Part I Design and Planning

Chris Delay here again. This is part of a five part series. In Part I, I will cover design considerations, and planning for deploying a PKI. When implementing a PKI planning is the most important phase, and you can prevent a lot of issues by properly planning your PKI implementation.

I recommend reading the following MSPress books on PKI and Certificate Services before implementing a Windows PKI, or any PKI for that matter. Both books are written by Brian Komar.


They are both excellent resources for anyone implementing, managing, or designing solution that use a Microsoft PKI. And both books go into far more detail then I can here.

**Why Deploy a PKI?**

There are a host of reasons to deploy a PKI; a few are listed here:

- Control access to the network with 802.1x authentication
- Approve and authorize applications with Code Signing
- Protect user data with EFS
- Secure network traffic IPSec
- Protect LDAP-based directory queries Secure LDAP
- Implement two-factor authentication with Smart Cards
- Protect traffic to internal web-sites with SSL
- Implement Secure Email

In addition a number of applications can use certificates in some fashion. Here is a brief list:

- Active Directory
- Exchange
- IIS
- Internet Security & Acceleration Server
- Office Communications Server
- Outlook
Another thing to consider is what future applications you may need to support with your PKI. This may not be an answerable question, nor should you be expected to know for sure. In fact some of the applications or technologies that your PKI may be required to support may not have even been conceived of yet. My point here is that your design should incorporate plenty of flexibility. Not only do you want to deploy a PKI solution that supports existing technologies, but one that is scalable, and can support future technologies.

**Costs**

The next thing you want to think about is cost. I understand how difficult it can be to get budgets approved in any business. Despite our wishes as technology professionals that we could implement the appropriate solutions, sometimes we are handicapped by the budget that we have to complete a project. How much money is your business willing to invest in the PKI solution? What are the costs for implementing a PKI? Here are some items that may need to be included in your budget:

**Hardware Costs**

- Servers
- Hardware Security Modules (HSMs)
- Backup Devices
- Backup Media

**Software Costs**

- Windows Server Licenses

**Human Capital**

- Paying someone to design, implement, and manage the PKI infrastructure.

**Cost Savings**

While you are planning your budget, it is important not to forget the cost savings that a Windows PKI solution can provide. The two key areas of savings that I see in a PKI solution are:

**Integration**

Microsoft CA’s, especially Enterprise CAs, have a tight integration with Microsoft products. The integration makes managing and requesting certificates from Microsoft Operating Systems and applications fairly straightforward, to the point that you do not really need any PKI experience to be able to request a certificate.

**Automation**

The greatest advantage of the Windows PKI solution is automation. An Enterprise CA is tightly integrated with Active Directory. Using autoenrollment, a simple group policy can be configured to automate the deployment of certificates to computers and users. The deployment is so transparent, that users do not have to do anything to request a certificate.

**Manageability**
In designing your PKI solution you will have to take into account the resources you have to manage the PKI solution. Day-to-day management for the most part is very limited, but you will need someone to provide the care and feeding of your PKI. You will need someone to issue and revoke certificates. You will need to have someone manage the hardware, apply patches, take backups. In other words, you need a Server Administrator. Also, you will need to have someone that publishes Certificate Revocation Lists and manages the CA itself.

**Security**

You will need to determine the level of security required for your PKI. In order to determine the level of security it is important to step back and understand what a Public Key Infrastructure and the certificates associated with the Public Key Infrastructure can be used for. Certificates can be used for identification, encryption, non-repudiation, and in some cases authentication. In your organization you probably have some standard on how a user receives a user account. When hired there was some form indicating that he/she needs a domain user account and the manager approves this form; in other words, the manager was assuring the identity of the user. Since certificates can be used for identification the same standard should be used when issuing certificates, if they are going to be used for that purpose. If you are using certificates just for encryption, you may be less concerned with the user's identity. If using the keys from the certificate for encryption, it would depend on what data is being decrypted. If a user is just encrypting his/her recipes you may perhaps not require the same level of protection of the private keys as you would if the user is encrypting top secret government documents. In other words the level of security is going to be determined by the level of risk. This determination should include any corporate security policies for PKI and certificates. When creating your PKI security policy, you should also consider any industry or government regulations.

**Flexibility and Scalability**

The flexibility and scalability of your solution should be taken into consideration. If you have a high level of confidence that you will not need to change or adapt your PKI solution you can have a fairly simple design. However, if you need a solution that will need to support a variety of technologies, different levels of security, and a global presence, then your solution can get much more complicated.

**Physical Design**

When designing your PKI solution you will have to determine the hierarchy that you will use. There are generally three types of hierarchies, and they are denoted by the number of tiers.

**Single/One Tier Hierarchy**

A single tier Hierarchy consists of one CA. The single CA is both a Root CA and an Issuing CA. A Root CA is the term for the trust anchor of the PKI. Any applications, users, or computers that trust the Root CA trust any certificates issued by the CA hierarchy. The Issuing CA is a CA that issues certificates to end entities. For security reasons, these two roles are normally separated. When using a single tier hierarchy they are combined. This may be sufficient for simple implementations where ease of manageability and lower cost outweigh the need for greater levels of security or flexibility. The level of security can be enhanced if the CA’s keys are protected by an HSM, but at the expense of higher equipment and management costs.

**Two Tier Hierarchy**
A two tier hierarchy is a design that meets most company’s needs. In some ways it is a compromise between the One and Three Tier hierarchies. In this design there is a Root CA that is offline, and a subordinate issuing CA that is online. The level of security is increased because the Root CA and Issuing CA roles are separated. But more importantly the Root CA is offline, and so the private key of the Root CA is better protected from compromise. It also increases scalability and flexibility. This is due to the fact that there can be multiple Issuing CA’s that are subordinate to the Root CA. This allows you to have CA’s in different geographical location, as well as with different security levels. Manageability is slightly increased since the Root CA has to be brought online to sign CRL’s. Cost is increased marginally. I say marginally, because all you need is a hard drive and Windows OS license to implement an Offline Root. Install the hard drive, install your OS, build your PKI hierarchy, and then remove the hard drive and store it in a safe. The hard drive can be attached to existing hardware when CRLs need to be re-signed. A virtual machine could be used as the Root CA, although you would still want to store it on a separate hard drive that can be stored in a safe.

Three Tier Hierarchy

Specifically the difference between a Two Tier Hierarchy is that second tier is placed between the Root CA and the issuing CA. The placement of this CA can be for a couple different reasons. The first reason would be to use the second tier CA as a Policy CA. In other words the Policy CA is configured to issue certificates to the Issuing CA that is restricted in what type of certificates it issues. The Policy CA can also just be used as an administrative boundary. In other words, you only issue certain certificates from subordinates of the Policy CA, and perform a certain level of verification before issuing certificates, but the policy is only enforced from an administrative not technical perspective.

The other reason to have the second tier added is so that if you need to revoke a number of CAs due to a key compromise, you can perform it at the Second Tier level, leaving other “branches from the root” available. It should be noted that Second Tier CAs in this hierarchy can, like the Root, be kept offline.
Following the paradigm, security increases with the addition of a Tier, and flexibility and scalability increase due to the increased design options. On the other hand, manageability increases as there are a larger number of CAs in the hierarchy to manage. And, of course, cost goes up.

**Security**

**Private Key Protection**

One of the key aspects of designing a PKI solution is to make sure the proper controls are in place. Security for a PKI solution mostly revolves around protecting the private key pair of the CA. Each CA has a private/public key pair. The private key is used to sign CRL’s as well as certificates that are issued by the CA. Clients and application verify the signature so that they can be assured that a certificate was issued by a particular CA. If you install a Microsoft CA, the private key is protected by software, or more specifically the Data Protection API (DPAPI). Although this method does provide protection it does not prevent a user that is a member of the Administrators group on the CA from accessing the private key. This can be a cause for concern, because you may have administrators whose job is just to patch the system, and yet they have access to the private key which violates the concept of least privilege.

There are generally two methods for protecting the private key of a CA. The first method is to keep the CA offline and the hard drive stored in a safe. By controlling the conditions the hard drive can be used, the opportunities for key compromise are reduced. The second method is to use a hardware device to protect the private key. For example, a smartcard can be used to store the private key of the CA. This is not the best solution since the smart card must remain in the reader attached to the CA in order to be used. Also, a smart card may not be as resilient, or provide the level of security that is required. It is however a low cost solution. A more standard solution is to use a Hardware Security Module (HSM). HSM’s are fairly expensive, but are normally certified for FIPS compliance -- a standardized measure of relative security. HSM’s are accepted as the most secure way to protect the private key for a CA.

**Role Separation**

Aside from private key protection you will most likely want to have some control as to the level of administrative access to a CA. In some cases you may have administrators that are responsible for performing every function on the CA. But in larger or higher security environments you will want to have some granular control over what access different role holders have. Below is a list of common roles on a CA:

- CA or PKI Administrator whose role is to manage the CA itself.
- Certificate Manager who issues and revokes certificates.
- Enrollment Agent is typically a role used in conjunction with smart cards; an Enrollment Agent enrolls for a certificate on behalf of another user.
- Key Recovery Manager if using key archival. If you are using key archival, the Key Recovery Manager is responsible for recovering private keys. Also, if you are using EFS an EFS Recovery Agent role may be created to recover data encrypted using EFS.

In addition to these roles that have direct interaction with the CA, you also will have ancillary roles that support the CA. These include:

- Backup Operator who is responsible for backing up the CA and restoring data in case of failure.
- Auditor who is responsible for reviewing audit logs and ensuring policy is not being violated.

**Physical Security**

Certificates issued by CAs are used in many cases for very sensitive operations such as establishment of identity, authentication and encryption. As such, it is important to not only protect the private key but to protect physical access. Law #3 of the Ten Immutable Laws of Security, states: “If a bad guy has unrestricted physical access to your computer, it's not your computer anymore.” For this reason you will want to protect physical access to the CAs. This will depend on the resources you have available, but typically in larger
organizations the CAs are stored in a locked cage in a data center. Only individuals that need physical access to the CA to perform their duties should be given access.

Policy

Generally the security requirements, such as those mentioned above, are dictated by a corporate security policy. A security policy usually takes into consideration regulatory and industry requirements as well as unique requirements for the individual company. The policy may also specify technical aspects of the PKI such as the encryption algorithms that must be used as well as operation of the Certificate Authorities.

In addition to security policies there may be CA-specific policies that need to be developed before implementing the PKI. These include Certificate Policy and Certification Practice Statement. The Certificate Policy explains what methods are used to establish the identity of a subject before issuing a certificate. A Certification Practice Statement outlines the operation of the PKI from a security perspective. Many companies, especially third parties companies that issue certificates, have their Certificate Policies and Certification Practice Statements available publicly. It may be helpful to view one of these public documents when writing your own policy documents.

Additional Security Considerations

In addition to the topics discussed it is important to apply any relevant security patches to your online CAs and to install them in a timely manner. In addition to patches, you should have an anti-malware solution installed on your CA.

So far we have covered reasons to deploy a Public Key Infrastructure. We also have covered the various costs involved in a PKI infrastructure, as well as the impact of various design considerations. Now we will dive a little deeper into specific configuration decisions and technical aspects of the Certificate Authorities.

