

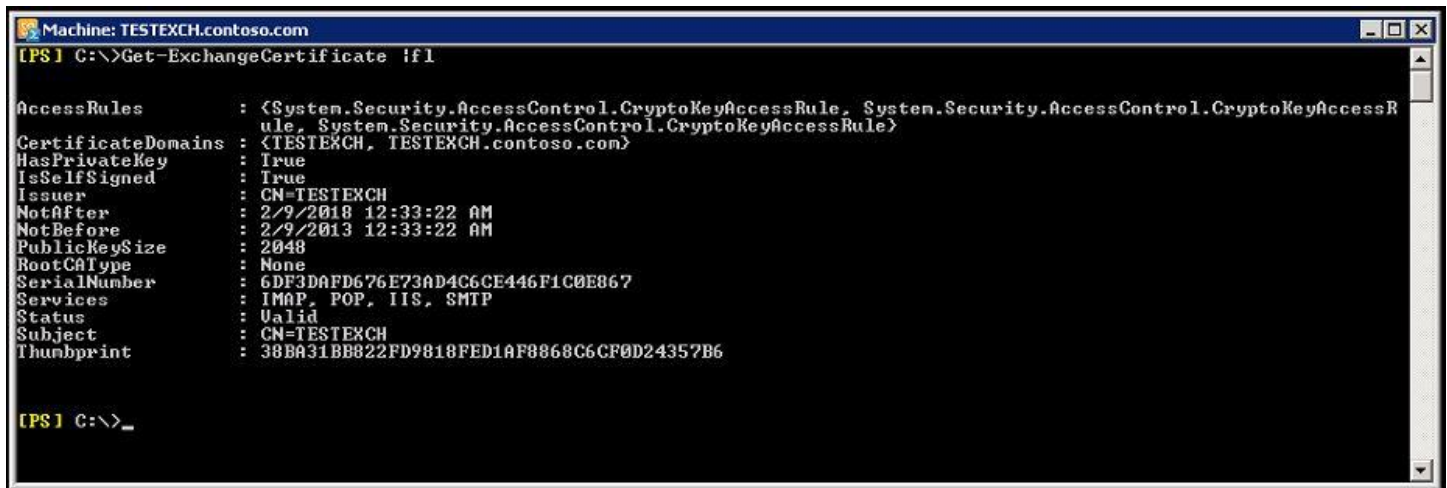
Publishing OWA/Outlook Anywhere with Self Signed Certificate

OM February 11, 2013 [Exchange Hosted 7 Comments](#)

My boss recently acquired a small company. Now he wants to permit the users, remote access via OWA/Outlook Anywhere. The user base is around 50, so he doesn't want to spend on SAN certificate. They doesn't have a PKI either, so only possible option left is to use a Self Signed Certificate for both OWA/Outlook Anywhere.

So, before deploying it to Production, I gave it a try on my Lab and it passed with flying colors.

When you install Exchange 2010, a self-signed certificate is automatically configured. A self-signed certificate is signed by the application that created it. The subject and the name of the certificate match. The issuer and the subject are defined on the certificate.

