Securing Public Key Infrastructure (PKI)

Microsoft IT

Information Security and Risk Management

Published: May 16. 2014

For the latest information, please see

<http://aka.ms/securingpki>

Contents

[Foreword 6](#_Toc381699478)

[Acknowledgements 7](#_Toc381699479)

[Executive Summary 8](#_Toc381699480)

[Introduction 10](#_Toc381699481)

[About this Document 11](#_Toc381699482)

[Content Origin and Organization 11](#_Toc381699483)

[PKI Assessments and Consulting 11](#_Toc381699484)

[Content Scope 11](#_Toc381699485)

[Introduction to PKI 11](#_Toc381699486)

[PKI Components 11](#_Toc381699487)

[PKI Governance 12](#_Toc381699488)

[Business Drivers for PKI 12](#_Toc381699489)

[Elements of a Successful PKI 13](#_Toc381699490)

[Determining the Level of Protection Required 14](#_Toc381699491)

[Compromising PKI 15](#_Toc381699492)

[How PKI Compromises Occur 15](#_Toc381699493)

[Protecting a PKI Deployment 16](#_Toc381699494)

[Planning a CA Hierarchy 17](#_Toc381699495)

[CA Hierarchy Options 18](#_Toc381699496)

[Conclusion 21](#_Toc381699497)

[Physical Controls for Securing PKI 22](#_Toc381699498)

[Functional Considerations 22](#_Toc381699499)

[Operational Considerations 23](#_Toc381699500)

[Designing Physical Security 23](#_Toc381699501)

[Track and Audit Physical Access Requests 23](#_Toc381699502)

[Consider Using Biometrics 24](#_Toc381699503)

[Use Multi Person Control 24](#_Toc381699504)

[Eliminate Tailgating to Sensitive Areas 24](#_Toc381699505)

[Use Alarm Systems as a Detective Control 25](#_Toc381699506)

[Use Cameras as a Detective Control 25](#_Toc381699507)

[Geographically Separate Primary and Backup Sites 25](#_Toc381699508)

[Use Security by Obscurity Carefully 25](#_Toc381699509)

[Conclusion 26](#_Toc381699510)

[PKI Process Security 27](#_Toc381699511)

[PKI Policy 27](#_Toc381699512)

[Certificate Policy 27](#_Toc381699513)

[Certification Practice Statement 28](#_Toc381699514)

[PKI Governance and Oversight 29](#_Toc381699515)

[Roles and Responsibilities 30](#_Toc381699516)

[Key Generation Ceremonies 31](#_Toc381699517)

[Conclusion 33](#_Toc381699518)

[Technical Controls for Securing PKI 34](#_Toc381699519)

[Securing the CA Operating System 34](#_Toc381699520)

[Creating a Baseline Configuration for all CAs and RAs 34](#_Toc381699521)

[Microsoft Security Compliance Manager 34](#_Toc381699522)

[Microsoft Security Configuration Wizard 34](#_Toc381699523)

[Online CA Hardening Recommendations 34](#_Toc381699524)

[Additional Roles on Certification Authorities 35](#_Toc381699525)

[Alternate Administrative Accounts 35](#_Toc381699526)

[Updating Online Certification Authorities 35](#_Toc381699527)

[Internet Access from Certification Authorities 36](#_Toc381699528)

[Local Administrators Group Membership 36](#_Toc381699529)

[Application Whitelisting 36](#_Toc381699530)

[Securing Remote Management Tasks 37](#_Toc381699531)

[Multi-factor Authentication for Certification Authority Access 37](#_Toc381699532)

[Securing Offline Certification Authorities 38](#_Toc381699533)

[Protect CA Private Keys 38](#_Toc381699534)

[Offline CAs Should Be Truly Offline 38](#_Toc381699535)

[Managing Data Transfer 39](#_Toc381699536)

[Updating Offline Certification Authorities 39](#_Toc381699537)

[Account Management 39](#_Toc381699538)

[Virtualizing Certification Authorities 39](#_Toc381699539)

[Offline Certification Authorities 39](#_Toc381699540)

[Online Certification Authorities 41](#_Toc381699541)

[Delegating PKI Tasks 41](#_Toc381699542)

[Role Separation 42](#_Toc381699543)

[Protecting CA Backups 43](#_Toc381699544)

[Network Isolation 44](#_Toc381699545)

[Securing Certificate Templates 45](#_Toc381699546)

[Controlling User Added Subject Alternative Names 47](#_Toc381699547)

[Conclusion 48](#_Toc381699548)

[Planning Certificate Algorithms and Usages 49](#_Toc381699549)

[Cryptographic Algorithms, Key Lengths, and Validity Period 49](#_Toc381699550)

[Selecting Algorithms and Key Lengths 49](#_Toc381699551)

