

App Center Certificate Request Process

Using an Internal Microsoft CA

Abstract:

The intent of this document is to describe the high-level process for both requesting and adding an SSL certificate to App Center when using an internal Microsoft CA. The CA server is a Windows 2008 R2 Enterprise server running AD Certificate Services.

Step 1 - Generating the private key and certificate request

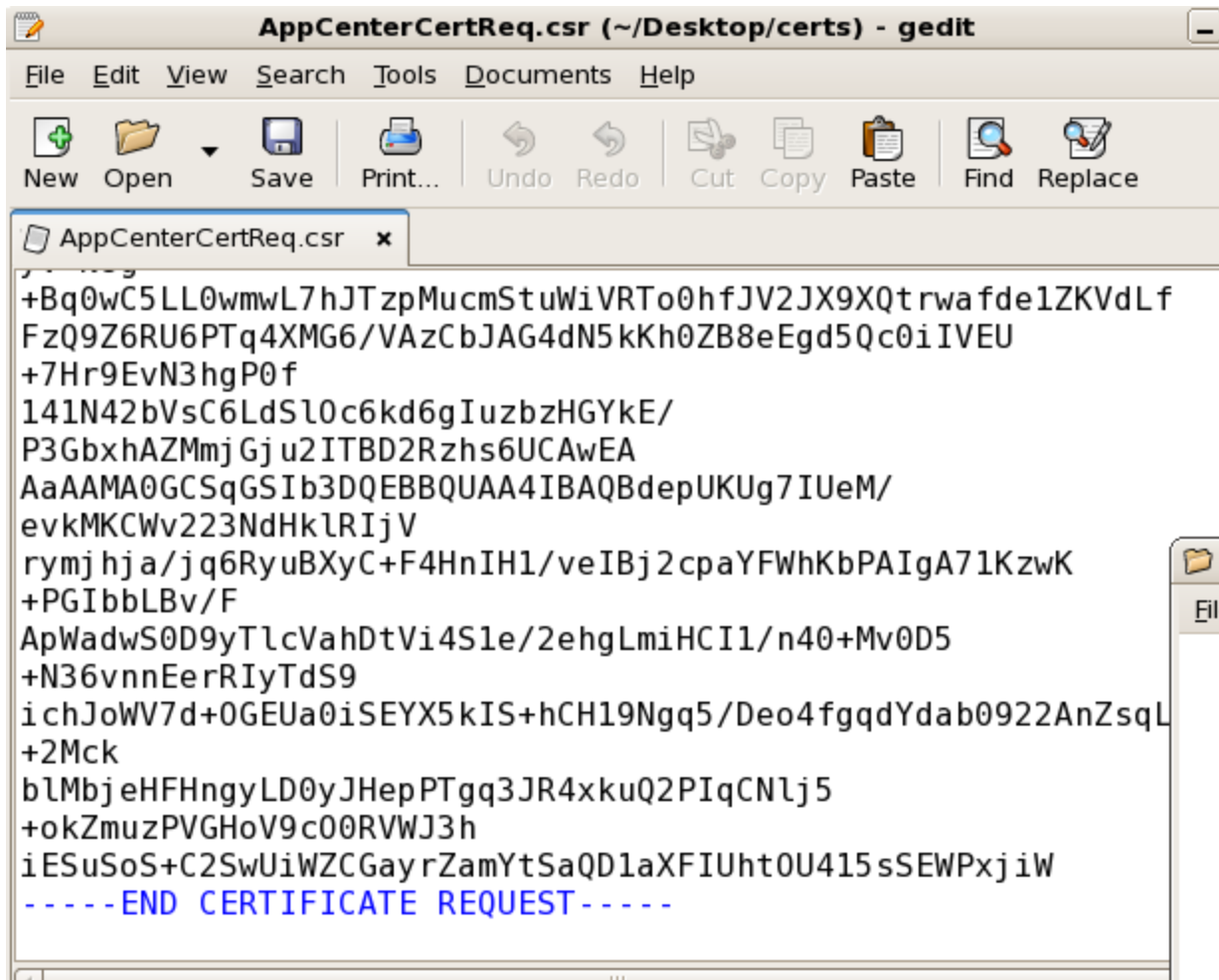
On your App Center server (RedHat or CentOS) run the following commands from a terminal. Please either be logged in as root or prefix each command with sudo.

