

Windows Server 2012 R2 Two-Tier PKI CA Pt. 1

 derekseaman.com/2014/01/windows-server-2012-r2-two-tier-pki-ca-pt-1.html

Derek Seaman

January 5, 2014



While I have written a number of articles focused on SSL certificates and templates, I have not done a mini-series on how to actually install a Windows Certificate Authority. For this series I'm using Windows Server 2012 R2, but the steps are pretty much identical for Windows Server 2012. Microsoft blogs have several PKI configuration series, which directly guided the content of this series. But I always have my own spin, so I think its worthwhile to do yet another blog post on configuring a MS CA...the "Mr. SSL" way.

Windows Server 2012 R2 Certificate Authority

The process is fairly simple: Build an offline root, create an online issuing CA, setup a couple of templates, setup auto-enrollment, then do a little post setup configuration. This requires two VMs, each running Windows Server 2012 R2 (or plain 2012 if you wish).

Building an enterprise CA is non-trivial, and should be highly process oriented. While this short series will provide the steps how to configure a two tiered hierarchy, it alone is not enterprise grade and ready for a fortune 500 company. Many operational procedures, access controls, etc. need to be defined by the organization. For example, who can issue certificates? Who can revoke them? Do users need PKI certificates or just computers? How about key recovery? Disaster recovery? Do you need a hardware security module (HSM)? Do you require FIPS compliance? What ciphers and hashing algorithms will you allow? Where do you store the offline CA?

As you can see, there are many questions and processes that need to be well documented for a solid PKI solution. However, for a lab environment where you want to test out a two-tiered model, then this short series is for you. Please don't take this solution as-is and throw it into production. You will have a false sense of security and possibly do more harm than good.

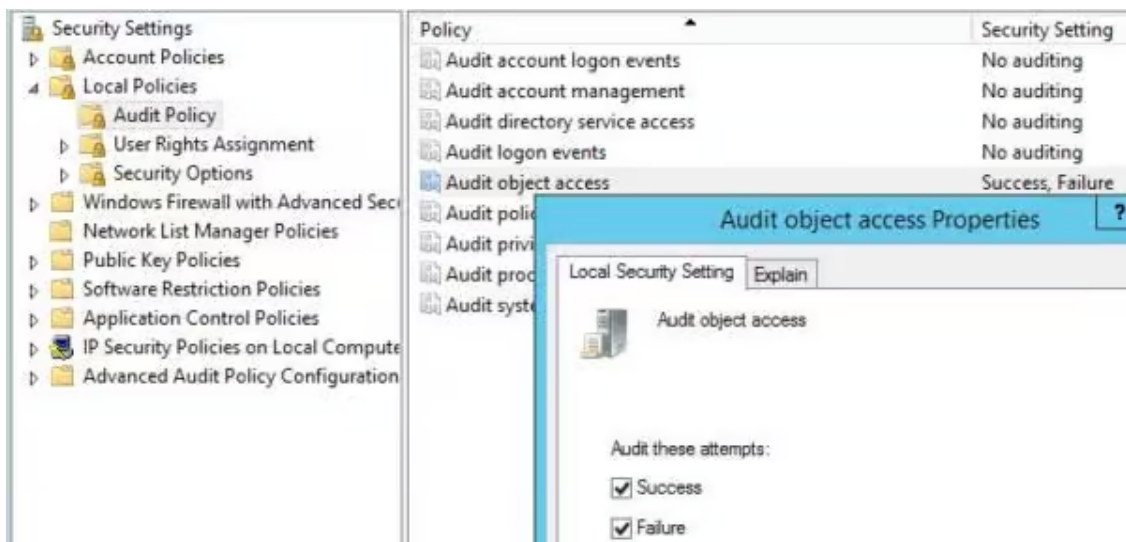
The Microsoft CA issues industry standard certificates (x.509), and thus will work with third party hardware and software. For instance, they will work perfectly fine on the Linux vCenter appliance, or your hardware load balancers. You just need to use the proper certificate template, and verify compatible algorithms.

