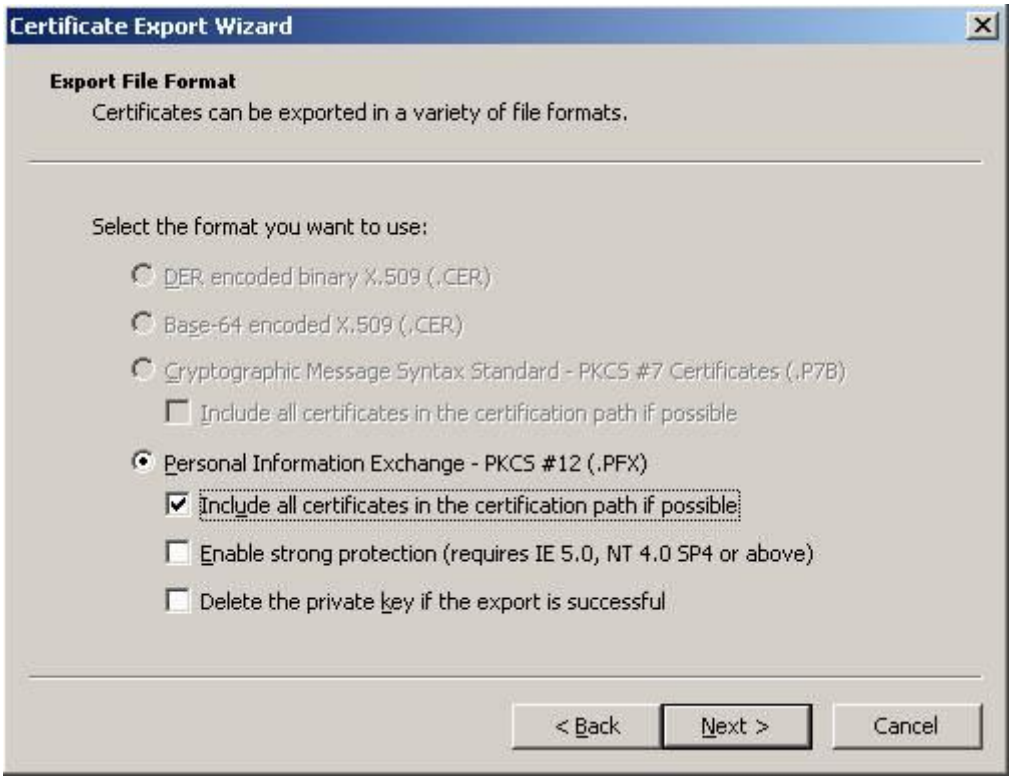

Figure 3.20: Selecting the format to use

Enter a password and click **Next** (**Figure 3.21**).

Note:
Make sure you remember this password as you need it when importing it on the second Client Access server.

Figure 3.21: Enter a password in order to protect the private key

Now specify the path to where you want to save the .PFX file (**Figure 3.22**), then click **Next**.

Figure 3.22: Specifying the path for the .PFX file

Finally click **Finish**.

Okay with the certificate exported, let's copy it to the C: drive of the second Client Access server, and then open the Exchange Management Shell on that server. To import the certificate, type the following command:

Import-ExchangeCertificate –Path c:\exported_cert.pfx –Password:(Get-Credential).password

When pressing Enter, you'll be prompted for the password you specified earlier on as shown **Figure 3.23**. It doesn't matter what username you specify as this isn't used in this type of authentication.

Figure 3.23: Importing the certificate

After clicking OK, the certificate has been imported (**Figure 3.24**).

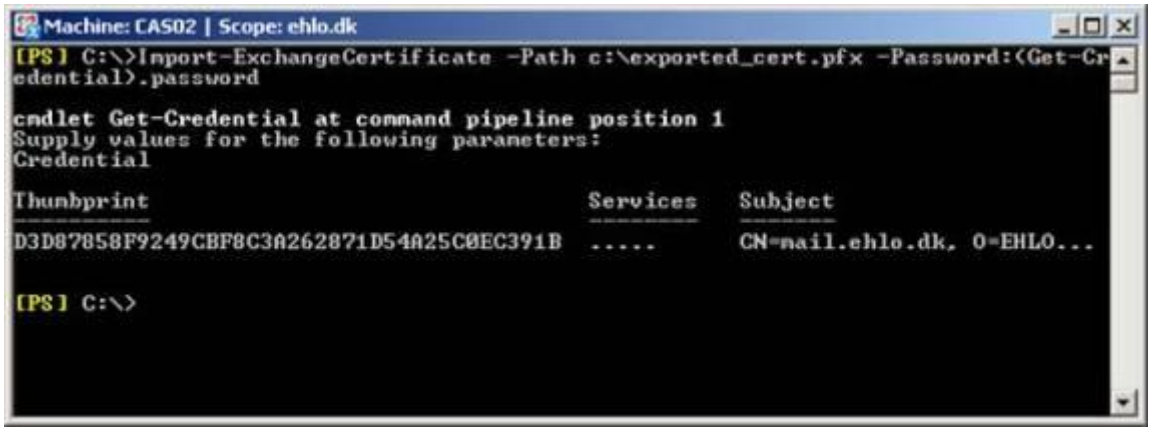

Figure 3.24: Certficate imported

Now copy the certificate thumbprint to the clipboard, then enable the certificate for the required services by typing the following command (just like we did on the first Client Access server):