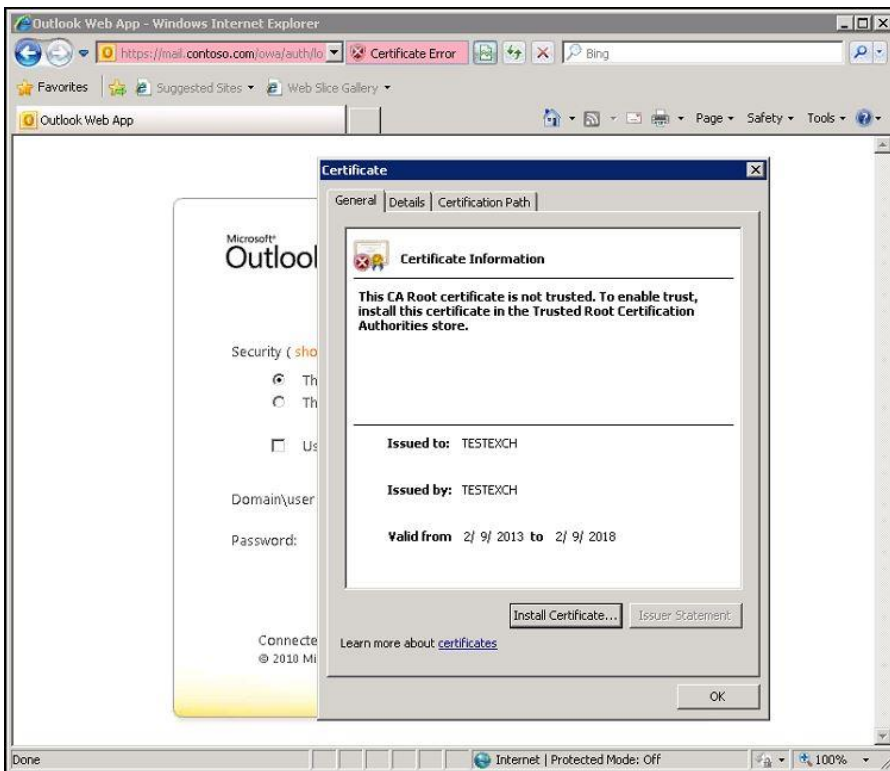


```
Machine: TESTEXCH.contoso.com
[PS] C:\>Get-ExchangeCertificate if1

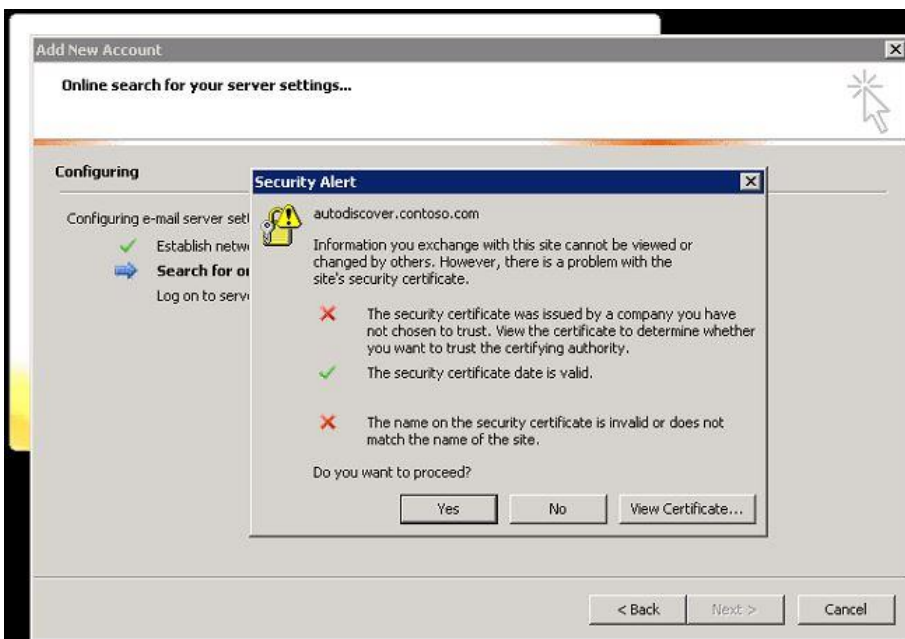
AccessRules           : <System.Security.AccessControl.CryptoKeyAccessRule, System.Security.AccessControl.CryptoKeyAccessRule>
CertificateDomains    : <TESTEXCH, TESTEXCH.contoso.com>
HasPrivateKey         : True
IsSelfSigned          : True
Issuer                : CN=TESTEXCH
NotAfter              : 2/9/2018 12:33:22 AM
NotBefore             : 2/9/2013 12:33:22 AM
PublicKeySize         : 2048
RootCAType            : None
SerialNumber          : 6DF3DAFD676E73AD4C6CE446F1C0E867
Services              : IMAP, POP, IIS, SMTP
Status                : Valid
Subject               : CN=TESTEXCH
Thumbprint             : 38BA31BB822FD9818FED1AF8868C6CF0D24357B6

[PS] C:\>_
```