Offline Root CA Hardening

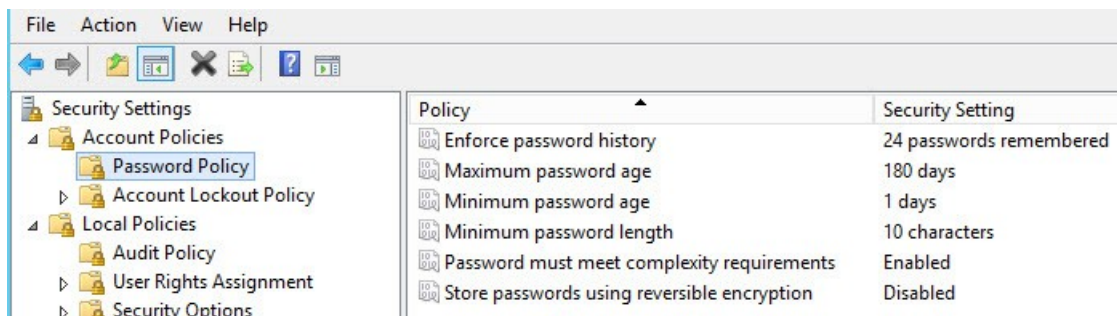
1. Provision a standalone Windows Server 2012 R2 server. I used vCenter 5.5 with customization specifications to create the VM. You can use the 'standard' edition of the OS since all SKUs in 2012 have the exact same feature set, unlike 2008 R2 and earlier. For security purposes I would not provision a NIC, or remove the NIC after you've built the CA to prevent future network attacks.

2. Configure a virtual floppy for the offline CA VM. This is a good way to transfer data between the offline CA and the subordinate, which is required during the configuration process. Yes you could connect a NIC, but then your offline CA is no longer offline and exposed to network attacks. Media needs to be read/write, so an ISO image will not suffice. You can use a tool like [WinImage](#) to create a floppy image.

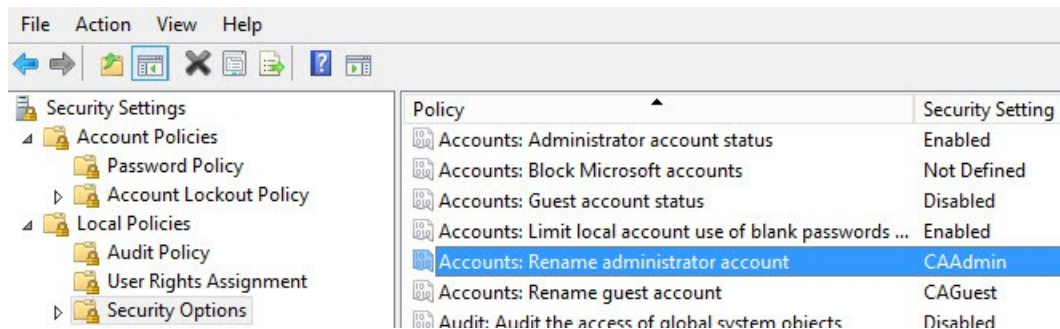
3. Open the local security policy and modify the Audit Object Access to record Success and Failures. This is needed to audit certain CA actions, in conjunction with a CA flag we will set later on.



4. Depending on your VM template hardening, you may or may not need to modify the password policy. Again in the Local Security editor. Modify to meet your organization's security requirements.



5. You should also rename the Administrator account, if that's not already built into your templates. Make sure to record the new name, or you could be in a pickle. For good measure I'd rename the guest account, although it should be disabled.



6. Obviously you should change the administrator password and not use your template default. Be sure to record the password in a secure location.

7. You should also think about where you will store the offline CA VM once it is build and this project is complete. If you leave it sitting on a production ESXi host, then it would be fairly trivial to power on the VM and compromise it. I would not call storing your “offline” CA in a powered off state on a production ESXi host “offline”. I would look at exporting the VM to an OVF file, then storing that file on removable media in a very secure location. You could use a DVD, Blu-Ray, or USB stick.