CA Configuration

Certificate Validity Period

Digital certificates have a lifetime, a start date and an end date for which they are considered valid. You should determine what values for this lifetime are appropriate for each CA certificate and end-entity certificate issued by your CA’s. For CA’s, this lifetime is set when the CA is installed and when the private key is renewed. For end-entity certificates there are a number of factors taken into account:

- These include the validity period for the issuing CA. The CA will not issue certificates that are valid past the CA’s lifetime.
- The validity period specified in the Certificate Template.
- The value of this registry key, specified in this KB article: http://support.microsoft.com/kb/254632

The certificate issued will be configured with the validity period that is the shortest of these items.

Key Length

The length of a key definitely affects security of information protected with that key. Thus, you will need to determine the key lengths you will use with each key pair. First you will need to determine the key lengths that will be used for each of the CA key pairs. Additionally, you will need to determine the key lengths for any certificates issued by the issuing CA. The key lengths for the CA certificates are determined by the key size requested when the CA is installed and when the key pair is renewed. The key length at installation is set during the CA Setup process. The key length for renewal is determined by a value set in the CAPolicy.inf configuration file installed on the CA.

For certificates issued by the issuing CA the maximum key size is limited by the CSP that is being used. The specific key size that is required can be specified in the certificate request or in the Certificate Template if using an Enterprise CA. As a general guideline, the
longer the lifetime of the certificate the longer the key length should be. For applications that will be using certificates you will need to determine the maximum key length they support. Some applications have limitation on the key size not only in the actual certificate it is using, but also for any certificates in the CA hierarchy. From a security standpoint it is recommend that 4096 bit key is used for Certification Authorities key pair. However, if you wanted to insure maximum compatibility with network devices and applications a 2048 bit key would be the better choice.

**AIA Locations**

When a client or application is validating a certificate it needs to not only validate the certificate that is being used but also the entire chain of the certificate. In other words, the application or client needs a certificate from each CA in the chain beginning with the issuing CA and ending with the Root CA. If the application or client does not have access to the certificates in the chain locally the application or client needs a place from which to obtain the certificates. This location is called the Authority Information Access or AIA. The AIA location is the repository where the CA certificate is stored so that it can be downloaded by clients or applications validating a certificate. The AIA location is included in the AIA extension of a certificate. Before implementing your PKI it is important to think about what types of clients will be validating the certificates and where they reside. If you are using Windows clients that are internal to your network and are domain members then LDAP locations in Active Directory are a good place for clients to access the AIA repository. If you have non-Windows clients or Windows clients that are not domain members that are internal then an internally hosted web site would be the ideal location for the AIA repository. However, if clients may need to validate a certificate when outside the network, then you will need an AIA repository that is available externally, perhaps on the public network.

**CDP Locations**

A CRL Distribution Point (CDP) is where clients or applications that are validating a certificate download the certificate revocation list (CRL) to obtain revocation status. CA’s periodically publish CRLs to allow clients and applications to determine if a certificate has been revoked. CRLs contain the serial number of the certificate that has been revoked, a timestamp indicating when the certificate was revoked, as well as the reason for revocation. Similar to AIA Locations, you need to keep in mind what types of clients you are supporting and where they are located.

**CRL Validity and Overlap Periods**

Like certificates, CRLs have a start date and an end date denoting a period for which they are valid. As such, you will need to consider what the CRL lifetime should be for each CA. In general, the CRL lifetime is proportional to the number of certificates the CA is expected to issue. Offline CA’s that issue relatively few certificates, and those only to other CAs, would tend to have CRLs with a more extended lifetime, for example, six months to a year. This reflects the fact that, in a properly managed PKI, an offline CA would rarely revoke a certificate. Issuing CAs, on the other hand, can be expected to issue large numbers of certificates to end-entities. It is quite common to revoke an end-entity certificate for any number of reasons, so the lifetime of an issuing CA’s CRL can be quite short; a few days or even hours.

Another thing to consider is the overlap period. The overlap period is a short time interval beyond the expiration date of the CRL, and reflects the period between when a new CRL is published, and when the old CRL actually expires. During this time both CRLs are valid. This allows time for the new CRL to replicate to all of the repositories before the old CRL expires.

**Delta CRLs**

Delta CRLs are CRLs that contain revocation information for certificates that have been revoked since the base CRL was last published. For example, you have a Certificate A and it is revoked. The CA then publishes a new Base CRL that includes the revocation information for Certificate A. Shortly thereafter, Certificate B is revoked. At the designated interval a Delta CRL is published which contains the serial number and reason for revocation for Certificate B. When a client needs to determine revocation status Certificates A or B it downloads both the base CRL and the Delta CRL. The client determines that Certificate A is revoked from the base CRL, and then determines that Certificate B is revoked from the Delta CRL.
The reason for Delta CRLs is due to limitations with base CRLs. Base CRLs can grow rather large over time as they contain the serial number and revocation reason for every valid certificate that has been revoked from a CA. Instead of publishing a large CRL over and over again, revocation status can be updated with the smaller Delta CRL. In this way clients that have a time valid CRL will just need to download the Delta CRL. Like base CRLs you will need to determine how often Delta CRLs are published. It should be noted that the use of Delta CRLs is completely optional and is not normally used with offline CAs for obvious reasons.

**OCSP URIs**

Windows Vista, Windows 7, Windows 2008 and Windows 2008 R2 can obtain revocation information from an Online Responder via the Online Certificate Status Protocol. If you are using an Online Responder to provide revocation status, you should include the URI that points to the Online Responder.

**Microsoft PKI**

Other than the benefits of the Windows PKI, most of the things I have mentioned so far apply to any Public Key Infrastructure. I am now going to focus on a Microsoft-specific implementation.

**Operating System Version**

Currently both Windows 2003 and Windows 2008 are supported so you need to determine which OS you’re going to use for your CAs. In order to make that decision you will need to know what additional features Windows Server 2008 has over Windows Server 2003. Here are a few of the many new features in Windows Server 2008:

- Supports Suite B Cryptographic Algorithms
- New Crypto API called Cryptography Next Generation (CNG)
- PKIView is installed with the Certificate Services Role (Really nice when troubleshooting)
- Certificate Authorities can be clustered (Active/Passive)
- Version 3 Certificate Templates
- New Certificate Templates
- Network Service permissions can be configured on Version 3 Templates
- Here is an article that goes into more detail on new features in Windows 2008 PKI

**Windows Server 2008 R2**

Windows Server 2008 R2 adds a number of new features to Certificate Services. These features include:

- **Cross-Forest enrollment** - Windows 2008 R2 Supports Cross-Forest enrollment which will allow a CA or multiple CAs in one forest to support clients in multiple Forests.
- **Certificate Enrollment Web Service and Policy Service** - Allows clients to enroll for certificates over web interfaces. This new capability allows clients to retrieve certificates even if they are not located on the same physical network as Active Directory and the CA. Clients query the Enrollment Policy Service, to determine which Certificates they should enroll for, the
Enrollment Policy Service contacts Active Directory and responds to the client with CA and Certificate Template information. The client then queries the Enrollment Web Service, to enroll for certificates. The Enrollment Web Service then contacts the CA on behalf of the client, and returns the enrolled certificates back to the client.

- **Non-persistent certificates (not stored in the CA database)** - Certificate Templates can be configured to not store certificates in the CA database. This is useful for CAs that issue certificates for network authentication, in which certificates have a lifetime of hours or days and the storage of the certificates in the database would impact CA performance unnecessarily.

**Operating System Edition**

There are three editions of the OS on which you can install the Certificate Authority role. Those editions are Standard, Enterprise, and Datacenter. Standard or Enterprise Editions are normally used.

Below are the key features that Enterprise and Datacenter Edition supports and Standard Edition does not. It is important to note that Datacenter Edition does not offer any additional functionality in terms of Certificate Services over Enterprise Edition.

- Version 2 Templates
- The ability to duplicate and modify Certificate Templates
- Certificate Autoenrollment (requires version 2 templates)
- Common Criteria Role Separation enforcement
- Key Archival and Retrieval

**Certificate Authority Type**

Next, you need to consider what type of Windows CA is required. A Standalone CA does not require Active Directory and can be installed on a non-domain member server. Requests for certificate enrollment can be sent through Web Enrollment if installed, or by sending a request file to the CA. The request files are usually generated through the `Certreq.exe` tool. Also, in Windows Server 2008 and Vista you can use the **Certificate Management Console to build custom requests.**

An Enterprise CA is integrated with Active Directory and requires AD in order to function. The Enterprise CA supports the same enrollment methods as the Standalone CA. In addition, however, an Enterprise can receive requests submitted through the Certificates MMC console. An Enterprise CA also allows for computer or user Autoenrollment which allows certificate request and issuance to be automated through a Group Policy setting. Enterprise CAs also use certificate templates which allow define what types of certificates a user or computer request from the CA.

Additionally, PKI related configuration information is stored in Active Directory. This makes it easy for applications and clients to locate an issuing CA, associated Certificate Templates, CRL’s and AIA information.

**Certificate Templates**

Certificate Templates allow you to create a template from which the values in the certificate request are generated. When a user requests a certificate based on a Certificate Template, the request generated by the client is built based upon the configuration of the template. Certificate templates are key to Autoenrollment, since Autoenrollment requires certificate templates. The importance of certificate templates is that it reduces management cost, and makes the CA easier to use. This is due to the fact that the user does not have to use complicated methods such as using `certreq.exe` and the associated configuration file to generate requests. Certificate Templates are stored in Active Directory and replicated to all domain controllers in the forest. This makes them highly available in order to support all clients in the forest.

For more information about Certificate Templates, please visit the following URLs:
Enrollment

Another important aspect to consider is how clients will enroll for certificates. There are several methods to enroll for certificates and the actual method you choose may vary. But for the majority of customers most enrollments will be done through Autoenrollment. For one off situations and testing enrollment, manual enrollment or Web Enrollment will typically be used. Beginning with Windows 2008 there is Network Device Enrollment Service that can be used for network devices to enroll.

Autoenrollment

Autoenrollment adds a high level of automation to certificate issuance. In order to deploy Autoenrollment, you must first configure a certificate template from which clients will generate their requests. You then need to enable a Group Policy setting that enables Autoenrollment. Once you enable autoenrollment the computer or user, depending on what template is configured will automatically request a certificate based on the template after Group Policy refreshes.

For more information about Autoenrollment, please visit the following URLs:


Manual Enrollment

There are several different methods for manual enrollment. The first method is to use the Certificates Management Console to enroll for certificates directly. The second method is to use the Certificate Management Console to generate a request file that can later be submitted to the Certification Authority. Third, a tool called certreq.exe can be used to create requests that can later be submitted to the CA. Certreq.exe can also be used to submit the request to the CA and also to download, and install the resulting certificate.

Web Enrollment

Web Enrollment is a web page that can be used to submit requests and download issued certificates from a CA. Web enrollment has typically been used to generate custom requests. However, with the ability to create custom requests in Windows Vista and Windows 2008 there is less of a need to use Web Enrollment.

Application Specific Enrollment

Many Microsoft applications, such as Internet Information Services and Office Communication Server have built in wizards that assist with enrolling for certificates that are used by those applications.

Network Device Enrollment Service (NDES)

NDES is a role that uses the SCEP protocol to allow network devices to enroll for certificates.