[Certificate Validity Periods 50](#_Toc381699552)

[Mixing Cryptographic Algorithms 52](#_Toc381699553)

[CA Key Usages and Certificate Extensions 53](#_Toc381699554)

[Key Usage 53](#_Toc381699555)

[Extended Key Usages 54](#_Toc381699556)

[Critical Extensions 56](#_Toc381699557)

[CA Certificate Constraints 56](#_Toc381699558)

[Basic Constraints 56](#_Toc381699559)

[Extended Key Usage Constraints 57](#_Toc381699560)

[Constraining CA Certificates 57](#_Toc381699561)

[Conclusion 58](#_Toc381699562)

[Protecting CA Keys and Critical Artifacts 59](#_Toc381699563)

[Protecting CA Keys 59](#_Toc381699564)

[Migrating Software Keys to HSMs 60](#_Toc381699565)

[Network Based HSMs 60](#_Toc381699566)

[Multi Person Control 60](#_Toc381699567)

[Artifact Protection and Chain of Custody 64](#_Toc381699568)

[Use Tamper-Evident Containers 64](#_Toc381699569)

[Artifact Storage 65](#_Toc381699570)

[Maintain a Chain of Custody and Asset Inventory 65](#_Toc381699571)

[Conclusion 66](#_Toc381699572)

[Monitoring Public Key Infrastructure 67](#_Toc381699573)

[Monitoring Events 67](#_Toc381699574)

[Monitoring Active Directory Objects and Attributes 68](#_Toc381699575)

[Monitoring Certification Authority Activity 68](#_Toc381699576)

[Recording and Reviewing Additional Events 69](#_Toc381699577)

[Configuring Microsoft Windows Audit Policy 70](#_Toc381699578)

[Command Line Process Auditing 71](#_Toc381699579)

[Configuring Certification Authority Auditing 71](#_Toc381699580)

[Advanced CA Monitoring 73](#_Toc381699581)

[Auditing CA Registry Changes 73](#_Toc381699582)

[Monitoring Changes to Certificate Templates 75](#_Toc381699583)

[Conclusion 76](#_Toc381699584)

[Compromise Response 77](#_Toc381699585)

[Inventory PKI Consumers 77](#_Toc381699586)

[Understanding Compromise Scenarios 77](#_Toc381699587)

[Full Key Compromise 77](#_Toc381699588)

[Full Key Access 77](#_Toc381699589)

[Limited Key Access 78](#_Toc381699590)

[Other Attacks 78](#_Toc381699591)

[Full Key Compromise and Access 78](#_Toc381699592)

[Operating System Compromise 78](#_Toc381699593)

[Cryptographic Compromise 79](#_Toc381699594)

[CA Compromise Response Actions 79](#_Toc381699595)

[Correlating Compromise Types and Response Actions 81](#_Toc381699596)

[Conclusion 83](#_Toc381699597)

[Appendices 84](#_Toc381699598)

[Appendix A: Events to Monitor 85](#_Toc381699599)

[Registry Values to Monitor 105](#_Toc381699600)

[Appendix B: Certification Authority Audit Filter 108](#_Toc381699601)

[Appendix C: Delegating Active Directory PKI Permissions 110](#_Toc381699602)

[Permissions for Enterprise CA Installation 110](#_Toc381699603)

[Delegating Rights to the Public Key Infrastructure Container 110](#_Toc381699604)

[Delegating Group Permissions 114](#_Toc381699605)

[Permissions for Managing Certificate Templates 116](#_Toc381699606)

[Appendix D: Glossary of Terms 117](#_Toc381699607)

[Appendix E: PKI Basics 121](#_Toc381699608)

[Appendix F: List of Recommendations by Impact Level 122](#_Toc381699609)

[Planning a CA Hierarchy 122](#_Toc381699610)

[Physical Security 122](#_Toc381699611)

[PKI Process Security 123](#_Toc381699612)

[Technical Controls for Securing PKI 124](#_Toc381699613)

[Planning Certificate Algorithms and Usages 128](#_Toc381699614)

[Protecting CA Keys and Critical Artifacts 129](#_Toc381699615)

[Monitoring Public Key Infrastructure 129](#_Toc381699616)

[Compromise Response 130](#_Toc381699617)

# Foreword

Attacks against computing infrastructures have existed as long as computers. However, in recent years, these attacks have become ubiquitous and increasingly sophisticated. This trend will continue and escalate with the requirement for global information exchange amongst employees, suppliers, partners and customers increasing at a rapid rate.

With increased sharing of information comes increased threats to the confidentiality, integrity and availability of sensitive business data. These threats require organizations to develop methods to provide increased security for their information.

Electronic credentials that prove identity are a critical necessity. Much like a passport proves identity in the offline world, Public Key Infrastructure (PKI) delivers a way to prove identity in the online world. PKI also supports secure information exchange over insecure networks and can protect the transport of information, which is critical when conducting online transactions.