`openssl req -out CSR.csr -new -newkey rsa:2048 -nodes -keyout privateKey.key`

During this process you will be prompted for some information. The most important item here is to make sure the “Common Name”/CN that you input will be the Fully Qualified Domain Name (FQDN) of the App Center server. If this does not make the FQDN of your App Center server you will have to repeat this process.

You should now have two files, “CSR.csr” and “privateKey.key” in the same directory that you ran the previous command. You will now need to take the contents of the CSR.csr file and provide them either to your CA admin, or if you have access to generate the certificate yourself take the following steps (or similar).

Example Screenshot of the CSR.csr file opened in gedit (notepad equivalent in linux).

A screenshot of a gedit text editor window. The title bar reads "AppCenterCertReq.csr (~/Desktop/certs) - gedit". The menu bar includes File, Edit, View, Search, Tools, Documents, and Help. The toolbar contains icons for New, Open, Save, Print..., Undo, Redo, Cut, Copy, Paste, Find, and Replace. A single tab is open, labeled "AppCenterCertReq.csr". The text area contains a Base64-encoded CSR request, starting with "+Bq0wC5LL0wmwL7hJTzpMucmStuWiVRT00hfJV2JX9XQtrwafde1ZKVdLf" and ending with "-----END CERTIFICATE REQUEST-----".

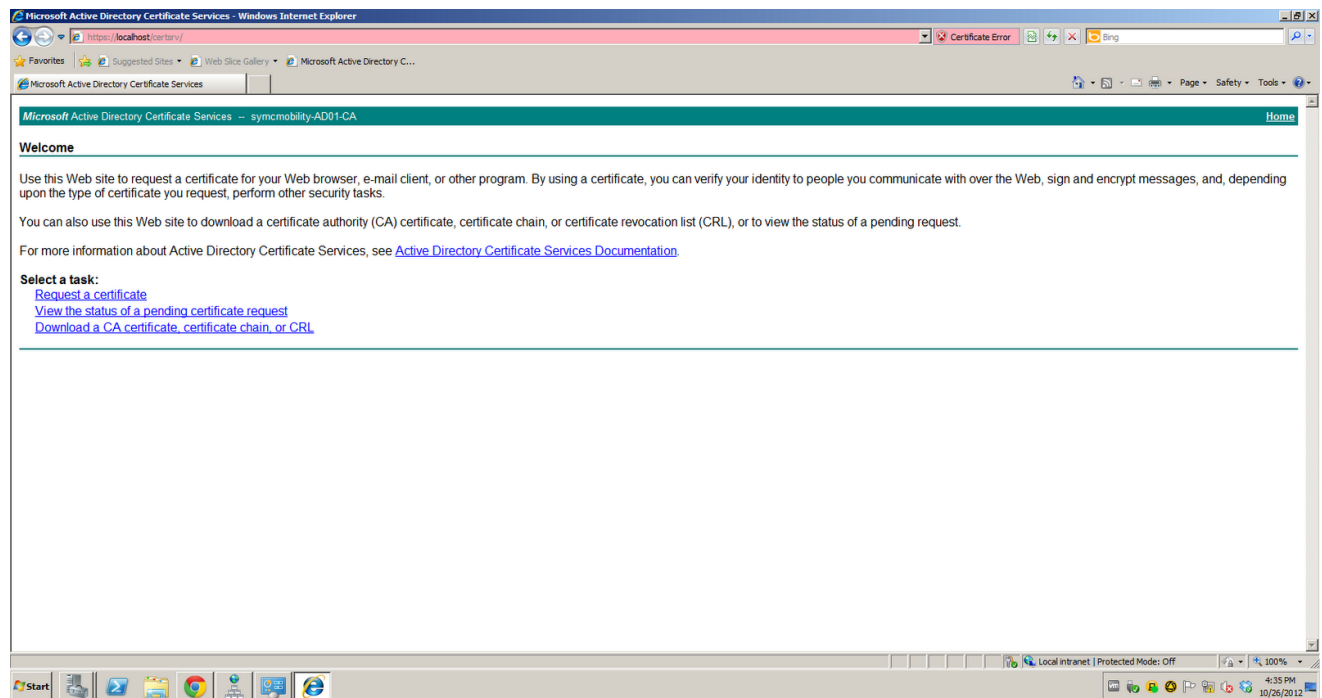
```
+Bq0wC5LL0wmwL7hJTzpMucmStuWiVRT00hfJV2JX9XQtrwafde1ZKVdLf
FzQ9Z6RU6PTq4XMG6/VAzCbJAG4dN5kKh0ZB8eEgd5Qc0iIVEU
+7Hr9EvN3hgP0f
141N42bVsC6LdS10c6kd6gIuzbzHGYkE/
P3GbxhAZMmjGju2ITBD2Rzhs6UCAwEA
AaAAMA0GCSqGSIsB3DQEBBQUAA4IBAQBdepUKUg7IUeM/
evkMKCWv223NdHklRIjV
rymj hja/jq6RyuBXyC+F4HnIH1/veIBj2cpaYFWhKbPAIgA71KzwK
+PGIbbLBv/F
ApWadwS0D9yTlcvahDtVi4S1e/2ehgLmiHCI1/n40+Mv0D5
+N36vnnEerRIyTdS9
ichJoWV7d+0GEUa0iSEYX5kIS+hCH19Ngq5/Deo4fgqdYdab0922AnZsqL
+2Mck
blMbj eHFHngyLD0yJHepPTgq3JR4xkuQ2PIqCNlj5
+okZmuzPVGHoV9c00RVWJ3h
iESuSoS+C2SwUiWZCGayrZamYtSaQD1aXFIUht0U415sSEWPxjiW
-----END CERTIFICATE REQUEST-----
```