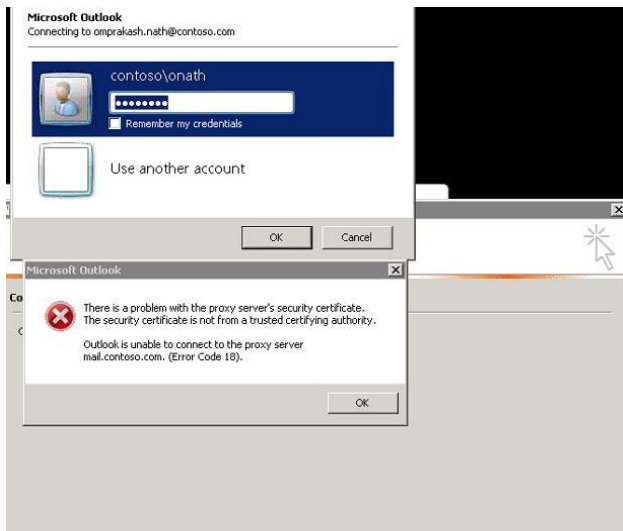
If you launch OWA at this point, you will be prompted with the Certificate error as the certificate was not a trusted root certificate.



I also tried to configure Outlook Anywhere and as expected, it also throws Security error.



Once you accept this warning, you will be prompted with credential, but as the certificate is not a trusted one, a secure channel won't be initiated and it will throw a proxy error.



So, you won't be able to configure Outlook Anywhere with the default Self Signed Certificate.

The solution is to generate a new Self Signed Certificate with the following names on it.

- mail.contoso.com
- autodiscover.contoso.com

We need to do this via Power Shell,

New-ExchangeCertificate -FriendlyName "SelfSigned Certificate" -KeySize 2048 -SubjectName "c=IN, s=, l=, o=CONTOSO, ou=IT, cn=CONTOSO.COM" -DomainName MAIL.CONTOSO.COM, AUTODISCOVER.CONTOSO.COM -PrivateKeyExportable \$True

```
Machine: TESTEXCH.contoso.com
[PS] C:\>New-ExchangeCertificate -FriendlyName "SelfSigned Certificate" -KeySize 2048 -SubjectName "c=IN, s=, l=, o=CONTOSO, ou=IT, cn=CONTOSO.COM" -DomainName MAIL.CONTOSO.COM, AUTODISCOVER.CONTOSO.COM -PrivateKeyExportable $True

Confirm
Overwrite the existing default SMTP certificate?
Current certificate: '38BA31BB822FD9818FED1AF9868C6CF0D24357B6' (expires 2/9/2018 12:33:22 AM)
Replace it with certificate: '7D363FBCA2742085671C2262E39038931AC11EFD' (expires 2/9/2018 12:50:35 AM)
[Y] Yes [N] No to All [?] Help (default is "Y"): y

Thumbprint          Services    Subject
-----
7D363FBCA2742085671C2262E39038931AC11EFD    ....S.    CN=CONTOSO.COM, OU=IT, O=CONTOSO, L="", S="", C=IN

[PS] C:\>Get-ExchangeCertificate 7D363FBCA2742085671C2262E39038931AC11EFD -f1

AccessRules          : <System.Security.AccessControl.CryptoKeyAccessRule, System.Security.AccessControl.CryptoKeyAccessRule>
CertificateDomains    : <CONTOSO.COM, MAIL.CONTOSO.COM, AUTODISCOVER.CONTOSO.COM>
HasPrivateKey         : True
IsSelfSigned         : True
Issuer               : CN=CONTOSO.COM, OU=IT, O=CONTOSO, L="", S="", C=IN
NotAfter              : 2/9/2018 12:50:35 AM
NotBefore             : 2/9/2013 12:50:35 AM
PublicKeySize        : 2048
RootCAType           : None
SerialNumber         : 12E368700F8E399C4F2F33F8580916AA
Services             : SMTP
Status               : Valid
Subject              : CN=CONTOSO.COM, OU=IT, O=CONTOSO, L="", S="", C=IN
Thumbprint           : 7D363FBCA2742085671C2262E39038931AC11EFD

[PS] C:\>
```

Once the certificate is generated, we have to enable the certificate for IIS, (SMTP) services.