Install Offline Root CA

1. After your VM is provisioned and hardened, make sure the computer name is configured. In my case the offline CA is name D002CA01. Reboot if you changed the name.

2. Use Notepad and create a file called **CAPolicy.inf** in **C:\Windows**. Use the code snippet below, but change the URL. This URL is where your Certification Practice Statement (CPS) is located. It will also be where the CRL (certificate revocation list) will be published. For a production deployment you’d want to create a CPS, but for this exercise we will skip it, however the URL will be configured for future usage. For additional details see this [TechNet](#) link. You probably want to use a different URL like CA.yourdomain or PKI.yourdomain since we will be publishing other data to this address such as the CRL. For simplicity I stuck with www.contoso.local. Make sure the filename does not have any extra extensions like .txt. Verify from the command line.

[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

9  [Certsrv_Server]

10 RenewalKeyLength=2048

11 RenewalValidityPeriod=Years

12 RenewalValidityPeriodUnits=20

13 CRLPeriod=weeks

14 CRLPeriodUnits=26

15 CRLDeltaPeriod=Days

16 CRLDeltaPeriodUnits=0

17 LoadDefaultTemplates=0

18 AlternateSignatureAlgorithm=1
```

3. Run the following PowerShell command. Change the CACCommonName as needed. The command will complete instantly. I would make it clear in the name that this is the Root CA. This name will be present in all issued certificates, so make it obvious what it is and not just some generic hostname that is not meaningful. Notice that we are using SHA256 here, since SHA1 is no longer considered secure. You could also use SHA512.

[view sourceprint?](#)

```
1  Add-WindowsFeature Adcs-Cert-Authority -IncludeManagementTools

2

3  Install-AdcsCertificationAuthority -CAType StandaloneRootCA -
   CACCommonName "ContosoRootCA" -KeyLength 2048 -HashAlgorithm SHA256 -
   CryptoProviderName "RSA#Microsoft Software Key Storage Provider"
```

```

PS C:\Users\Administrator> Add-WindowsFeature Adcs-Cert-Authority -IncludeManagementTools

Success Restart Needed Exit Code      Feature Result
-----
True      No          Success      {Active Directory Certificate Services, Ce...}
WARNING: Windows automatic updating is not enabled. To ensure that your newly-installed role or feature is
automatically updated, turn on Windows Update.

PS C:\Users\Administrator> Install-AdcsCertificationAuthority -CAType StandaloneRootCA -CACommonName "ContosoRootCA" -Ke
yLength 2048 -HashAlgorithm SHA256 -CryptoProviderName "RSA#Microsoft Software Key Storage Provider"

Confirm
Are you sure you want to perform this action?
Performing the operation "Install-AdcsCertificationAuthority" on target "D002CA01".
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "Y"): y

ErrorId ErrorString
-----
0

```

4. Run the following commands, using the appropriate URL for your organization. We aren't using HTTPS here, because that requires SSL and certificate validation. This is just used to download the CPS and CRLs, so don't get clever and use HTTPS here. We will configure SSL for the web enrollment module, though.

[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.crl -PublishToServer -Force

3 Add-CACrLDistributionPoint -Uri http://www.contoso.local/pki/%3%8.crl -
   AddToCertificateCDP -Force

4 $aiaList = Get-CAAuthorityInformationAccess; foreach ($aia in $aiaList)
   {Remove-CAAuthorityInformationAccess $aia.uri -Force};

5 Certutil -setreg CA\CRLOverlapPeriodUnits 12

6 Certutil -setreg CA\CRLOverlapPeriod "Hours"

7 Certutil -setreg CA\ValidityPeriodUnits 10

8 Certutil -setreg CA\ValidityPeriod "Years"

9 Certutil -setreg CA\AuditFilter 127

10 restart-service certsrv

11 certutil -crl

```

5. Verify that two and only two CRL distribution points are configured.

[view sourceprint?](#)