Step 2 - Requesting the SSL certificate

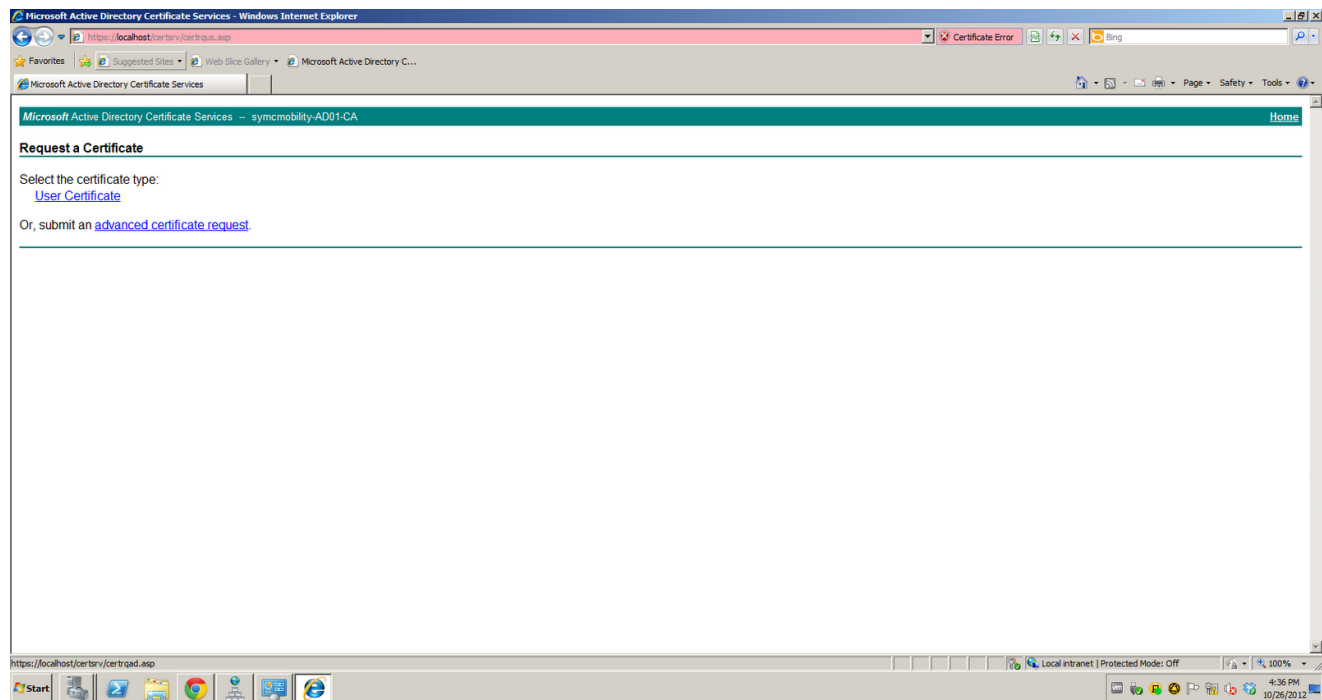
Go to the following URL on your Server 2008 R2 system.