At this point, we need to delete the original self signed certificate which was generated during the installation.

```
Machine: TESTEXCH.contoso.com
[PS] C:\New-ExchangeCertificate -FriendlyName "SelfSigned Certificate" -KeySize 2048 -SubjectName "c=IN, s=, l=, o=CONTOSO, ou=IT, cn=CONTOSO.COM" -DomainName MAIL.CONTOSO.COM, AUTODISCOVER.CONTOSO.COM -PrivateKeyExportable $true

Confirm
Overwrite the existing default SMTP certificate?

Current certificate: '38BA31BB822FD9818FED1AF8868C6CF0D24357B6' (expires 2/9/2018 12:33:22 AM)
Replace it with certificate: '7D363FBCA2742085671C2262E39038931AC11EFD' (expires 2/9/2018 12:50:35 AM)
[Y] Yes [A] Yes to All [N] No [L] No to All [?] Help (default is "Y"): y

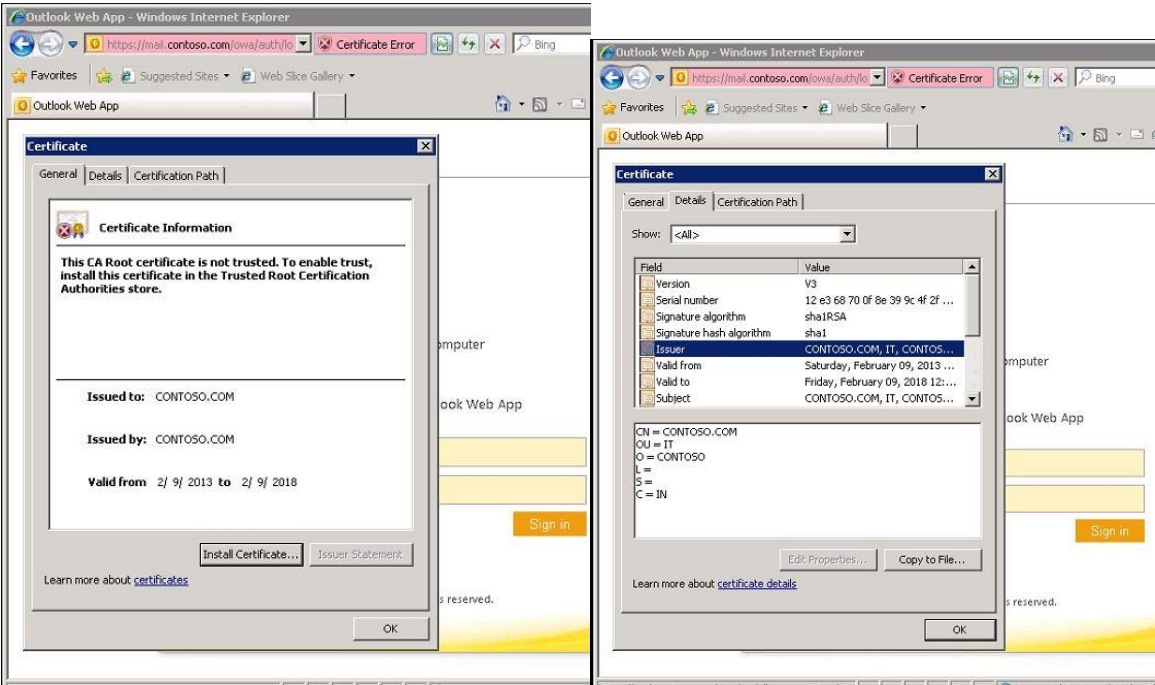
Thumbprint          Services          Subject
-----
7D363FBCA2742085671C2262E39038931AC11EFD ....S. CN=CONTOSO.COM, OU=IT, O=CONTOSO, L="", S="", C=IN

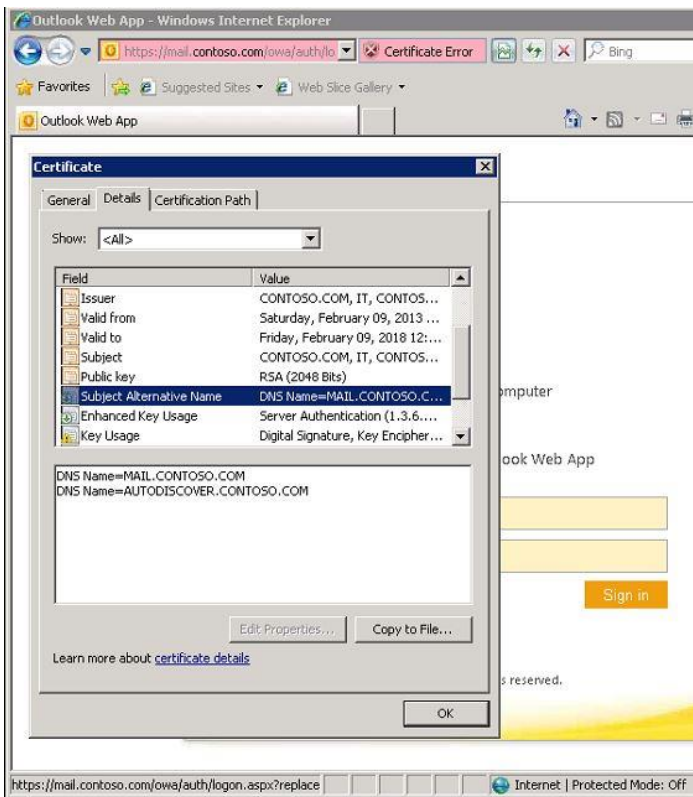
[PS] C:\>Get-ExchangeCertificate 7D363FBCA2742085671C2262E39038931AC11EFD -f1

AccessRules          : <System.Security.AccessControl.CryptoKeyAccessRule, System.Security.AccessControl.CryptoKeyAccessRule, System.Security.AccessControl.CryptoKeyAccessRule>
CertificateDomains    : <CONTOSO.COM, MAIL.CONTOSO.COM, AUTODISCOVER.CONTOSO.COM>
HasPrivateKey        : True
IsSelfSigned         : True
Issuer               : CN=CONTOSO.COM, OU=IT, O=CONTOSO, L="", S="", C=IN
NotAfter             : 2/9/2018 12:50:35 AM
NotBefore            : 2/9/2013 12:50:35 AM
PublicKeySize        : 2048
RootCAType           : None
SerialNumber         : 12E368700F8E399C4F2F33F8580916AA
Services             : SMTP
Status               : Valid
Subject              : CN=CONTOSO.COM, OU=IT, O=CONTOSO, L="", S="", C=IN
Thumbprint           : 7D363FBCA2742085671C2262E39038931AC11EFD

[PS] C:\>
```