Additional Considerations

Key Archival and Retrieval
Key Archival is a feature that allows the CA to archive the private key associated with a certificate in its database. A Key Recovery Manager can then recover the private key for a certificate if required. Although separate from Key Archival, an EFS Recovery Agent can be configured to recover EFS encrypted files.

For more information about Key Archival and Retrieval, please visit the following URLs:


**Active Directory and Group Policy**

Aside from Autoenrollment, Active Directory and Group Policy allow the configuration of PKI related settings for clients. This includes EFS-related configuration, automatically publishing Root CA certificates to the Trusted Root Certification Store on clients, revocation checking configuration, and more...

For more information about PKI-related Group Policy settings, please visit the following URL:


**Conclusion**

I attempted to cover many aspects of the PKI implementation you will need to consider before deploying a Public Key Infrastructure. There is a great deal of information out there. As I mentioned before I would strongly recommend reading the Brian Komar books. I would also invite you to review PKI related posts on the Ask Directory Services Team Blog:


and the Windows PKI Blog:


As well as the Best Practices for Implementing a Microsoft Windows Server 2003 Public Key Infrastructure, which is located at:


And the Windows Server 2003 PKI Operations Guide that can be downloaded from:


---

**Part II Implementation Phases and Certificate Authority Installation**

Chris here again. In **Part I** of this series we covered design considerations for implementing a PKI. In this part of the series I will cover the steps required to implement a PKI. We will cover the steps and build the configuration files we will use for implementing the PKI.
Here is a breakdown of the steps for implementing a PKI. The steps can vary somewhat, but below are my recommendations to help ensure that the actual implementation is painless.

**Planning Phase**

During the planning phase you document the requirements of your PKI.

You then take the requirements and translate them into a physical design.

**Pre-Implementation Phase**

During this phase you will build the configuration files. There are essentially two configuration files. One is the CAPolicy.inf file. The CAPolicy.inf file is used during the installation of Active Directory Certification Services (ADCS) and during the renewal of the CA certificate. In the CAPolicy.inf you will configure some basic settings for the CA.

The second configuration file is the Post-Installation script. Although not required, using a Post-Installation script will allow you to quickly configure the new CA. Also, since you can review the script before implementation, it helps alleviate any misconfiguration due to typos.

**Testing Phase**

This is the phase that is most often avoided and to be honest I am not quite sure why. I would recommend walking through the installation process of the CA hierarchy at least once if not multiple times in a test environment. With the rise of virtualization software (Hyper-V, VMWare) it is fairly easy to do testing, even with limited access to hardware. Testing the implementation not only will get you familiar with the installation process, but will also help you identify any potential issues with your installation steps. Also, during testing you can thoroughly document the steps that you will take during the installation in the production environment.

**Implementation Phase**

During this phase you will implement the actual CA hierarchy. The implementation phase would also include any configuration that needs to be done after ADCS is installed.

**Walkthrough**

I am going to walk through these phases with you, so you feel more comfortable with the process. In my walk through, we are going to be installing a PKI for a fictional company called Fabrikam.

**Pre-Implementation Planning with Fabrikam**

We met with Fabrikam and Fabrikam is interested in deploying a Public Key Infrastructure to be leveraged by a number of current and future projects. Fabrikam currently has its corporate offices located at one site, although the company is rapidly expanding and may add additional sites in the future. Fabrikam currently either uses certificates from Commercial Certification Authorities or has put on hold projects that require certificates until the internal PKI is put in place. We discussed the importance of having a secure PKI deployment.

We discussed with Fabrikam the fact that a Hardware Security Module (HSM) would provide stronger key protection then the protection offered by DPAPI. However, Fabrikam has decided at this point to forgo the use of an HSM, as they felt the use of an HSM was not a requirement for them. After some discussion it was determined that Fabrikam would deploy a two tier hierarchy which would provide key protection for the Root CA. If security requirements change Fabrikam will consider implementing Common Criteria Role Separation in the future to further increase the security of the CA and the Private Key.
To additionally increase the level of private key protection the root CA will be kept offline. The hard drive of the offline Root CA will be stored in a safe that only certain members of the security team have access to. Access to the safe is logged and monitored. This protects the private key of the Root CA. This also limits the ability of an attacker or malicious employee to create additional CAs from the key pair of the Root CA.

Fabrikam is also interested in having the ability to add additional Issuing CAs in the future. They want the flexibility to add more issuing CAs based on load or geography. To increase the level of manageability they would like all current and future issuing certificate authorities to chain up to the same Root CA (Trust Anchor).

In terms of certificates Fabrikam has a list of requirements that must be met. They have some third party applications and devices that can use Secure LDAP, and they would like to issue certificates to Domain Controllers to support those apps. Also, Fabrikam is looking to deploy Outlook Web Access and has several intranet sites that currently have SSL configured. Right now, Fabrikam purchases certificates for these web sites from a third-party and would like to move this process in house to save money. In the future, Fabrikam is also looking at other technologies that could benefit from the internal PKI such for protecting wireless networks, and enabling remote access for user through VPN. However, these are future initiatives and Fabrikam just needs the flexibility to issue certificates in the future to support these technologies.

Fabrikam has also decided that it would like the issuing CA to be valid for 5 years to limit the lifetime of the public/private key pair. They plan to renew with a new key pair at the 3 year mark. They have also decided with some consultation that they would like to limit the root CA certificate to being valid for 10 years and plan to renew with a new key pair at the 5 year mark.

We looked into the applications and network devices that Fabrikam currently uses and they all support key lengths of 4096 bits or less. Fabrikam is going to make it a requirement that all future applications and network devices support key lengths as large as 4096 bits.

For the Root CA, Fabrikam was looking for a compromise between security and manageability in terms of renewing the Root CRL. The security team wanted to renew the CRL for the Root CA every month and the operations team wanted to have the Root CA renewed once a year, to limit the number of times they needed to bring the Root CA online. Eventually a compromise was reach of publishing a new CRL for the root CA after every 6 months. Similar discussions were held for determining the CRL validity period for the Issuing CA. It was eventually determined that a base CRL would be valid for a week and delta CRLs would be valid for 2 hours. Microsoft warned Fabrikam that if there was a hardware failure on the CA, there would be certificate validation issues after 2 hours due to the Delta CRLs expiring. Fabrikam plan to have a process in place to perform offline CRL signing in case of a CA failure.

We also discussed placement of AIA and CDP paths. Most of Fabrikam’s systems are Windows domain joined clients. However, they would like to have the flexibility for third party Operating Systems to access the CDP and AIA locations. Based on this discussion we decided to publish CRLs and CA certificates to both Active Directory and to a web server.

After having the discussions with Fabrikam, we are ready to look at the business requirements and translate those requirements into the implementation steps.

**Planning Phase walkthrough**

After reviewing the business requirements, Fabrikam documented the following design and configuration.

**PKI Configuration**

Number of Tiers: 2  
Number of CAs in each Tier: 1  
CA Type: Root CA will be Standalone. Issuing CA will be Enterprise.
CA Issuance: Domain Controller Authentication and Web Server
Private Key Protection: Root will be offline. No additional private key protection for Issuing CA.
Policy: No constraints will be defined.

Root CA Configuration

Validity Period for Root CA Certificate: 10 years
Key Length for CA Certificate: 4096 bits
Certificate Validity Period for issued Certificates: 5 years
AIA Locations: LDAP and HTTP
CDP Locations: LDAP and HTTP
CRL Validity: 26 Weeks
Delta CRLs: Delta CRLs will not be used
CA Name: Fabrikam Root CA

Issuing CA Configuration

Validity Period for Issuing CA Certificate: 5 years (determined by certificate validity period of root CA.)
Key Length for CA Certificate: 2048 bits
Certificate Validity Period for issued Certificates: 2 years
AIA Locations: LDAP and HTTP
CDP Locations: LDAP and HTTP
CRL Validity: 1 week
Delta CRLs Validity: 2 hours
CA Name: Fabrikam Issuing CA 1

Pre-installation Phase

In this phase we will be building our configuration files. Since we are building a two tier hierarchy with one CA in each tier, we will need the following configuration files.

Root CA

CAPolicy.inf
Post-Installation Configuration Script

Issuing CA

CAPolicy.inf
Post-Installation Configuration Script

Root CA CAPolicy.inf

The CAPolicy.inf file is fairly straightforward. It is simply an .INF file consisting of named sections containing keys associated with some data. The only required section is [Version]. Then we have the [Certsrv_Server] section which specifies certain CA settings, most of which only apply when renewing the CA certificate. The initial CA configuration will be defined when the ADCS role is installed, but in the future the settings specified in the CAPolicy.inf file will be used. We also specify an empty CDP and AIA location by listing the [CRLDistributionPoint] and [AuthorityInformationAccess] sections with no values. These empty CDP and AIA values are not required in Windows Server 2008 since the installation of a Root CA will omit these extensions by default. However, in Windows Server 2003 you would have to specify these sections in order to omit these extensions from your Root CA Certificate.
Root CA Post-Installation Configuration Script

The Post-Installation script I am using is based on the one provided in the Best Practices for Implementing a Microsoft Windows Server 2003 Public Key Infrastructure. If you are looking to deploy a PKI this is a must read, and goes in to greater depth than this blog series. Below is the Post-Installation configurations script that I will be using:

You will have to modify line 1 of the script to reflect your forest root domain name. For our environment the line is modified to:

```
SET myADnamingcontext=DC=fabrikam,DC=com
```

In line 2 we set the DSConfigDN setting, which is used by the CA to build the LDAP paths for the AIA and CDP location.

In lines 3-6 we configure the CRL publication options.

Before we look at this section of the script, we first need to understand the options, variables, and structure of this command. So, clearly the command `certutil -setreg CA\CRLPublicationURLs` is going to set locations for publishing CRLs.

The format is `<publication option>[:<publication URL>]`. Multiple publication locations are separated by `\n`.

The Publication Option includes a number that maps to the options that are configured in the GUI. For CDP there are the following options as outlined in the table below. Each option is represented by a number, and the list of all options selected is represented by the combined total of their option numbers. This total value is what is placed in the CRLPublicationURLs publication options value.

<table>
<thead>
<tr>
<th>Option Number</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No Options Defined</td>
</tr>
<tr>
<td>1</td>
<td>Publish CRLs to this location</td>
</tr>
<tr>
<td>2</td>
<td>Include in the CDP extensions of issued certificates</td>
</tr>
<tr>
<td>4</td>
<td>Include in CRLs. Clients use this to find Delta CRL Locations</td>
</tr>
<tr>
<td>8</td>
<td>Include in all CRLs. Specifies where to publish in the Active Directory when publishing manually.</td>
</tr>
<tr>
<td>64</td>
<td>Publish Delta CRLs to this location</td>
</tr>
<tr>
<td>128</td>
<td>Include in the IDP extension of issued CRLs</td>
</tr>
</tbody>
</table>

There are also variables that can be included in the publication location. Variables are preferred over hard coded values, due to some issues that can arise with hard coded values. For example, customers that hardcode these URLs frequently leave off the variable that represents the index value of the CA certificate’s private key (%4). As a result, and after a CA certificate is renewed, publishing the CRL
signed by the second private key overwrites the CRL signed by the first private key because the index that would differentiate the filename has been omitted.