<https://nameofyourCAserver/certsrv>

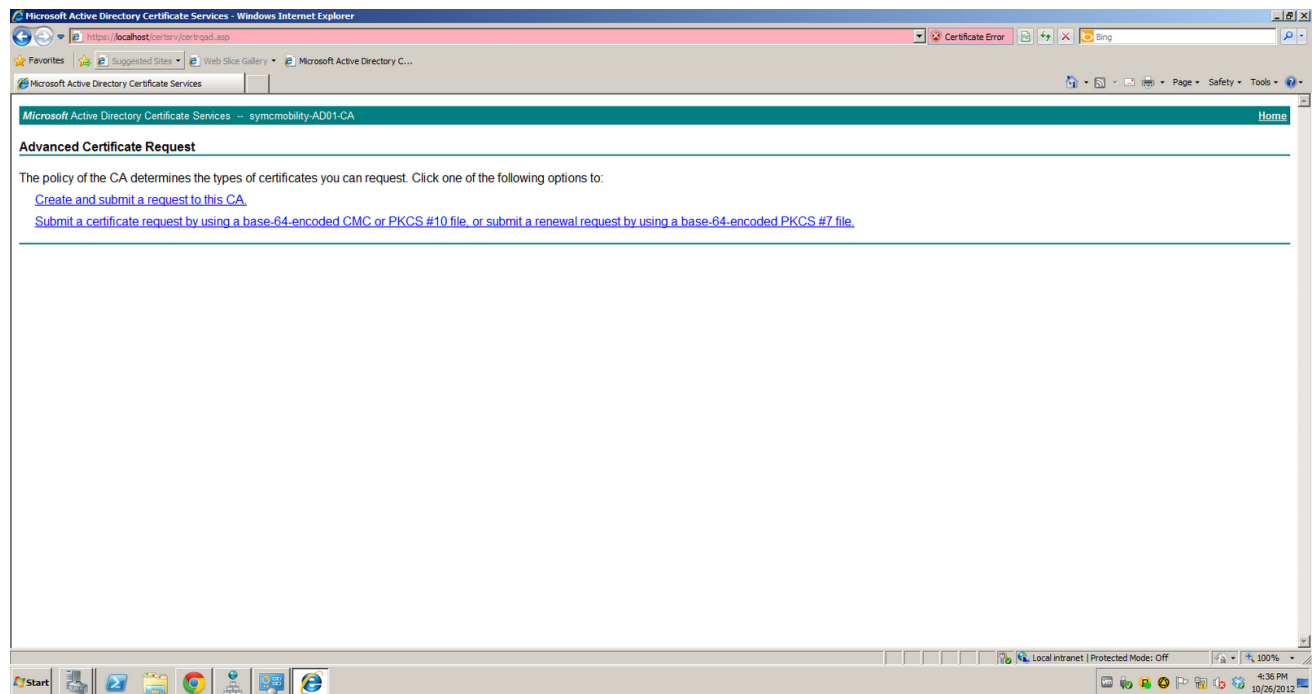
Select "request a certificate"