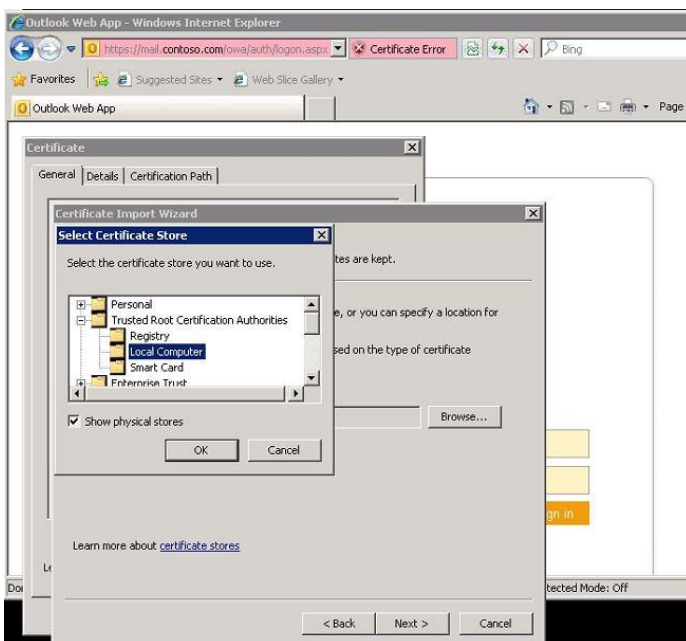
Now, if we launch OWA, we can see the new certificate with all the necessary names on this required for proper functioning of OWA/Outlook Anywhere.