Enable-ExchangeCertificate –Thumbprint <thumbprint> -Services "IIS, POP, IMAP"

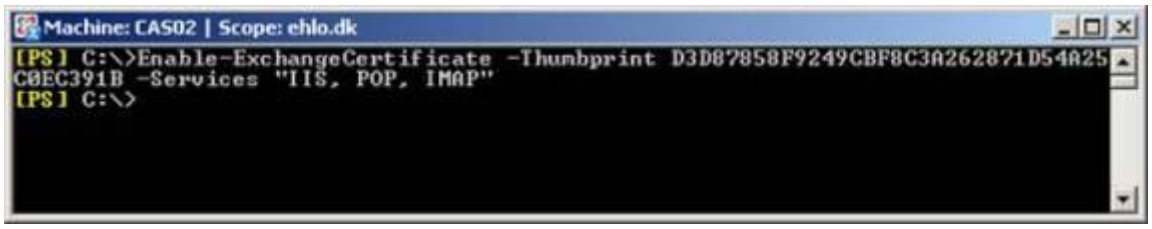

Figure 3.25: Enabling the SAN certificate on the second Client Access Server

The SAN certificate has now been properly enabled on both servers, and if the clients trust the root CA from our internal Microsoft CA, we should no longer get security warnings, when accessing OWA via the NLB cluster name as shown in **Figure 3.26**.

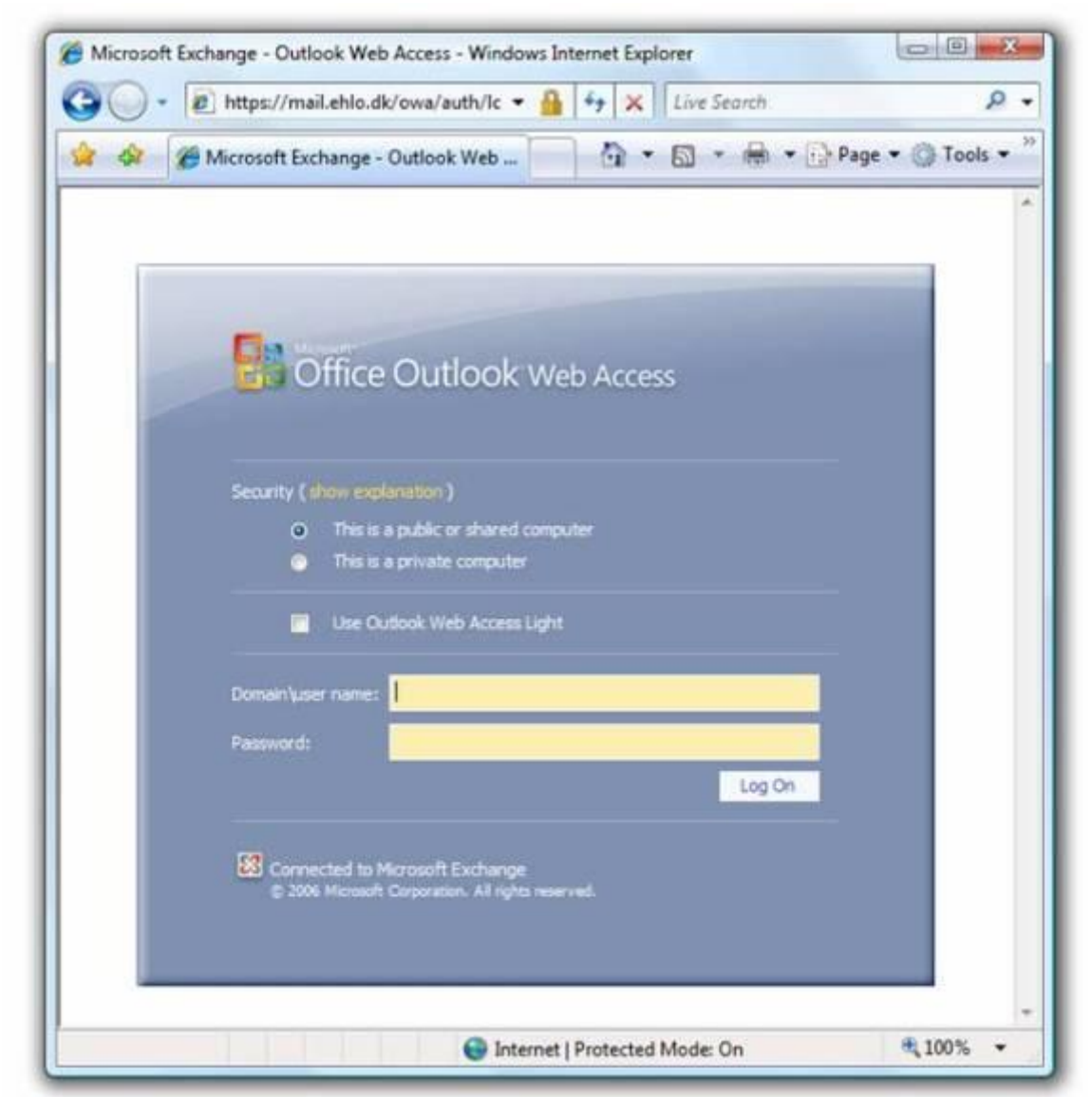

Figure 3.26: Accessing OWA 2007 without security warnings