Below is a table that explains these variables:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Variable Display Name in the CertSrv MMC</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%1</td>
<td>&lt;ServerDNSName&gt;</td>
<td>The DNS name of the certification authority server</td>
</tr>
<tr>
<td>%2</td>
<td>&lt;ServerShortName&gt;</td>
<td>The NetBIOS name of the certification authority server</td>
</tr>
<tr>
<td>%3</td>
<td>&lt;CaName&gt;</td>
<td>The name of the Certificate Authority</td>
</tr>
<tr>
<td>%4</td>
<td>&lt;CertificateName&gt;</td>
<td>The renewal extension of the certification authority</td>
</tr>
<tr>
<td>%6</td>
<td>&lt;ConfigurationContainer&gt;</td>
<td>The location of the Configuration container in Active Directory</td>
</tr>
<tr>
<td>%7</td>
<td>&lt;CATruncatedName&gt;</td>
<td>The &quot;sanitized&quot; name of the certification authority, truncated to 32 characters with a hash on the end</td>
</tr>
<tr>
<td>%8</td>
<td>&lt;CRLNameSuffix&gt;</td>
<td>Inserts a name suffix at the end of the file name when publishing a CRL to a file or URL location</td>
</tr>
<tr>
<td>%9</td>
<td>&lt;DeltaCRLAllowed&gt;</td>
<td>When a delta CRL is published, this replaces the CRLNameSuffix with a separate suffix to distinguish the delta CRL</td>
</tr>
<tr>
<td>%10</td>
<td>&lt;CDPObjectClass&gt;</td>
<td>The object class identifier for CRL distribution points, used when publishing to an LDAP URL</td>
</tr>
<tr>
<td>%11</td>
<td>&lt;CAObjectClass&gt;</td>
<td>The object class identifier for a certification authority, used when publishing to an LDAP URL</td>
</tr>
</tbody>
</table>

*Descriptions were taken from the following online document: Specify certificate revocation list distribution points in issued certificates, which is located at [http://technet.microsoft.com/en-us/library/cc773036(WS.10).aspx](http://technet.microsoft.com/en-us/library/cc773036(WS.10).aspx)*

So for example from line 4 of the configuration script we have the following:

```
1:%WINDIR%\system32\CertSrv\CertEnroll\%3%%8%%9.crl\n```

This translates to the following:

1. Publish CRLs to %WINDIR%\System32\CertSrv\CertEnroll\ directory.

2. The filename of the CRL will be the name of the CA + a CRL name suffix + a "+" character if the file represents a CRL.

3. Note that the only publication option set by this string instructs the CA to write the CRL to the specified location. This location will not appear in the actual certificates issued by the CA because that option was not specified.

Understanding the variables and options not only allows you to understand what the script does, but also how to modify the script. It also allows you to understand the configuration when reviewing the Certificate Authorities configuration in the registry.

In my example, I am happy with the default settings, so I am going to leave them in the default.

On lines 7 through 9 of the script we define publication options for the CA certificate, commonly referred to as Authority Information Access.
In interpreting the variables in the script you can use the Variable Table we used previously to decode the script or to encode your desired setting. However, since we will not be using any of the CRL related variables, only a subset of the options are available when working with AIA, they are outlined in the table below:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Variable Display Name in the CertSrv MMC</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%1</td>
<td>&lt;ServerDNSName&gt;</td>
<td>The DNS name of the certification authority server</td>
</tr>
<tr>
<td>%2</td>
<td>&lt;ServerShortName&gt;</td>
<td>The NetBIOS name of the certification authority server</td>
</tr>
<tr>
<td>%3</td>
<td>&lt;CaName&gt;</td>
<td>The name of the Certificate Authority</td>
</tr>
<tr>
<td>%4</td>
<td>&lt;CertificateName&gt;</td>
<td>The renewal extension of the certification authority</td>
</tr>
<tr>
<td>%6</td>
<td>&lt;ConfigurationContainer&gt;</td>
<td>The location of the Configuration container in Active Directory</td>
</tr>
<tr>
<td>%7</td>
<td>&lt;CATruncatedName&gt;</td>
<td>The &quot;sanitized&quot; name of the certification authority, truncated to 32 characters with a hash on the end</td>
</tr>
<tr>
<td>%11</td>
<td>&lt;CAObjectClass&gt;</td>
<td>The object class identifier for a certification authority, used when publishing to an LDAP URL</td>
</tr>
</tbody>
</table>

When setting AIA publication, a different, smaller set of Options is available:

<table>
<thead>
<tr>
<th>Option Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Not Defined</td>
</tr>
<tr>
<td>1</td>
<td>Automatically Publish CA Certificate to this location</td>
</tr>
<tr>
<td>2</td>
<td>Include in the AIA extension of issued certificates</td>
</tr>
<tr>
<td>32</td>
<td>Include in the online certificate status protocol (OCSP) extension</td>
</tr>
</tbody>
</table>

As an example on line 8 we have the following:

1: %WINDIR%\system32\CertSrv\CertEnroll\%%1_%%3%%4.crt

This translates to:

1. Publish the CA Certificate to %WINDIR%\system32\CertSrv\CertEnroll

2. The file name of the CA certificate will be <ServerDNSName>_.<CaName> with the renewal extension of the Certificate Authority.

3. As with the CRL, the option specified indicates only that the CA should write the CA certificate to this location. This location will not be included in the AIA extension of certificates issued by this CA since that option was not specified.

Next, on lines 10 and 11 we have the following:

certutil -setreg CA\CRLPeriodUnits 180

These set the CRL Validity and Publishing period 180 days or roughly six months.

Then on line 12 we have the following:
certutil -setreg CA\CRLDeltaPeriodUnits 0

This means that Delta CRLs will not be published because the publication period is zero.

Next, on lines 13 and 14 we have the following:

certutil -setreg ca\ValidityPeriodUnits 10
certutil -setreg ca\ValidityPeriod "Years"

These settings define the maximum validity period for certificates issued by the CA. Since this is a Root CA and only Subordinate CA certificates are going to be issued, this would be the validity period for the Subordinate CA certificate. In my case I only want my Subordinate CA to be valid for 5 years so I am going to change the entry in line 13 to reflect that, with the following change:

certutil -setreg ca\ValidityPeriodUnits 5

Then on line 15 we stop and start certificate services. We do that so that the new configuration can be read by the certificate authority:

net stop certsvc & net start certsvc

On line 16 we ensure that the CertEnroll share is created, and if IIS is installed, that the CertSrv virtual directory is installed.

certutil -vroot

Finally, on line 17 we publish a new CRL:

certutil -CRL

So now we are done with the post-installation configuration script. The reason for understanding this script is so that you validate the values in the sample script are the ones you want implemented on your CA. If not, you can of course modify the sample script or create your own. The reason for using the sample script on the Root CA is that you validate all of the settings prior to implementing them. Also, if you are going to test your installation in a test environment, you can run this script and verify you get the results you are expecting. This will help eliminate any surprises you may have when implementing the Root CA in production.

Issuing CA CAPolicy.inf

The CAPolicy.inf file for my Issuing CA is fairly straightforward. You can actually configure quite a bit with these configuration files. Jonathan Stephens will be posting a blog on all of the settings that can be configured with the CAPolicy.inf file, so stay tuned. However, the PKI I am setting up is pretty straightforward, so I am only going to specify the Renewal Key Length, Renewal Validity, and direct the CA to not load the default certificate templates. The reason for not deploying the default certificate templates is that machines and users will be able to begin enrolling for the default templates if they have access once the installation of the CA is complete. In my case I want to make sure that I setup my templates to meet my business requirements before users and machines can enroll for certificates. The LoadDefaultTemplates setting works on Windows 2003 Root Enterprise CAs, Windows 2008 Root CAs, and Windows 2008 Subordinate CAs.

[Version]
Signature= "$Windows NT$"
[Certsrv_Server]
RenewalKeyLength=4096
RenewalValidityPeriod=Years
RenewalValidityPeriodUnits=5
LoadDefaultTemplates=0

Issuing CA Post-Installation Configuration Script

The post-installation script I am using is based on the one in Best Practices for Implementing a Microsoft Windows Server 2003 Public Key Infrastructure. I am not going to go in as much detail for this script, since I covered all of the settings for the Root CA. Below is the script I will be using:

certutil -setreg CA\CRLPublicationURLs "65:%WINDIR\system32\CertSvc\CertEnroll\%%3\%%8\%%9.crl\n6:http://FCCA01.FOURTHCOFFEE.COM/certenrol\n79:ldap:///CN=%%7\%%8,CN=%%2,CN=CDP,CN=Public Key Services,CN=Services,%%6\%%10"

certutil -setreg CA\CACertPublicationURLs "1:%WINDIR\system32\CertSvc\CertEnroll\%%1_%%3\%%4.crt\n2:http://FCCA01.FOURTHCOFFEE.COM/certenrol\n3:ldap:///CN=%%7,CN=AIA,CN=Public Key Services,CN=Services,%%6\%%11"

certutil -setreg CA\CRLPeriodUnits 7

certutil -setreg CA\CRLPeriod "Days"

certutil -setreg CA\DeltaCRLPeriodUnits 2

certutil -setreg CA\DeltaCRLPeriod "Hours"

net stop certsvc & net start certsvc

certutil.exe -vroot

certutil -CRL

Lines 1-8 of the script will Configure CRL and AIA publication. You can use the tables in the previous section to determine if these are configured to meet your requirements. After examining the CDP and AIA publishing, the settings in the script happen to match the requirements of my environment.

In lines 9 and 10 we configure the validity period and publication interval for the Base CRL. In my environment, my requirement is publication once a week. So, I will change line 10 to reflect this (Since this setting is used in combination with the certutil -setreg CA\CRLPeriod "Days" the resulting period will be seven days):

certutil -setreg CA\CRLPeriodUnits 7

After line 10 I am going to add two additional lines to configure my delta CRL publication so that a new delta CRL is published every two hours:

Certutil -setreg CA\DeltaCRLPeriodUnits 2
Certutil -setreg CA\DeltaCRLPeriod “Hours”

PKI Installation Walkthrough

Root CA Install
1. Copy CAPolicy.inf to the c:\Windows directory.

2. Start Server Manager and select Roles. In the Roles section click on Add Roles.

3. This will start the Add Roles Wizard, click Next.


6. On the Select Role Services page, select Certification Authority, and click Next.

7. On the Specify Setup Type page, make sure Standalone is selected, and click Next.

8. On the Specify CA Type, select Root CA, and click Next.

9. On the Set Up Private Key page of the wizard, and make sure Create a new private key is selected and click Next.

10. On the Configure Cryptography for CA, you can select the CSP or KSP you wish to use, the Key Length for the Root Certification Authorities key, and the hash algorithm. Due to the business requirements of Fabrikam, I will be using the RSA Microsoft Software Key Storage Provider KSP, a key length of 4096 bits, and the SHA1 algorithm. (Check to determine supportability of stronger hash algorithms.) When finished click Next.

11. On the Configure CA Name page enter the Common Name for the CA and the name suffix.

12. When setting up the PKI for Fabrikam we are going to use Fabrikam Root Certification Authority for the Common Name and O=Fabrikam,C=US for the name suffix.

13. When complete, click Next.

14. On the Set Validity Period of the wizard, select the validity for the Root CA Certificate. As specified at the beginning of this post, Fabrikam is using a validity period of 10 years for the Root CA. Once complete, click Next to continue the wizard.

15. On the Configure Certificate Database page, select the locations for the Certificate Services Database and logs. Fabrikam will be using the default file paths. Select Next, when complete.

16. The Confirm Installation Selection page will summarize your selections. Review this information and then click Install.

17. Once setup is complete, run the post-installation configuration script. Review the settings to determine they are set to your specifications.

18. Copy the CRL and CRT files from c:\windows\system32\certsrv\certenroll to a Floppy or other removable media.

19. Logon to the machine that will become the Enterprise CA. Attach the removable media. Run the following command to publish the Root CA Certificate to Active Directory: certutil -f –dpublish <CaCertFileName> RootCA.

20. Then publish the CRL from the Root CA to Active Directory with the following command: certutil -f –dpublish <CrlFileName>.

21. Wait sufficient time for replication to complete within your site.
22. Run `certutil -pulse`. Running this command will trigger autoenrollment and the Root CAs Certificate and CRL will be downloaded automatically to the Trusted Root Certification Authority store on the local machine.

### Issuing CA Installation

1. Copy the CAPolicy.inf file to the `c:\Windows` directory.

2. Start Server Manager and select Roles. In the Roles section click on Add Roles.

3. This will start the Add Roles Wizard, click Next.


6. Part of the business requirements of Fabrikam is to have Web Enrollment available. As part of the plan we decided to install Web Enrollment on the Issuing CA, and in this way we could use it as the HTTP repository for CRLs and CA Certificates as well. If you wish to use the same approach, on the Select Role Services page, select Certification Authority and Certification Authority Web Enrollment, and click Next.

7. You will then be prompted to add the additional IIS role services required to run Web Enrollment. Click on Add Required Role Services.

8. On the Specify Setup Type page, select Enterprise, and click Next. Note: If you see the Enterprise option grayed out, that is an indication that the computer is not joined to the domain or you are not logged in with Enterprise Admin credentials.

9. On the Specify CA Type page, select Subordinate CA, and click Next.

10. On the Setup Private Key page, select Create a new private key, and click Next.

11. On the Configure Cryptography for CA, you can select the CSP or KSP you wish to use, the Key Length for the Issuing Certification Authorities key, and the hash algorithm. Due to the business requirements of Fabrikam, I will be using the RSA Microsoft Software Key Storage Provider KSP, a key length of 2048 bits, and the SHA1 algorithm. (Check to determine supportability of stronger hash algorithms.) When finished click Next.

12. On the Configure CA Name, enter the Common Name for the Issuing CA. The distinguished name should already be populated with the DN of the domain in which you are installing the CA. Fabrikam has the Common name set as "Fabrikam Issuing Certification Authority", and the DN of course is set to DC=Fabrikam,DC=com. When finished click Next.

13. On the Request Certificate from a Parent CA, select Save a certificate request to file and manually send it later to a parent CA. Then click the Browse button, navigate to Removable Media, enter a file name for the request, and click Save. Then click Next.

14. On the Configure Certificate Database page, select the locations for the Certificate Services Database and logs. Fabrikam will be using the default file paths. Select Next, when complete.

15. On the Web Server (IIS) page, click Next.

16. On the Select Role Services page of the wizard, click Next.
17. On the **Confirm Installation Selections**, select **Install**.

18. You will then be prompted with the **Installation Results**. The results will indicate that installation is not complete because you still need to submit the request to the Root CA and then install the resulting certificate. Click Close to acknowledge the results.

19. Take the removable media that you saved the request to and connect it to the Root CA. In the Certificate Services MMC (certsrv.msc) on the Root CA, select the root node (CA Name), right click, then select **All Tasks**, then **Submit new request**..., from the context menus.

20. Browse to the request file and then select **Open**.

21. The request will now be pending. Navigate to the **Pending Request** Folder and locate the request. Right click on the request, select **All Tasks**, and then **Issue**.

22. Navigate to the **Issued Certificates** folder. Locate the certificate for the Issuing CA, Right click on the certificate and select **Open**. This will open the Certificate Properties. Select the **Details** tab, then click on the **Copy to File**... button.

23. This will open the **Certificate Export Wizard**, click **Next**.

24. Select DER encoded binary X.509 (.CER) and click **Next**.

25. Click **Browse**...

26. Navigate to Removable Media, enter a name for the certificate and then click **Save**.

27. Then click **Next**.

28. Then click **Finish**.

29. You will be prompted that **The export was successful**. Click **OK** to acknowledge.

30. Open the Certification Authority MMC on the Issuing CA. Right click on the Root Node (CA Name), select **All Tasks**, and **Start Service**.

31. You will then be asked if you would like to install the CA Certificate, click **Yes**.

32. Insert the removable media that contains the CA Certificate for the issuing CA. Browse to the certificate, and select **Open**.

33. Next run the Post Installation Configuration Script and review the configuration to ensure that it meets your requirements.

**Conclusion**

In this installment we covered the requirements for a sample PKI deployment. We covered configuring the CAPolicy.inf and Post Configuration script. In Part III of this series I will cover a sample configuration and setup of Certificate Templates.

- Chris ”Duuuuude” Delay