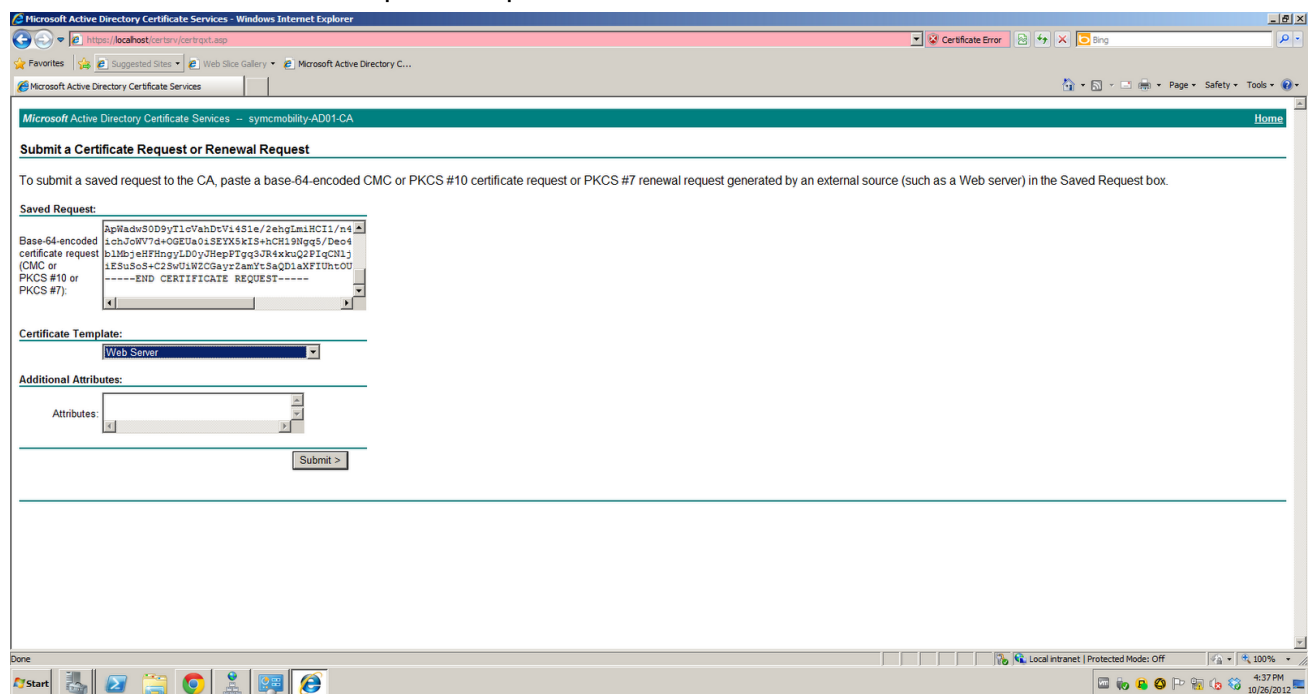
Select “advanced certificate request”



