Windows Server 2012 R2 Two-Tier PKI CA Pt. 2

derekseaman.com/2014/01/windows-server-2012-r2-two-tier-pki-ca-pt-2.html

Derek Seaman January 6, 2014

Now that our root Windows Server 2012 R2 certificate authority is installed and published to Active Directory from Part 1, it is time to bring online our subordinate CA. The subordinate CA will be our online issuing CA, since it will be the CA which issues all certificates, be they for users, computers, ESXi hosts, etc. The VM will be joined to the domain, and be online 100% of the time.

As with the offline root, you should perform hardening of this VM as well. Enabling the Windows firewall (or a third party one), anti-virus software, Microsoft EMET, and following Microsoft security baseline settings are all strongly recommended. If you have security software that can monitor file changes or system integrity, that too would be a great idea. Auditing tools such as Splunk, for real time alerting, would be ideal for defense in depth.

Install Windows Server 2012 R2 Subordinate CA

1. Use Notepad and create a file called **CAPolicy.inf** in **C:\Windows** on your subordinate VM. Use the code snippet below, but change the URL to match that previously used in configuring your offline root.

view sourceprint?

```
1  [Version]
2  Signature="$Windows NT$"
3  [PolicyStatementExtension]
4  Policies=InternalPolicy
5  [InternalPolicy]
6  OID= 1.2.3.4.1455.67.89.5
7  Notice="Legal Policy Statement"
8  URL=http://www.contoso.local/pki/cps.txt
```

```
[Certsrv_Server]
RenewalKeyLength=2048
RenewalValidityPeriod=Years
RenewalValidityPeriodUnits=5
LoadDefaultTemplates=0
AlternateSignatureAlgorithm=1
```

4. Run the following PowerShell command. Change the CACommonName as needed. The command will completely instantly.

view sourceprint?

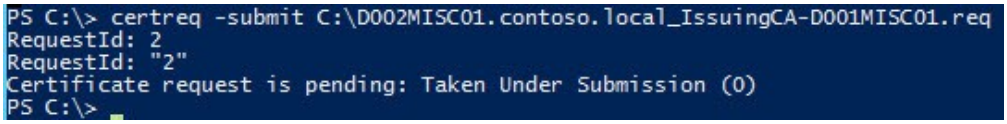
```
Add-WindowsFeature Adcs-Cert-Authority -IncludeManagementTools
Add-WindowsFeature Adcs-web-enrollment
install-adcswebenrollment
Install-AdcsCertificationAuthority -CAType EnterpriseSubordinateCA -
CACommonName "IssuingCA-D002MISC01" -KeyLength 2048 -HashAlgorithm SHA256 -
CryptoProviderName "RSA#Microsoft Software Key Storage Provider"
```

5. Copy the resulting request (see the yellow information text from the last command for the path and file name) to the offline CA.

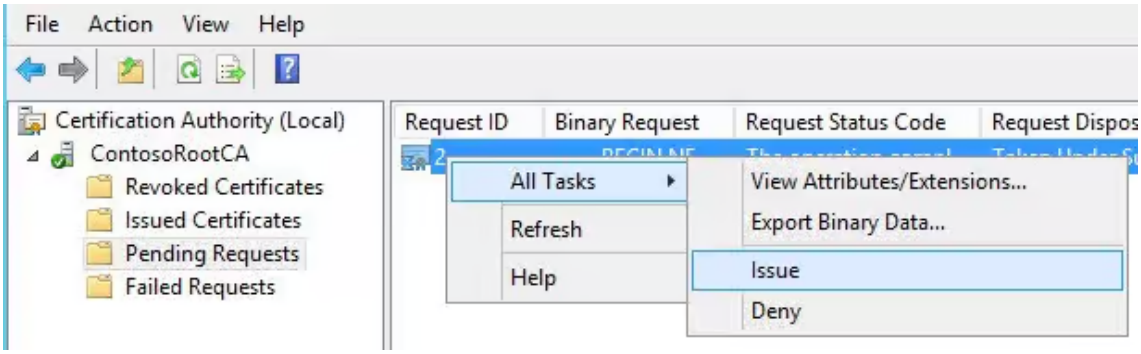
6. On the offline CA type the following command, using your filename:

certreq -submit D002MISC01.contoso.local_IssuingCA-D002MISC01.req

7. You will now see that the request is pending. Take note of the RequestId, as it will be unique to you.



8. Open the CA Manager snap-in on your offline root and issue the pending certificate.



9. While still on the offline CA, enter the following command to download the new certificate. Replace “2” with your request ID, and change the filename as you see fit.

certreq -retrieve 2 c:\D002MISC01.contoso.local_IssuingCA-D002MISC01.crt

10. Copy the certificate file to the online subordinate CA. Note: Do NOT place it in the pki directory. Run the commands below to install the new certificate. Once the certificate is installed, delete the file and empty the trashcan.

view sourceprint?

```
1  Certutil –installcert a:\ D002MISC01.contoso.local_IssuingCA-D002MISC01.crt

2  start-service certsvc

3  copy c:\Windows\system32\certsrv\certenroll\*.cr* d:\pki\
```

Configure Subordinate CDPs

1. Next up we need to configure the proper CRLs for our subordinate CA. Enter the following commands in an elevated Powershell on your subordinate CA.

view sourceprint?