```powershell
SET myADnamingcontext=DC=fourthcoffee,DC=com
certutil.exe -setreg ca\DSConfigDN "CN=Configuration,%myADnamingcontext%"
```
certutil -setreg CA\CRLPublicationURLs
"1:%WINDIR%\system32\CertSvc\CertEnroll\%3\8\9.crl\n2:http://FCCA01.fourthcoffee.com/certenroll/%3\8\9.crl\n10:ldap://CN=%7,CN=%2,CN=CDP,CN=Public Key Services,CN=Services,%6%10"

certutil -setreg CA\CACertPublicationURLs
"1:%WINDIR%\system32\CertSvc\CertEnroll\%1_3\4.crt\n2:http://FCCA01.fourthcoffee.com/certenroll/%1_3\4.crt\n2:ldap://CN=%7,CN=AIA,CN=Public Key Services,CN=Services,%6%11"

certutil -setreg CA\CRLPeriodUnits 180

certutil -setreg CA\CRLPeriod "Days"

certutil -setreg CA\CRLDeltaPeriodUnits 0

certutil -setreg ca\ValidityPeriodUnits 5

certutil -setreg ca\ValidityPeriod "Years"

net stop certsvc & net start certsvc

certutil -vroot

certutil -CRL

---

**Part III Certificate Templates**

Chris here again. In this segment I will be covering setting up certificate templates on the newly created CA hierarchy. Enterprise Certification Authorities (CAs), as well as clients, utilize what are called certificate templates. Certificate templates contain properties that would be common to all certificates issued by the CA based on that template. Windows includes several predefined templates, but Administrators also have the ability to create their own templates specific for their enterprise. When requesting a certificate, a client can just specify the template name in the request and the CA will build the certificate based upon the requestor’s information in Active Directory and the properties defined in the template.

Certificate templates are also used to define the enrollment policy on the CA. First, an Enterprise CA can only issue certificates based upon the templates it is configure to use. For example, if the CorpUserEmail template is not available on the CA then the CA cannot issue certificates based on that template. Second, permissions set on the certificate template’s Active Directory object determine whether or not a user or computer is permitted to request a certificate based on that template. If a user does not have Enroll permissions on a particular template, the CA will deny any request submitted by the user for a certificate based on that template.

As the Windows Server operating system has evolved over the last ten years, so has the concept of the certificate template. Currently, there are three versions of templates:

Version 1 templates were introduced in Windows 2000, and can be used by Windows 2000, Windows Server 2003 (R2), and Windows Server 2008 (R2) Enterprise CAs. Version 1 templates Active Directory objects are created the first time an Enterprise CA is created in the forest. These templates were designed to reflect the most common scenarios for digital certificates in the Enterprise. Unfortunately, if you don’t like the settings we selected you’re pretty much out of luck. Creating new v1 templates, or editing the existing templates, is not supported. The only customization supported is to the permissions on the template.
Version 2 templates were introduced in Windows Server 2003 and are a vast improvement over v1 templates. First and foremost, v2 templates can be modified by an Enterprise Admin. In addition, the Admin can duplicate an existing v1 or v2 template to create a new v2 template, and then customize the result. Finally, v2 templates expose a larger number of properties that can be configured, and also expose some controls to take advantage of some other new features introduced in Windows Server 2003. One of these features, for example, is key archival. Version 2 templates can be used by Windows Server 2003 and Windows Server 2008 Enterprise or Datacenter Editions. On Windows Server 2008 R2, v2 templates can be used by a CA installed on Standard, Enterprise, Datacenter, Foundation and Server Core Editions.

Version 3 templates were introduced in Windows Server 2008. Version 3 templates have all the features of a version 2 template with two major additions. First, v3 templates support the use of Crypto Next Generation (CNG) providers, which means that the certificates support Suite B algorithms based on Elliptical Curve Cryptography (ECC). Second, v3 templates have a setting that instructs Windows to grant the Network Service account access to the private key created on the requesting computer. This is great for those certificates that will be used by applications or services that run as Network Service rather than Local System. Version 3 templates are supported by CAs installed on Windows Server 2008 Enterprise and Datacenter Editions. They are also supported by CAs installed on Windows Server 2008 R2 Standard, Enterprise, Datacenter, Foundation and Server Core Editions.

For a complete table of Windows Server SKU and the features supported by it, check out this blog post by the Windows PKI development team.

Deploying Certificate Templates

For the purpose of example, I am going to use a fictional company called Fabrikam. The diligent IT staff at Fabrikam have done their research, performed some testing, consulted the auguries, and they've determined what types of certificates they need to issue to meet the specified business needs. The next step is to look at what templates are available that they can use out of the box and which ones they need to modify to suit their purposes.

Here's a quick overview of what Fabrikam determined:

**CA Issuance:** Domain Controller Authentication, Web Server, and User certificates  
**Key Archival:** The private keys for User certificates should be archived  
**Domain Controller Authentication template:** No additional requirements  
**Web Server template:**
- A version 2 template must be created from the default Web Server template.  
- The security group *Fabrikam Web Servers* should have Enroll permissions.  
- The Subject name must contain the DNS name of the web server, and should be provided automatically.

**User Certificate Template:**
- A version 2 template must be created from the default User template  
- Key Archival must be implemented for the template

**Certificate Templates Setup**

Fabrikam has decided that they need to deploy the following certificate templates: Domain Controller Authentication, Web Server, and User. In addition, the fact that Key Archival is to be enabled for the User template means that the CA should also be configured to issue certificates based on the Key Recovery Agent template (Actually, this is not a requirement if there is another Windows Enterprise CA in environment that is configured to issue Key Recovery Agent certificates, and is trusted to do so.)
Let’s assume that the PKI hierarchy has been set up and is functional. The next step is to configure the certificate templates. Let’s check the configuration of the templates before deploying them.

To manage the certificate templates, you use the Certificate Templates MMC snap-in. In the Certificate Services MMC snap-in, right-click on the Certificate Templates folder and select **Manage** from the context menu.

In the view pane of the Certificate Templates snap-in you’ll see all the certificate templates available in Active Directory. If you locate the Domain Controller Authentication template and double-click on it, you’ll see the properties available for that template. Our fictional IT staff has already reviewed the settings and determined that no changes need to be made, so we’ll just click Cancel, here.

Next, locate the Web Server template. The default Web Server template already meets the current requirements that arose from an analysis of business needs. However, to allow for future changes Fabrikam has decided that they need to duplicate this default template and create a v2 template.
To duplicate the existing Web Server template and create Version 2 template:

1. I right click on the Web Server template and select **Duplicate Template** from the context menu.

Fabrikam still has a lot of Windows Server 2003 servers and Windows XP workstations (But they are steadily upgrading. No, really! They are!! Trust me! Sigh.) This means that we can't use the latest and greatest v3 templates available on our Windows Server 2008 CA. We'll have to specify that we’re creating a template for **Windows 2003 Server, Enterprise Edition** which will create a v2 certificate template.
2. We’ll give a new name to the template: Fabrikam WebServer.

3. Clients within Fabrikam will connect to the web servers via the server’s DNS name. This means that the requesting server’s fully qualified DNS name must be in the Subject of the certificate it receives. To meet this requirement, click on the **Subject Name** tab and select **Build from this Active Directory** information. For the **Subject Name Format**, select **DNS Name**. Finally, deselect all of the
check boxes under **Include this information in the alternate Subject name.**

Now that the new template is configured per the specified requirements, we need to set the security. The computer account for a particular web server will be the principal enrolling for the **Fabrikam WebServer** template, so we have to make sure that all the web server computer accounts have **Enroll** permission on the new template. Fabrikam, luckily, has a Security Group containing all of their web servers called, oddly enough, **Fabrikam Web Servers**. We can simply grant the necessary permissions to that group.

1. In the template properties, elect the **Security** tab, and click **Add...**
2. Enter the group name (**Fabrikam Web Servers**) and click the **Check Names** button.
3. After the name of the security group is resolved, click **OK**.
4. Grant the group **Enroll** permission.

The permissions in the security tab should like this when these changes are complete.
Once all the necessary changes have been made, click Ok to commit the new template and save it to Active Directory. The Fabrikam WebServer template is now ready to be added to the CA.

**User Certificate Template**

We’ll use essentially the same process to duplicate the default User template and modify the resulting v2 template to suit Fabrikam’s requirements.

Just as with the default WebServer, we’ll duplicate the existing User template to create the custom v2 template. We need to do this because the default User template is a v1 template, so its properties cannot be modified. One of our requirements is to enable Key Archival which requires configuring a setting in the template, so in order to do this a v2 template is required.

To create and configure our new User template:

1. Select the User template, right click on it, and select Duplicate Template from the context menu.

![Duplicate Template]

3. Change the Template Display name to **Fabrikam User**.

![Properties of New Template]

4. Navigate to the Request Handling Tab, and select **Archive subject’s encryption private key** to enable key archival for this template.
5. Next, set permissions on the new template. Domain Users will already have Enroll permission, but since this certificate will be deployed via user Autoenrollment, Domain Users will also require Autoenroll permission. The permissions, when set properly, should look like this:
Once all the necessary changes have been made, click Ok to commit the new template and save it to Active Directory. The Fabrikam User template is now ready to be added to the CA.

**Key Recovery Agent Certificate Template**

Although this template was not mentioned as one of Fabrikam’s requirements, it is a requirement to issue at least one Key Recovery Agent certificate to support Key Archival. This step is only necessary if there is not another Windows Enterprise CA configured to issue certificates based on the Key Recovery Agent template.

For this example, however, let’s assume that there is not and go ahead and configure the Key Recovery Agent template. The only setting that requires modification is the permissions. We’ll assign enroll permissions to the Fabrikam KRA security group so that members of that group can enroll for a Key Recovery Agent certificate.

1. Open up the **Key Recovery Agent** certificate template by double-clicking on it and selecting the **Security** tab. I click **Add...**
2. Enter the name **Fabrikam KRA** and click the **Check Names** button.
3. After the name of the security group is resolved, click OK.
4. Check the **Enroll** permission.

**Configuring the CA to issue certificates**

To configure the CA to issue the desired certificate templates, I right-click on the **Certificate Templates** folder, select **New**, then select **Certificate Templates to Issue** from the context menu.