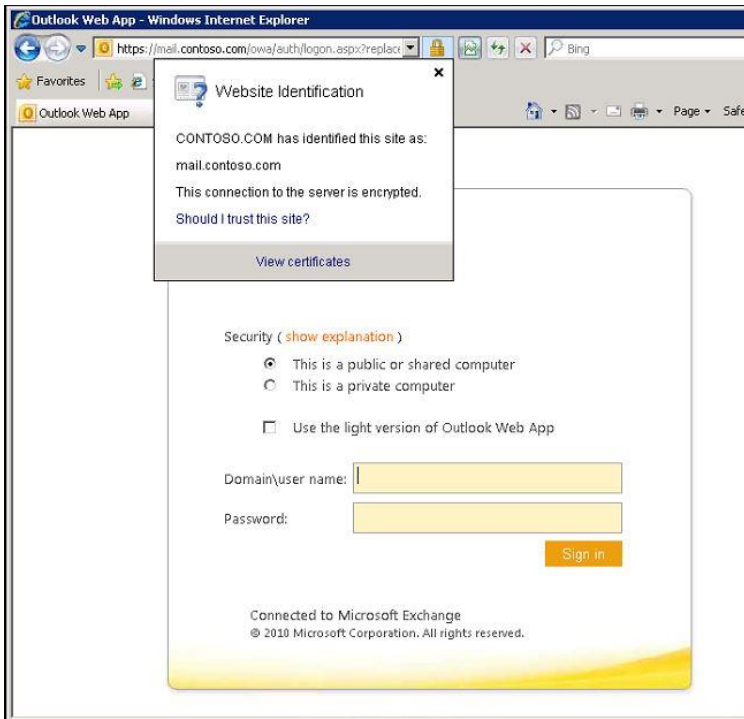


The Self-signed certificates must be manually copied to the trusted root certificate store on the client computer or mobile device. When a client connects to a server over SSL and the server presents a self-signed certificate, the client will be prompted to verify that the certificate was issued by a trusted authority. The client must explicitly trust the issuing authority. If the client confirms the trust, then, SSL communications can continue.

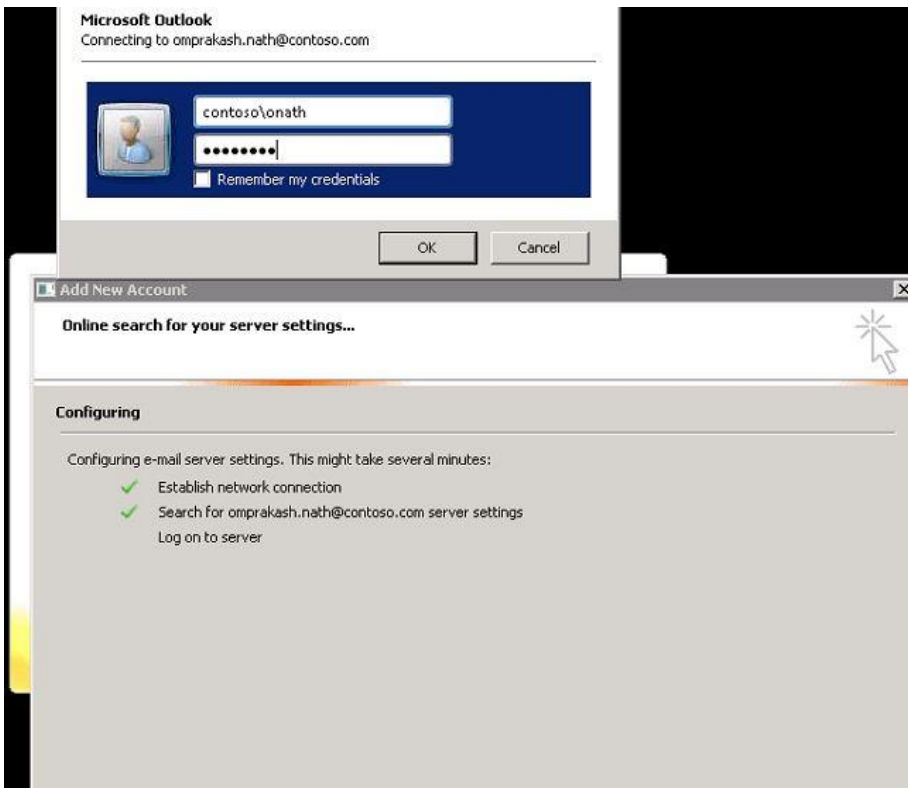
Now, we have to install this certificate in the local trusted certificate store on every machine which will be using OWA/Outlook Anywhere.