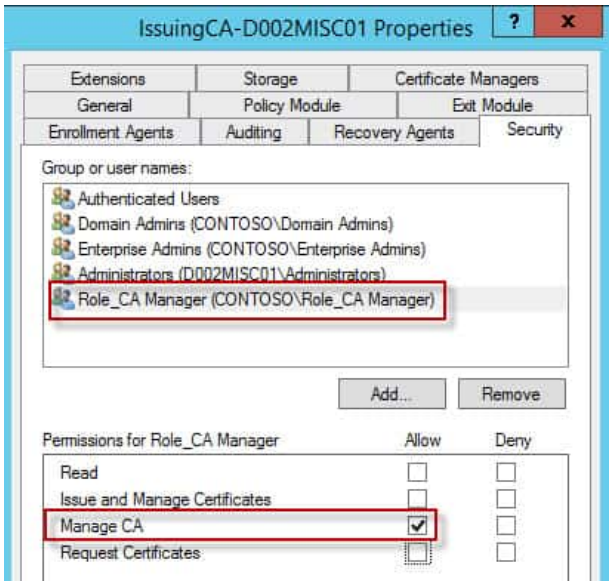
```
1  $crllist = Get-CACrlDistributionPoint; foreach ($crl in $crllist) {Remove-
   CACrlDistributionPoint $crl.uri -Force};

2  Add-CACRLDistributionPoint -Uri
   C:\Windows\System32\CertSrv\CertEnroll\%3%8%9.crl -PublishToServer -
   PublishDeltaToServer -Force

3  Add-CACRLDistributionPoint -Uri
   http://www.contoso.local/pki/%3%8%9.crl">http://www.contoso.local/pki/%3%8%9.crl
   -AddToCertificateCDP -Force

4  Add-CACRLDistributionPoint -Uri
   file://\\D002Misc01.contoso.local\pki\%3%8%9.crl"
   file://\\D002Misc01.contoso.local\pki\%3%8%9.crl -PublishToServer -
   PublishDeltaToServer -Force
```

```
5  $aialist = Get-CAAuthorityInformationAccess; foreach ($aia in $aialist)
   {Remove-CAAuthorityInformationAccess $aia.uri -Force};

6  Add-CAAuthorityInformationAccess -AddToCertificateAia
   http://www.contoso.local/pki/%1_%3%4.crt"
   http://www.contoso.local/pki/%1_%3%4.crt -Force

7  Certutil -setreg CA\CRLPeriodUnits 2

8  Certutil -setreg CA\CRLPeriod "Weeks"

9  Certutil -setreg CA\CRLDeltaPeriodUnits 1

10  Certutil -setreg CA\CRLDeltaPeriod "Days"

11  Certutil -setreg CA\CRLOverlapPeriodUnits 12

12  Certutil -setreg CA\CRLOverlapPeriod "Hours"

13  Certutil -setreg CA\ValidityPeriodUnits 5

14  Certutil -setreg CA\ValidityPeriod "Years"

15  certutil -setreg CA\AuditFilter 127

16  restart-service certsvc

17  certutil -crl
```

CA Delegation

1. Now that our online subordinate CA is up and running, for the most part, it is a good idea to delegate who has rights to manage the CA and issue certificates. I'm going to create two roles: One that can manage all aspects of the CA, and another that can just mint specific certificates. In AD create two groups: **Role_CA Manager** and **Role_Issue Certificates**. Or use whatever names you like.

2. On your subordinate CA, launch the CA MMC Snap-in. Right click on the CA name, open the properties, and select the **Security** tab, and add the **Role_CA Manager** group. Give it Manage CA permissions. If you want, you can remove rights from Domain Admins or Enterprise Admins, should you want to more tightly control CA access (which you should).



Summary

At this point in the configuration there are no published templates. So in the following post we will configure a couple of templates, and I'll show you how to delegate permissions so that other administrators can mint their own certificates. In this installment we've done the bulk of the subordinate CA configuration. At this point the CA is now functional, although no templates have been configured. So coming up in the next installment is, among other things, the process to configure templates and computer autoenrollment. Check out Part 3 here.