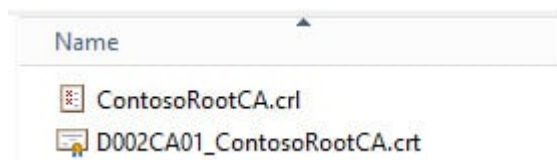
```
1 Get-CACRLDistributionPoint | format-list
```

```
PS C:\Users\Administrator> Get-CACRLDistributionPoint | format-list

PublishToServer      : True
PublishDeltaToServer : False
AddToCertificateCdp  : False
AddToFreshestCr1    : False
AddToCr1Cdp         : False
AddToCr1Idp         : False
Uri                  : C:\Windows\System32\CertSrv\CertEnroll\<CAName><CRLNameSuffix>.cr1

PublishToServer      : False
PublishDeltaToServer : False
AddToCertificateCdp  : True
AddToFreshestCr1    : False
AddToCr1Cdp         : False
AddToCr1Idp         : False
Uri                  : http://www.contoso.local/pki/<CAName><CRLNameSuffix>.cr1
```

6. Navigate to **C:\Windows\System32\CertSrv\CertEnroll**. You should see two files, one ending in CRL and another ending in .CRT. These two files need to be copied to what will be the online subordinate CA.



Publish Root CA to the Forest

1. Provision a Windows Server 2012 R2 VM which will be your online CA. Join it to the domain. In my case the VM is named D002MISC01. Do not try and be clever and use a Domain Controller. The server will later need IIS installed and access to local accounts, which is not possible on a DC. So use a member server for your online CA, even in a home lab.

2. Login to what will be your online subordinate CA with an account that is a member of both Domain Admins and Enterprise Admins. Mount the media which has the two files copied from your offline CA. Open an elevated Powershell and enter the following commands, using the file names for your instance. This will publish the offline root CA information to AD, just as if it were an online CA. By doing this all domain joined clients will automatically trust your root CA. If you have standalone computers, then you can import the .crt file into their trusted certificate store.

[view sourceprint?](#)

```
1 certutil -dspublish -f D002CA01_ContosoRootCA.crt RootCA
```

```
2 certutil -addstore -f root D002CA01_ContosoRootCA.crt
```

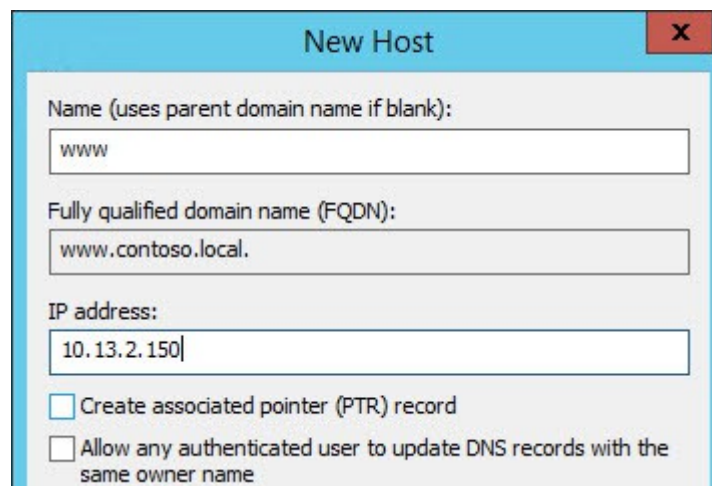


```
3 certutil -addstore -f root ContosoRootCA.crl
```

```
PS C:\temp> certutil -dsPublish -f D002CA01_ContosoRootCA.crt RootCA
ldap:///CN=ContosoRootCA,CN=Certification Authorities,CN=Public Key Services,CN=Services,CN=Configuration,DC=contoso,DC=local?cACertificate
Certificate added to DS store.
ldap:///CN=ContosoRootCA,CN=AIA,CN=Public Key Services,CN=Services,CN=Configuration,DC=contoso,DC=local?cACertificate
Certificate added to DS store.
CertUtil: -dsPublish command completed successfully.
PS C:\temp> certutil -addstore -f root D002CA01_ContosoRootCA.crt
root "Trusted Root Certification Authorities"
Signature matches Public Key
Certificate "ContosoRootCA" added to store.
CertUtil: -addstore command completed successfully.
PS C:\temp> certutil -addstore -f root ContosoRootCA.crl
root "Trusted Root Certification Authorities"
CRL "CN=ContosoRootCA" added to store.
CertUtil: -addstore command completed successfully.
```