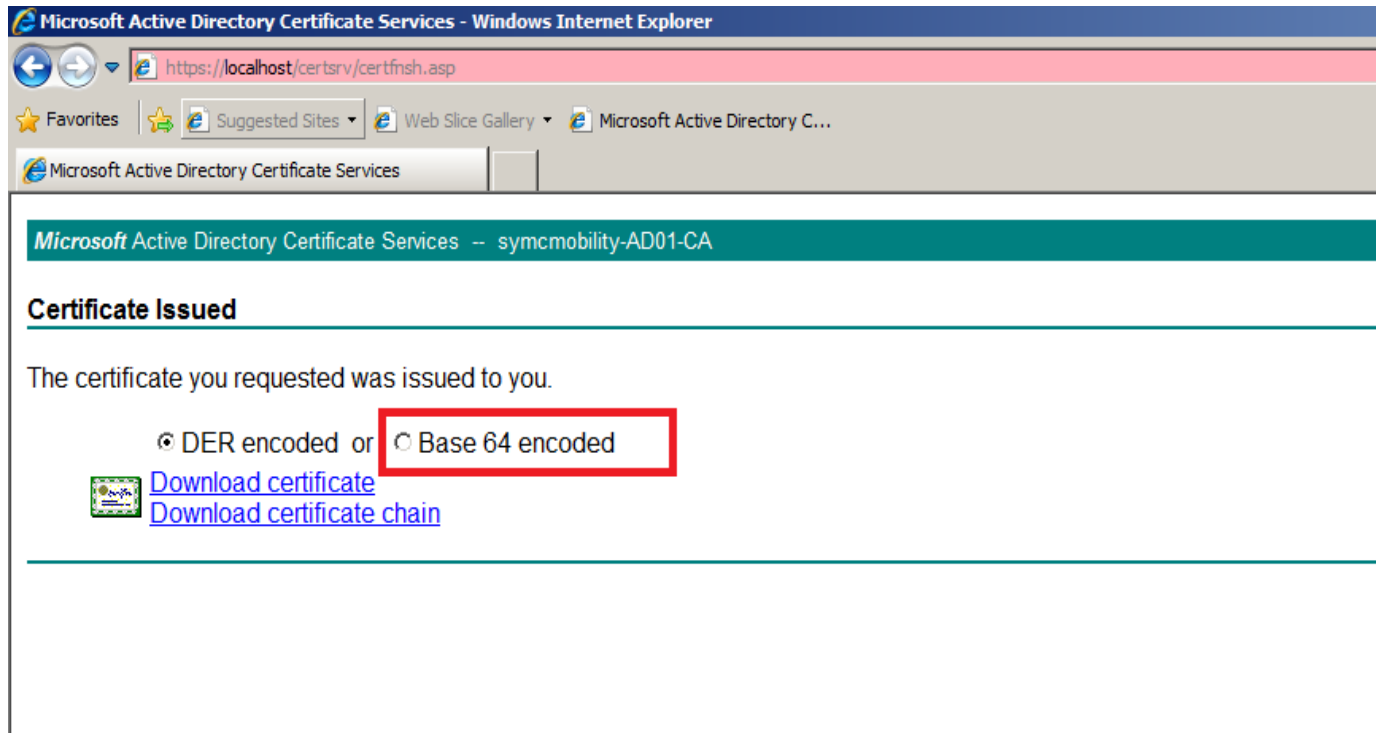
Select “Submit a certificate request by using a base-64-encoded CMC...”



Paste in the contents from the CSR.csr file into the “Base-64-encoded certificate request” box and from the “Certificate Template” drop-down select “Web Server” and click submit.



You should now have an option to download the issued certificate either “DER encoded” or “Base 64 encoded”. SELECT BASE 64 ENCODED for both the “Download Certificate” and “Download Certificate Chain”



As a second to final step you will need to convert the “Certificate Chain” file you just downloaded from P7B to PEM format. First make sure you copy both those the aforementioned files to your App Center server. Now, you can do this by running the following command, where certificatechain.p7b is the “certificate chain” you downloaded from the CA server and certificatechain.cer will be the name of the converted certificate.p7b file.