Then I select the certificate templates I wish to issue, by holding down the control key and selecting multiple templates, and then clicking OK.
This CA can now issue certificates based on the selected certificated templates.

Conclusion

That's really all there is to it. While in this segment we only modified a few properties of our templates, in the vast majority of cases there should be no need for making extreme changes. The default templates should be sufficient for most implementations, and the changes we made were more to ease certificate deployment than actually create truly custom templates. Perhaps in a later blog post we’ll cover some of the more esoteric settings. However, this shouldn’t stop you from exploring on your own using the online help.

In Part IV of this series we’ll cover implementing Web Enrollment and Key Archival.

Part IV Configuring SSL for Web Enrollment and Enabling Key Archival
Chris here again. Today we are going to cover configuring SSL for the Web Enrollment website which will allow Windows Server 2008 and Windows clients to use the Web Enrollment website. We are also going to cover enabling Key Archival.

Configuring SSL for Web Enrollment

Windows Server 2008 (R2) requires SSL in order to connect to the Web Enrollment pages. The first thing that must be done after installing the Web Enrollment role is to enable SSL on the web site within IIS. To begin, I am going to go through the process to request an SSL certificate for my web server.

Requesting an SSL Certificate

In Part III I covered implementing certificate templates. The fictional company, Fabrikam, created a customized template for web servers called Fabrikam WebServer. This template was configured to construct the certificate subject information from Active Directory. When a server requests a Fabrikam Webserver certificate from the CA, the CA will place the DNS name of the server in the Subject of the issued certificate. Below are the steps to follow in order to request a certificate based on the Fabrikam WebServer template.

1. Click Start button, then Run, and enter MMC in the Run box and click OK.

2. Click on File from the menu bar and select Add/Remove Snap-in...

3. Select Certificates and click the Add button

4. When prompted for the context that the Certificates MMC should run in select Computer Account, and then click Next, then Finish.

5. Click OK, to close the Add or Remove Snap-ins page.

6. Expand Certificates (Local Computer), right-click on Personal, and select Request New Certificate... from the context menu.

7. This starts the Certificate Enrollment wizard. Click Next to continue.

8. Select the Fabrikam WebServer certificate template, and then click Next to request the certificate.

9. As seen below, the certificate has been successfully requested. Click Finish to close the wizard.
**Enabling SSL**

Now that the certificate has been requested, the next step is to bind the certificate to the default web site in IIS.

To enable SSL for the Web Enrollment site on the CA server:

1. Launch the IIS Manager MMC located in Administrative Tools.

2. Expand the server name, then Sites, and then select **Default Web Site**.

3. In the **Actions** menu, select **Bindings...**

4. The **Site Bindings** settings will open. Click **Add...**
5. Select **https** for **Type**, and select the appropriate certificate from the **SSL certificate** drop down. Review the settings, and click **OK**.

![Add Site Binding](image1)

6. Click Close to commit the changes to IIS. The selected server authentication certificate is now bound to port 443 on the IIS server.

![Site Bindings](image2)

The Web Enrollment website is now configured to support HTTP over SSL connections via the fully qualified domain name. Since the site is accessed via FQDN, the server, in this example https://fabca01.fabrikam.com, must be added to the list of trusted sites in Internet Explorer of clients that will attempt to access this page. This is so that so that user credentials are automatically passed to the Web Enrollment site. For domain clients, this can be done via **Group Policy** (see *Site to Zone Assignment List* policy).

**Key Archival**

Next, we'll look at setting up **Key Archival**. There are two parts for setting up Key Archival. The first is designating a Key Recovery Agent for the CA. The second is configuring the Certificate Template for archival which we touched on in the previous part, Configuring Certificate Templates.

Key Archival is important for certificates that are used for encryption. If a user's encryption private key is lost for some reason, any encrypted data can be recovered by extracting the archived private key from the CA database and returned to the user.

**Designating a Key Recover Agent (KRA)**

In the previous part, we configured the CA to issue KRA certificates. Specifically, we added the default KRA template to the list of certificate templates available on the CA, and set the permission on template to allow members of the Fabrikam KRA security group enroll. The next step is to have at least one member of that group request the KRA certificate.
The user, Magnus Hedlund, is a member of the Fabrikam KRA group. Here’s how he’d request a KRA certificate.

1. Connect to the Web Enrollment site and select **Request a certificate** from the Web Enrollment webpage.

2. Select **Create and submit a request to this CA**.

3. From the Certificate Template drop-down, select Key Recovery Agent. Magnus should also make sure the option **Mark keys as exportable** is selected, but the rest of the default settings can be accepted.

4. Set the friendly name to **KRA Cert**, and click **Submit**.

5. The default Key Recovery Agent template is configured to require Certificate Manager approval (on the Request Handling tab), meaning that a Certificate Manager (a local Administrator on the server, by default) must manually issue the certificate. As such, Magnus will see a message stating that his request is in a pending state. Magnus then emails the Certificate Manager to get the request approved.

6. The Certificate Manager launches the Certificate Services MMC, selects **Pending Requests**, and locates Magnus’ request. She then right-clicks on the request and selects **All Tasks** from the context menu and then selects **Issue**.

7. In order to retrieve his issued certificate, Magnus must returns to the Web Enrollment site. This time he must select **View the status of a pending certificate request**. REQUIRED: Magnus must reconnect to the site using the same client he used to submit the request. The Web Enrollment pages use a cookie to record pending requests.

8. Magnus then clicks on his request that is identified by the date and time that it was submitted.

9. Magnus then has a link to **Install this certificate**.

10. Magnus is then presented with a **Potential Scripting Violation** error asking if the certificate should be added to the certificate store. He clicks **Yes** to acknowledge the warning.
The Certificate is then successfully installed.

User’s with Key Recovery Agent certificates should take care to protect their certificates and keys. One way of doing that is exporting the KRA certificate and private key to a PFX file – deleting the private key stored on the client – and keeping that password protected file in a safe location. The KRA certificate and private key can then be imported as needed.

To do this, Magnus follows these steps:

1. Open the Certificates MMC targeted to his user account (Certmgr.msc).
2. Expand Personal, then Certificates. Locate the KRA Cert, and right-click on it. Select Export from the context menu.
3. This launches the Certificate Export Wizard. Click Next to continue.
4. On the Export Private Key page of the wizard, select Yes, export the private key, and click Next.
5. On the Export File Format page, select Delete the private key if the export is successful and make sure all the other options are deselected. Click Next.
6. Enter a password to secure the Private Key, and click Next.
7. On the File to Export page, click Browse....
8. Browse to a secure location in the file system, give the PFX file a name, and click Save.
9. Click Next on the File to Export page.
10. Click Finish to complete the export.
11. He is then prompted that the export was successful, and clicks OK.

In a high security setting, one option may be to save the PFX file to removable media, and then secure that media in a locked safe until it is needed. Why are such measures necessary? Well, they may not be; it totally depends on your environment. What is important to realize is that Key Recover Agents can decrypt the encrypted key blob for any user with an archived key. The CA’s design tries to mitigate this risk somewhat by requiring a Certificate Manager to actually export the encrypted key blob from the CA database. A Key Recovery Agent can only decrypted the exported key blob, but he can’t actually export the key blob.
Enabling Key Archival

Now that we actually have a Key Recovery Agent published in Active Directory (any KRA certificate issued by the CA is published to the CN=KRA container in Active Directory), we can proceed to enable Key Archival on the CA.

To enable Key Archival:

1. Open the Certificate Services MMC. Right-click on the name of the CA in the tree-view pane and then select Properties from the context menu.

2. Select the Recovery Agents tab, and select Archive the key. In this case, we only have one Key Recovery Agent, so we'll leave Number of recovery key agents to use at the default of 1. Click Add... to add the KRA certificate to the CA.
3. Select the appropriate KRA Certificate, and click **OK**.

![Key Recovery Agent Selection](image1)

4. Click **OK** to close the properties and commit the changes we've made to the CA.

![Fabricam Issuing Certification Authority Properties](image2)

5. When prompted to restart Certificate Services, click **Yes**.

![Certification Authority](image3)

Key archival is now enabled on the CA.
Additional information on the mechanics of Key Archival is available here: http://blogs.technet.com/pki/archive/2009/08/07/understanding-key-archival.aspx

Conclusion

And that’s it for this part of the series. Today, we configured the IIS server hosting our Web Enrollment pages to use SSL when serving the site. We also configured our CA with a Key Recovery Agent to enable Key Archival. Neither of these steps are required in order for the CA to issue certificates, but setting up the features properly will increase the usefulness of your PKI.

In the final segment in this series, I will cover Disaster Recovery Scenarios.

Part V Disaster Recovery

Chris here again. We are now going to move onto Disaster Recovery. One of the many tasks you want to complete during the planning phase is to plan for disaster recovery. When planning for disaster recovery not only is the backup/restore process important, but the actual design of the PKI can affect how resilient your PKI infrastructure is. Additionally, proper planning can alleviate the impact of a system failure.

When the system hosting Certificates Services becomes unusable due to a failure, there are a couple of consequences of that failure.

1. The CA can no longer sign its Certificate Revocation List (CRL) or delta CRL (dCRL)

2. The CA can no longer issue certificates.

3. The CA database includes a record of certificates that have been issue or revoked, and is unavailable until the CA is recovered.

Signing CRLs and Delta CRLs

CRLs and delta CRLs are used by clients to determine if a certificate has been revoked. In general, applications will fail when they cannot determine the revocation status for a certificate, though some applications have the ability to disable revocation checking while others do not.

Like certificates, CRLs and delta CRLs have a period during which they are valid. Once the CRL and/or delta CRL expires an application checking the revocation status of a certificate against the expired CRL will fail. The point of this discussion is that typically the first impact you will see when a Certification Authority fails is the inability of applications to check revocation status of any certificates.

When you design and implement a PKI you configure the validity period of the CA’s CRL and delta CRL. This design consideration has an impact in terms of disaster recovery. The maximum time you have after a CA failure to institute your recovery process without impacting certificate validation is determined by these settings.