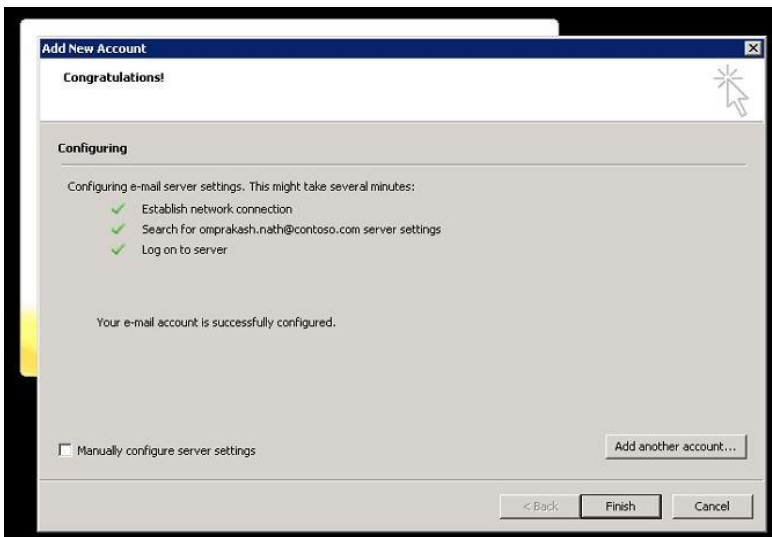


At this point if we launch OWA, no longer we will get the certificate error and our OWA site will now be a trusted.

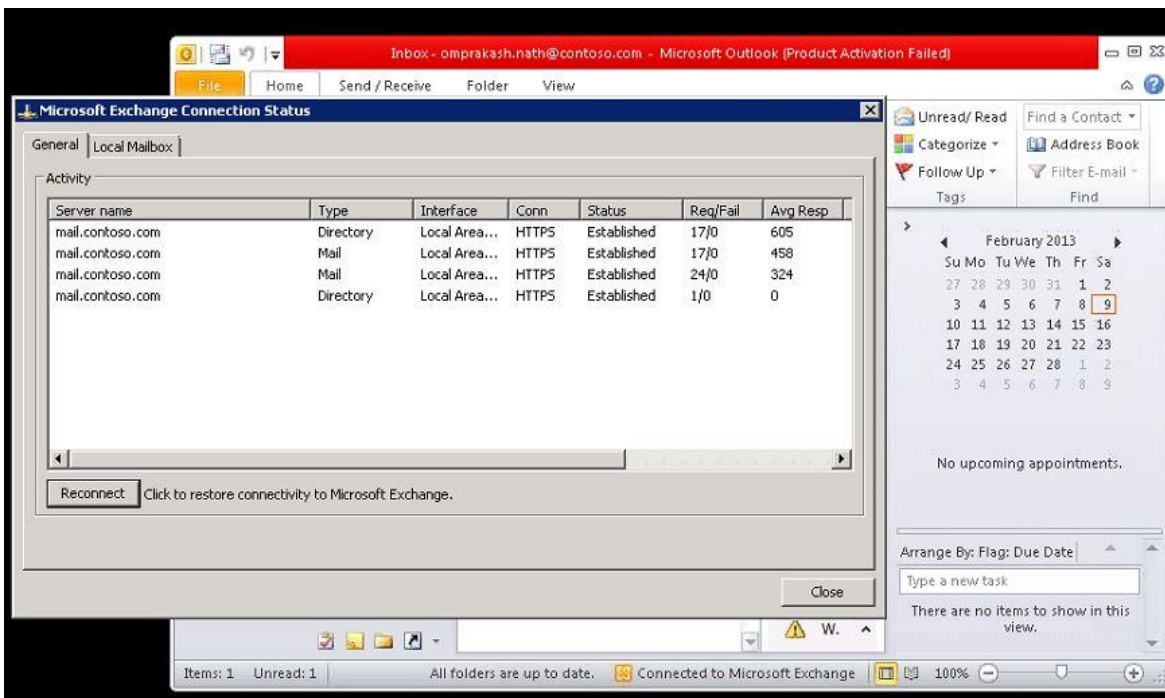


Now, if we start configuring Outlook Anywhere, it will be completed without any security/proxy error.