CPS and CRL Distribution

1. Now you need create a DNS record for the host that will be publishing your online CA information. In this case it's D002MISC01, and per my previous steps I stuck with 'www' as the site name. I'm assuming the proper DNS zone already exists, since you have a domain with Active Directory up and running. This must be configured prior to continuing, as the subordinate will fail to properly configure if the CRL file is not available.



2. We need to install IIS, since we will be distributing the CPS and CRL via the HTTP. On the VM which will be your online CA, run the following command:

Install-WindowsFeature Web-WebServer -IncludeManagementTools

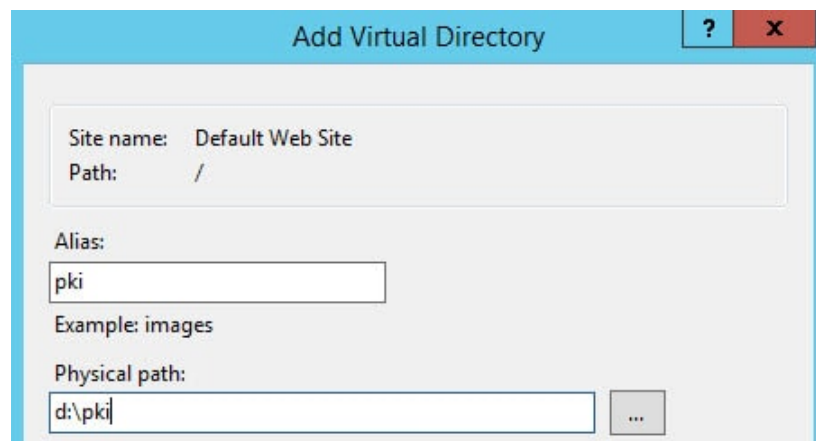
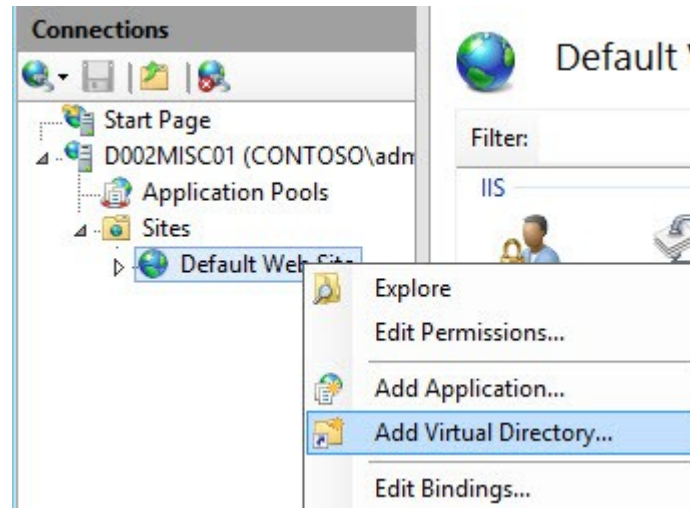
3. Open an elevated PowerShell and enter the following commands. If you have an official CPS, then you can skip the second command and just copy your cps.txt file to the directory. For security purposes I'd recommend putting the files on the D: drive, so you aren't serving content from the OS drive.