Example 1. You have an issuing Certification Authority and it is publishing a base CRL once every 7 days and delta CRL once every day. You have approximately 24 hours since the last delta CRL was published to either restore the CA or re-sign the delta CRL before certificate validation starts failing.
Example 2. You have an issuing Certification Authority and it signs a CRL once every 7 days, but is not configured to publish a delta CRL. In this scenario you have 7 days – (the number of days since the base CRL was signed) before validation will begin to fail due to the inability to check revocation status against a valid CRL.

**Mitigation**

There are several ways that you can minimize the impact that a CA failure will have on certificate validation.

One way is to install a clustered issuing certification authority. If the active node of the cluster fails the CA can be failed over to the second node. Clustering, however, will not protect against the failure of a shared component such as storage or a Hardware Security Module (HSM). So these devices should have methods to provide failover as well, if possible.

Another option is to increase the period the base and delta CRL publication intervals (and hence, their validity periods). This can potentially give you more time to kick off your recovery process, but if the CA fails shortly before the new base or delta CRL is about to be published increasing the publication interval has done little good. One must also realize there is a trade-off involved here. Increasing the publication interval means that it will take longer for certificate consumers to become aware that a certificate has been revoked and added to the CRL.

A more complicated strategy is to set the automatic publishing interval to a longer period, and then manually publish the CRL more often. In other words you set the CRL publication interval to 7 days, and then publish a new CRL every day. This way, if the CA fails you have 6 or 7 days recognize the problem and start your recovery process. The Windows CA does not automatically publish CRLs in this fashion, but you can set up a scheduled task on the CA server to publish the CRL every 24 hours using the command line utility, certutil.exe. The command `certutil -crl` will instruct the CA to publish a new base CRL with the validity period defined in the CA configuration.

There are also some group policies that you can consider as part of your overall disaster recovery planning. If you have workstations and servers running Windows Vista, 7, Server 2008, or Server 2008 R2 there is a group policy setting that extends the period of time for which the OS will consider a given CRL valid, independent of the actual validity period of the CRL. The group policy setting is located in the following location:

Computer Configuration\Windows Settings\Security Settings\Public Key Policies\Certificate Path Validation Settings.

This setting forces the client to consider the CRL or OCSP response to be valid for longer than it actually is. Below is a screenshot of the specific settings:

![Allow CRL and OCSP responses to be valid longer than their lifetime (not recommended)](https://example.com/screenshot)

**Recovery**

In terms of recovery there is a short term workaround and a long term resolution. The short term workaround is to use a process called CRL re-signing to manually re-sign an existing CRL and extend its validity period. By doing this, you can give yourself additional time to recover the CA. CRL re-signing requires that you have a backup of the CA’s public/private key pair. I will be covering this process later in this blog posting.

The longer term fix is to restore the certification authority. This of course is not possible unless you have previously backed up the certification authority. I will also cover this later in the blog post.

**CA can no longer issue certificates**
Another issue that occurs when you have a CA failure is that it can no longer issue certificates. In some scenarios where certificates are issued less frequently, the inability to issue certificates may not have a business impact. In other cases, however, the impact could be considerable. For example, if a CA dedicated to issuing certificates for Network Access Protection (NAP) fails the problem would be almost immediately noticeable. NAP certificates have a lifetime of only 24 hours, so a failed CA can be a considerable problem.

Mitigation

One way to eliminate this issue completely is to have multiple CAs that are issuing certificates based on the same certificate templates. In this way, if one CA fails clients can still enroll for certificates on one of the other certificate authorities.

A clustered issuing certification authority is another way to mitigate against a failed CA. If one of the CAs in the cluster fails the cluster will fail over to the second node. Clustering, as mentioned earlier, will not protect against the failure of a shared component such as storage or an HSM. I’ll re-iterate the need for these devices to have methods for failover as well.

Recovery

Ultimately, recovering from the inability to issue certificates can be resolved by recovering the failed certification authority or installing a new issuing certification authority to issue certificates. The preferred method would be to restore the failed certification authority since it already has information about issued certificates in its CA Database.

CA Database

By default, the CA database contains a copy of every certificate issued, every certificate that has been revoked, and a copy of failed and pending requests. The CA Manager may decide, however, to clear out any expired certificates from the CA database in order to recover free space in the database.

Note: In Windows Server 2008 R2 you can configure a template such that issued certificates based on that template are not stored in the CA database. These so call "ephemeral certificates” generally have validity periods shorter than the publication interval of the issuing CA, so recording them so they can be later revoked makes little sense. Further, these short-lived certificates may be issued in great numbers and with great frequency. Storing them in the database can dramatically increase the database’s rate of growth. Certificates issued for NAP are examples of these ephemeral certificates.

If a CA is configured for key Archival and Recovery, the CA database will also contain the private keys for any certificates whose templates are configured for archival. Failure to recover the CA database in this case would result in losing all of these archived keys.

When a certificate authority fails the database is unavailable which makes it difficult to revoke certificates that were previously issued by the CA. It also makes it impossible to recover any certificates that have been archived in the database. Again, the database will be unavailable when the CA is unavailable. However, in rare circumstances it is possible that the CA database can become corrupted. Like all ESE databases, the CA database can be affected by hardware or disk issues that impact the database or log files.

Mitigation

One option to mitigate the database becoming unavailable due to a CA failure is to set up a clustered certification authority. Another option is to take regular backups of the CA. If the CA fails, you can then restore the CA from the backup. Below I discuss options for backing up the CA as well as for restoring the CA.

Recovery
For corrupt databases, repairs can be made with esentutil.exe. However, in most case it would be preferred to restore from a backup to avoid data loss that can be incurred when using some of the functions in esentutil.exe. Esentutil.exe can repair the structure of the database, but usually at the expense of the data stored within that structure.

**CA Backup**

**System State**

There are two different ways to backup the Certification Authority. The first is through a System State backup. A system state backup will back up the entire CA as well as its configuration. If the private key is stored on the CA and not on an HSM, the private key will be backed up as well. Here is additional information on System State. A system state backup should be used when you will need to restore to the same hardware.

**Backing up system state in Windows Server 2003.**

1. To start NT Backup, click **Start** then **Run**, type `ntbackup.exe` and press **Enter**.
2. If this is the first time you've run this tool, it will start the **Welcome to the Backup or Restore Wizard**.
3. Uncheck the **Always start in wizard mode**, and then click **Cancel**.
4. Launch NT Backup again.
5. Once NT Backup launches, select the **Backup** Tab, and check just **System State** as the item to backup.
6. Under the **Backup media or file name** section, select your backup media or file location where you wish to save the backup.
7. Click the **Start Backup** button. This will bring up the **Backup Job Information** dialogue box.
8. If you wish to start the backup immediately, click **Start Backup**.
9. If you wish to schedule the backup, click the **Schedule** button.
10. When prompted **You must save the backup selections before you can schedule a backup. Do you want to save your current selections now?**, click **Yes**.
11. Save the selection script.
12. After you save the selection script, the Scheduled Job Options dialogue box will open. Give the Job a name. Then click the **Properties** button.
13. Configure the desired schedule, and click **OK**. Then enter the credentials for the user that you wish the backup to run under. This account will need to either have **Backup files and directories** right or be a member of the **Backup Operators** group on the CA. Then click OK again. Click OK again, you will be prompted for the credentials again.
14. You can then click on the Schedule Jobs tab in NT Backup to check the schedule.

**Restore System State in Windows Server 2003**

1. On the Windows Server 2003 system on which you plan on restoring system state, open the NT Backup utility.
2. Click on the **Restore and Manage** Media tab.

3. Navigate to the backup of the system state, make sure that **System State** is checked. Under **Restore files to**, make sure **Original location** is selected, and click **Start Restore**.

4. You will then be prompted that **Restoring System State will always overwrite current System State unless restore to an alternate location.** Click **OK**. Then click **OK**, to **Confirm Restore**.

5. When the Restore completes, click **Close**.

6. You will then be prompted to restart your computer, click **Yes**.

---

**Performing System State Backup Windows Server 2008 R2**

1. If you have not installed Windows Backup, you will first have to install this feature. Open **Server Manager**, select the **Features** node, then click **Add Features**.

2. In the **Add Features Wizard**, select **Windows Server Backup Features**, then click **Next**, and then **Install**. When the installation completes, click **Close**.

3. You can then launch the Windows Server Backup tool, by clicking **Start**, then **Administrative Tools**, then **Windows Server Backup**.

4. Also, to use Windows Server Backup, you have to have an additional drive or a network location to backup to. In other words you cannot save the backup on the system drive.

5. The wizard allows you to configure a one-time backup, or schedule a backup.

6. To schedule a backup, click **Backup Schedule...**, under the **Actions** sections of the **Windows Server Backup** tool.

7. This will start the **Backup Schedule Wizard**, click **Next**.

8. On the **Select Backup Configuration** page, select **Custom**, and then click **Next**.

9. On the **Select Items for Backup** page of the wizard, click the **Add Items** button.

10. Select **System State**, and click **OK**, then click **Next**.

11. On the **Specify Backup Time** page of the wizard, select the time that you would like the backup to be scheduled for, and click **Next**.

12. On the **Specify Destination Type** page of the wizard, select either Hard Disk, Volume, or Shared Network Folder, and click **Next**. In this example, I am selecting Hard Disk

13. Select the Hard Disk you would like to use for backup, if it is not listed, click **Show All Available Disks...**, and select the appropriate disk, and click **OK**. Click **Next**.

14. You will be prompted that the disk will be reformatted and existing volumes will be deleted, click **Yes** if you are using this disk solely for backups, if not choose another backup destination.
15. On the **Confirmation** page, click **Finish**.

16. On the **Summary** page, click **Close**.

**Restoring System State in Windows Server 2008 R2.**

1. In the Actions page of the Windows Server Backup tool, click **Recover…**