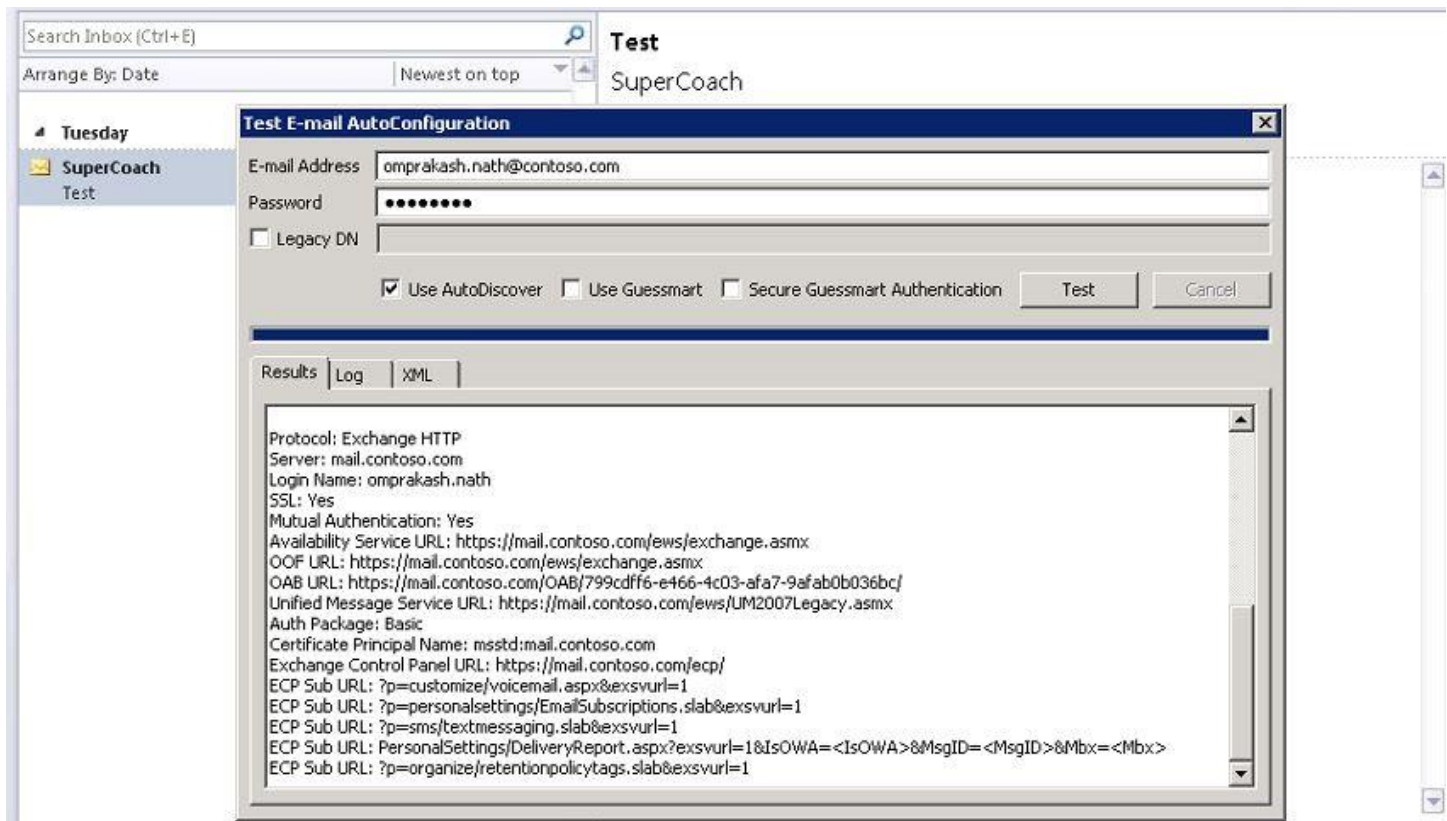




In my test, Outlook Anywhere was fully functional without any issues and was able to contact with CAS without any issues.



The OOF/OAB/Free Busy was fully functional along with Autodiscover Service.



Frequently, small organizations decide not to use a third-party certificate or not to install their own PKI to issue their own certificates. They might make this decision because those solutions are too expensive, because their administrators lack the experience and knowledge to create their own certificate hierarchy, or for both reasons. The cost is minimal and the setup is simple when you use self-signed certificates. However, it's much more difficult to establish an infrastructure for certificate life-cycle management, renewal, trust management, and revocation when you use self-signed certificates.