[view sourceprint?](#)

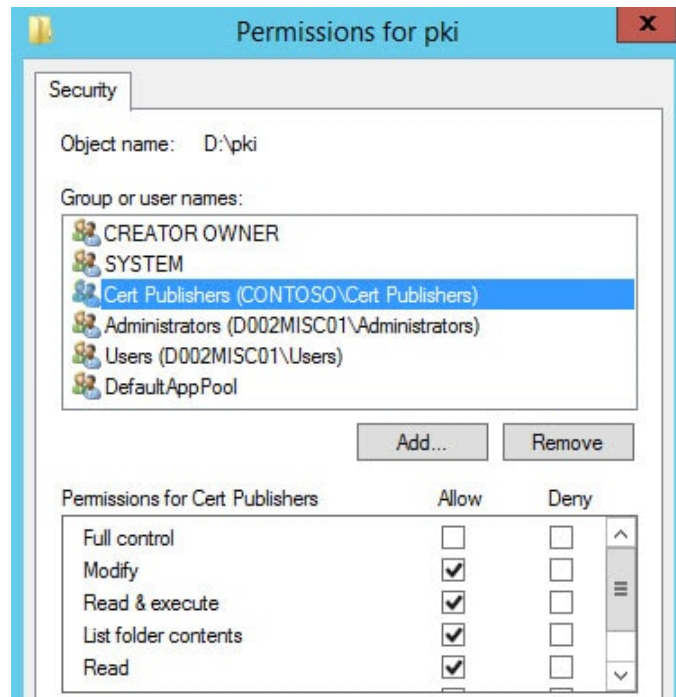
```
1 new-item -path D:\pki -type directory
```

- 2 `write-output "This is a sample CPS. Modify as needed." | out-file D:\pki\cps.txt`
- 3 `new-smbshare -name pki D:\pki -FullAccess SYSTEM,"Contoso\Domain Admins" -ChangeAccess "Contoso\Cert Publishers"`

4. Open the IIS Manager and add a Virtual Directory as shown below.



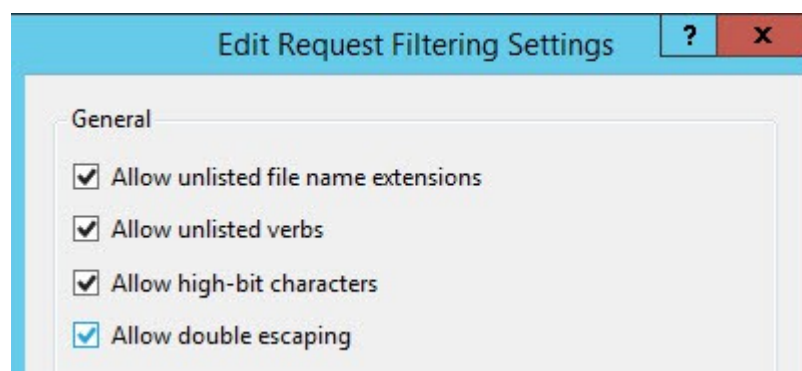
5. Verify **pki** is selected in the left pane, then single click **Authentication** in the middle pane, and in the right **Actions** pane click on **Edit Permissions**.
6. Select the **Security** tab and select **Edit**. Add the **Cert Publishers** group with **Modify** permissions (which will add several others under it).



7. In the same dialog box, click **add** but change the **from this location** to the local computer. Manually enter **IIS AppPool\DefaultAppPool**. Leave the default permissions. If you use the user/group browser this will not be listed, so please manually enter it.

8. At this point any anonymous browser can now read your CPS statement and see the public root certificate. You can test this by going to **http://www.yourdomain/pki/cps.txt** and verify the sample file opens.

9. In the middle pane, with pki still selected, click once on **Request Filtering**. In the right pane click on **Edit Feature Settings** and check the box next to **Allow double escaping**.



10. Run **iisreset** from an elevated Powershell command.

Summary

In this installment we've configured our offline root CA, performed some hardening, and published the root CA information to the domain. All computers in the domain will now trust your root CA. We also configured IIS to serve up your CPS and CRLs to anonymous users. Next up is configuring the online subordinate CA. Check out the next installment in [Part 2](#).