2. This will start the **Recovery Wizard**, select the location of the backup, and click **Next**.

3. On the **Select Backup Date** of the wizard, select the date and time of the backup and click **Next**.

4. On the **Select Recovery Type**, select **System state**, and click **Next**.

5. On the **Select Location for System State Recovery** page, select **Original location**, and click **Next**.

6. On the **Confirmation** page of the wizard, click the **Recover** button.

7. You will be prompted that the recovery cannot be paused or cancelled once started, click **Yes**.

**Manual Backup of the Certification Authority**

A good guide to use for backing up and restoring a certification authority is:

298138 How to move a certification authority to another server


Steps 1 through 3 of this document cover manually backing up the CA.

Essentially, you want to do a manual back up of the private key, CA certificate, and CA database. If you are using an HSM to protect the private key pair, you will either need to backup the private key through a method provide by the HSM vendor or have a highly available configuration for the HSMs. In general, if the private key is stored on an HSM, you do not want to backup the private key to any type of media, as this will degrade the overall security and protection of the private key. The configuration for the Certification Authority is stored in the registry so you would want to backup that registry location as well. The registry location is HKLM\System\CurrentControlSet\Services\CertSvc\Configuration\<CA Name>.

Generally the private key, CA certificate and CA configuration are going to remain relatively static. You will, however, need to perform a fresh backup should you ever renew the CA certificate or update the configuration. However, the CA database is going to grow over time as certificates are issued, requests are denied, and certificates are revoked, so you are going to want to periodically backup the database. How often you perform this back up will depend on how rapidly changes to the database are made and how tolerant you are to discrepancies between the back up and the live data.

The first time you run the backup you will want to back up the CA’s certificate and private key, the CA database, and the certificate database log. To perform this task through the GUI, open up the Certification Authority MMC snap-in (certsrv.msc).

1. Right click on the certification authority name and select **All Tasks** from the context menu, and then select **Back up CA**...

2. This will launch the Certification Authority Backup Wizard, click **Next**.
3. Select **Private key and CA certificate** and **Certificate database and certificate database log.** Browse to a local or network location to save the backup. The backup location must be an empty folder, and click Next.

4. Enter a password to protect the private key, and click Next, then **Finish.**

To backup the CA via the command line, open an elevated command prompt and type `certutil –backup Path`. Path is the empty directory where the backed up information will be stored. You will then be prompted for a password to protect the private key. Enter the password and then press the Enter key. You will then be prompted to confirm the password. Confirm the password and press the Enter key. A message will be sent to the console indicating what has been backed up and that the `certutil –backup` command completed successfully.

To backup the registry run the following command: `REG EXPORT "HKLM\System\CurrentControlSet\Services\CertSvc\Configuration\<CA Name>" caconfig.reg`

Copy `caconfig.reg` to your backup directory so that all the necessary data is in the same place.

Once you have completed a full back up of the Certification Authority, you can perform incremental backups of the CA database. Alternatively, you could choose to periodically backup the entire CA database.

Although, you can back up the database through the Certification Authority console, you will most likely want to use some sort of script of scheduled task to perform the backup periodically.

**Manual Restore of the Certification Authority**

Once you relocate the server that will serve as the replacement for the failed CA, you must do some initial configuration of the server. Give that server the same name as the failed CA and join it to the same domain.

**Configure AD permissions**

Since you have brought online a new machine to be the CA we need to modify the security of Active Directory to allow the new machine to be able to update PKI configuration information in AD. This is because the new machine will have a new SID associated with the machine account, even though the machine account has the same name.

Open ADSIEDIT.MSC. Open the Configuration container of the Active Directory database. Browse to CN=Public Key Services, CN=Services, CN=Configuration. Next open the AIA container. Locate the object that is associated with the failed CA. Right click on that object, and select **Properties** from the context menu. Click on the **Security** Tab. Remove the CA’s computer account. Then re-add the CA’s computer account, and give it full control. This will associate the permissions with the new account.

Next open the CDP container. Locate the container associated with the failed CA. Open that container and then select the CRL object contained within that container. Right click on the CRL object, and select **Properties** from the context menu. Click on the **Security** Tab. Remove the CA’s computer account. Then re-add the CA’s computer account, and give it full control.

Next open the Enrollment Services container. Locate the object associated with the failed CA. Right click on that object, and select **Properties** from the context menu. Click on the **Security** Tab. Remove the CA’s computer account. Then click **Advanced**. In the Permissions tab of the **Advanced Security Settings** dialog box, click **Add...** Add the computer object for the CA. On the Permission Entry screen, select **Allow** for all Permissions except **Full Control**. Click **OK** 3 times.

Next open the KRA container. Locate the object that is associated with the failed CA. Right click on that object, and select **Properties** from the context menu. Click on the **Security** Tab. Remove the CA’s computer account. Then re-add the CA’s computer account, and give it full control. This will associate the permissions with the new account.
Installing the Certification Authority Role

Next we need to restore the Certification Authority. Log on with an account that has Enterprise Admin credentials. The first thing we will need to do is to install the Certification Authority Role. The instructions below are for a Windows Server 2008 and Windows Server 2008 R2 based CA. For exact procedures in Windows Server 2003. Please see the following article:

298138 How to move a certification authority to another server

1. Open Server Manager.
2. Click on the Roles Node, then click Add Roles.
3. When the Add Roles Wizard opens, click Next.
5. Then Click Next Again.
6. On the Select Role Services page of the wizard, select Certification Authority, and then click Next.
7. On the Specify Setup Type page of the wizard, Select Enterprise or Standalone depending on the configuration of the failed CA, and then click Next.
8. On the Specify CA Type page of the wizard, select either Root CA or Subordinate CA, depending on the configuration of the failed CA, and then click Next.
9. On the Set Up private key page of the wizard, select Use existing private key, and the sub-option of Select a certificate and use its associated private key, then click Next.
10. On the Select Existing Certificate page of the wizard, click Import.
11. Browse to the backup of the failed CA and select the P12 file from the backup, click Open. Then enter the password for the P12 file, and click OK.
12. Then click Next.
13. On the Configure Certificate Database page of the wizard, select the same database and log file locations as were specified on the failed CA, then click Next, then Install.
14. When the installation completes, click Close.

Open an elevated command prompt and use the following command to import the previously backed up CA configuration: REG IMPORT <Previously backed up registry file>.

Restore the CA Database

At this point, you can restore the CA database from your backup.

1. Right click on the certification authority name and select All Tasks from the context menu, and then select Restore CA...
2. You will be prompted to stop Certificate Services. Click Ok.

3. When the Certification Authority Backup Wizard starts, click Next.

4. Select Certificate database and certificate database log. Browse to a local or network location of your previously saved backup.

5. Click Next.

6. Click Finish.

7. You will be prompted to restart the CA. Unless you have further incremental backups to restore, click Yes. If you have incremental backups then click No, and walk through the steps above to restore your incremental backups.

Now if there were any additional Certificate Services roles such as Online Responder (OCSP) or Web Enrollment, you can go ahead and install those at this point.

**CRL Re-signing**

CRL re-signing is a manual process whereby the Administrator can use the CA's backed up certificate and private keys to re-sign an existing CRL file. This process allows you to extend the lifetime of the existing CRL, and even add certificates to the CRL, effectively revoking them.

**Importing the CA certificate and private key**

To begin, you will need to have a backup of the private key of the CA. If you have the private key stored on an HSM, you will have to follow the HSM vendor’s instructions for making the private key available to another machine. If you are not using an HSM, perform the following to import the CA public and private key pair to the machine where you will be re-signing the CRLs.

1. Click Start, then Run, and type MMC, and the press Enter.

2. Select the File Menu, and then select Add/Remove Snap-in...

3. Select Certificates, and then click Add >.

4. Then select Computer account, and click Next.

5. Then select Local computer, and then click Finish.

6. Then click OK.

7. Expand the Certificates (Local Computer) node.

8. Right click on the Personal node, then select All Tasks from the context menu, and then select Import...

9. This will open the Certificate Import Wizard, click Next.

10. Click the Browse button, to browse to the P12 file located in the CA's backup location.

11. In the drop down for the extension type, select Personal Information Exchange (*.pfx;*.p12)
12. Locate the P12 file that was previously backed up, and click **Open**.

13. Click **Next**.

14. Type the Password for the P12 file and click **Next**, click **Next** again, and click **Finish**.

15. Click **OK** to acknowledge that the import was successful.

To re-sign the CRL and Delta CRL with the same validity period as they have been previously published, use the following command:

```
certutil -sign <existing CRL file name> <re-signed CRL file name>
```


You will then have to manually publish the CRL to all CDP locations.

If you wish to adjust the validity period you can specify the validity period at the end of command in the following format D:D:HH, where D=Days, and H=Hours. For example, the following command would re-sign a CRL that is valid for 14 days:

```
certutil -sign <existing CRL file name> <re-signed CRL file name> 14:00
```

If you wish to add one or more issued certificates to the CRL, you specify the serial numbers in a comma separated list on the command line. For example, the following command would add serial numbers to the CRL:

```
certutil -sign <existing CRL file name> <re-signed CRL file name> +SerialNumber1,SerialNumber2,SerialNumber3
```

**Summary**

When building a PKI infrastructure it is critical to take into consideration how your design will have an effect on the availability of your PKI. However, the design also affects the way in which you may have to recover the CAs in the PKI.

You should definitely consider the criticality of PKI to your environment, and how much downtime is acceptable. This will help drive your decisions when designing the PKI and implementing the Certification Authorities.

Also, many customers make the mistake of either not being aware of how to recover a Certification Authority or do not have a documented process for doing so. When designing and implementing your PKI, I recommend that you test recovery and document the recovery steps for CAs in your PKI.

Chris "CLEAR!" Delay