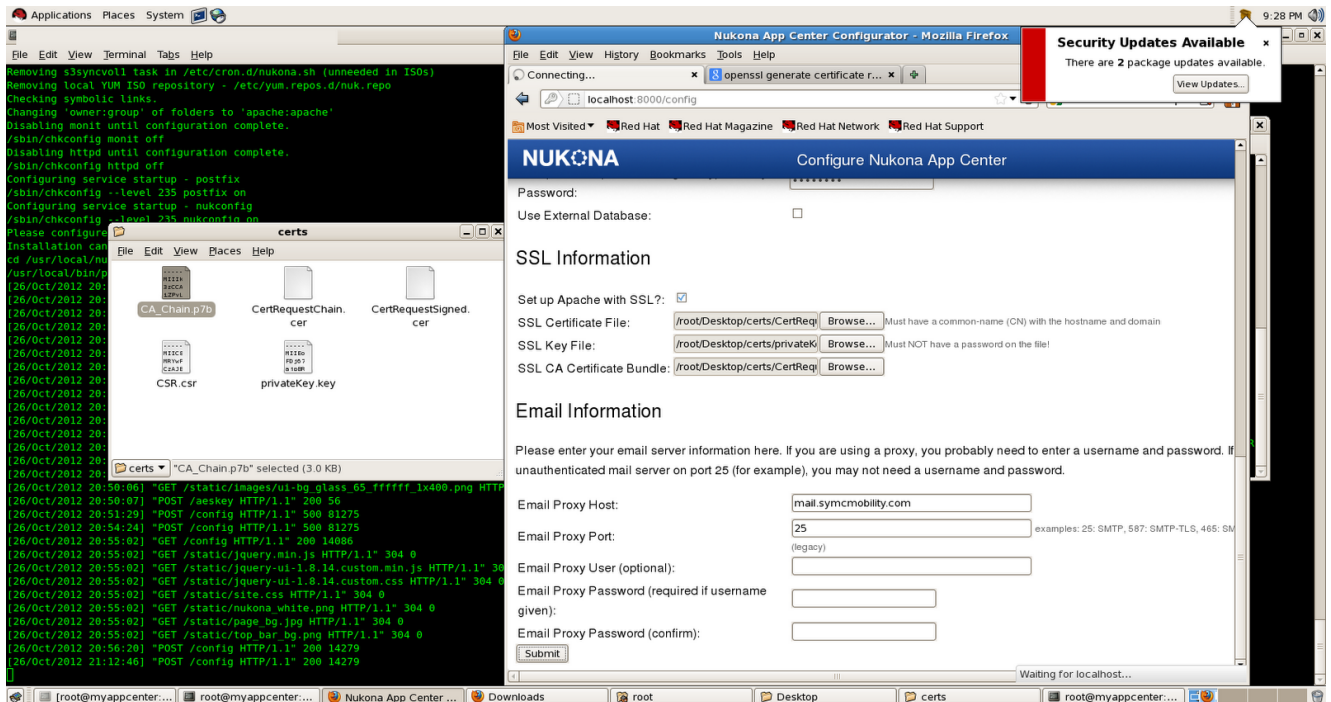
```
openssl pkcs7 -print_certs -in certificatechain.p7b -out certificatechain.cer
```

You should now have the following files

- privateKey.key
- certnew.cer (this is the default name of the SSL certificate that is issued from the Microsoft CA server).
- certificatechain.cer (the certificate chain file we just converted in the last step)

Step 3 - Selecting the certs to upload

These files will be uploaded to the App Center during the configuration process. Per the screenshot below, the signed certificate (certnew.cer) should be pointed to by the “SSL Certificate File”. the privateKey.key (from step 1) should be pointed to by the “SSL Key File”, and the certificatechain.cer (that we converted previously from p7b), should be pointed to by “SSL CA Certificate Bundle”.



Caveats:

Because I was using my own internal CA, none of my iOS or really any other devices outside my AD infrastructure, will trust this SSL certificate. Therefore, you must download the CA's root certificate and import it into each devices trusted certificate store. This can be done a number of ways, on Windows systems simply use the Certificates MMC snap in, for iOS devices you can use the iPCU. In either case, the same site I browsed to to request my certificate from (<https://nameofyourcaserver/certsrv>) will have a download link/area similar to the screenshot below. Use that to import into your devices OS. Again, Base 64 is the encoding option that should be used.