This whitepaper contains recommendations for establishing a robust, secure PKI to help organizations provide basic security controls such as confidentiality and integrity to key business processes. If properly implemented, a PKI becomes a foundational component used to build effective information security controls over information resources. PKI plays a critical role in the protection of sensitive business data and is an enabling technology that enhances information systems security and promotes secure electronic commerce.

This paper contains guidance and recommendations necessary for establishing a Certification Authority (CA), an understanding of the physical controls for securing a PKI, the processes vital to establishing a PKI, the technical controls for securing a PKI, procedures for planning certificate algorithms and their usages, procedures for protecting CA Keys and critical artifacts, monitoring the PKI, and compromise response. Information security and risk management practitioners will find the policies, procedures, and recommendations to be a significant contribution to their understanding in the practical implementation of a robust PKI.

Bret Arsenault  
Microsoft Chief Information Security Officer

# Acknowledgements

|  |  |  |  |
| --- | --- | --- | --- |
| **Authors**  Mike Ricks – Lead Author  Sergey Simakov  Shawn Rabourn  **Contributors**  Laura A. Robinson  Roger Grimes  Kelvin Yiu  **Executive Oversight**  Alex Simons  Dan Plastina  Dave Gasiewicz  Dustin Ingalls  Bret Arsenault  Pat Arnold  Joe Lindstrom  Mario Rodriguez | **Reviewers**  Adam Arndt  Ahmad Mahdi  Amer Kamal  Ashish Popli  David Hoyle  Eric Leonard  Fernando Cima  Jason Lee  Jenn LeMond  John Mason  Keith Proctor  Kelvin Yiu  Larry Talbot  Mark Cooper  Rashmi Jha  Troy Arwine  Zane Lewiston |  |  |

# Executive Summary

Many organizations deploy a Public Key Infrastructure (PKI) to support critical business functions. PKI plays a critical role in helping a business provide basic security controls such as confidentiality and integrity to key business processes. PKI often is used as a mechanism to provide strong authentication of users for uses such as Virtual Private Networks (VPNs) or access to business critical data. Since PKI systems often act as a central resource that can allow a high level of access to an IT infrastructure, they are a logical target for any persistent and determined attacker. While attacking a PKI is typically not the end goal of an attacker, compromising a PKI can provide an attacker with the credentials they need to further their attack more effectively.

Security of the systems and processes comprising a PKI should be the first and foremost considerations when designing and deploying a PKI. Prior to deployment, careful consideration should be given to the numbers and types of certification authorities that you will deploy. Design your PKI with any potential future use cases in mind, because an improper design can lead to significant rework or security ramifications as your PKI sees new and different use cases.

Physical controls should not be overlooked during the design. While much emphasis is placed on controlling access to systems over the network, physical attacks can still occur and can lead to the full compromise of your PKI, and subsequent compromise of other critical systems that depend on it. Secure critical infrastructure from physical attacks that may originate from an outside attacker or potentially an insider threat.

A large portion of the work involved with running a successful PKI is establishing secure and repeatable processes that are well documented and followed. Any PKI that intends to be trusted for critical business functions should document the policies and procedures. These policies and procedures can then be used in Certificate Policy and Certification Practice Statement documents for a PKI trusted by customers and partners. Strong processes should be developed to ensure that the PKI is run with oversight from the proper teams within your organization.

PKI systems should be treated as critical systems and have strong technical controls deployed to protect them from unauthorized access. As with other critical systems, hardened baselines and strict management access should be implemented. Special consideration should be given to PKI systems that are always kept offline. Implement controls for offline systems so they are never brought online and do not have malicious software introduced to them.

When running a PKI deployed with Microsoft Windows® Active Directory® Certification Services, care should be taken to control access to sensitive PKI tasks. Control access to certificate templates that allow for certificate enrollment, and ensure that supporting processes such as backing up sensitive data and publishing PKI data are properly secured.

Another key decision in PKI design is the suite of cryptographic algorithms that are used and the length of time certificates are valid. Ensure that you select the strongest cryptographic algorithms afforded by your environment, and be aware of potential compatibility issues between different operating systems and cryptographic software packages. After proper algorithms are selected, ensure that the attributes you allow in your certificates enable only the intended usages and do not expose your organization to misuse of issued certificates.

Protection of the keys associated with a Certification Authority (CA) or other PKI systems is paramount. Strong protection mechanisms should be deployed because a compromise of a key can lead to a compromise of your IT infrastructure. Deploy critical CAs with hardware security modules (HSMs) to protect keys and ensure that any artifacts (backups, HSM objects, etc.) kept are stored securely.

PKI systems should be monitored continuously for signs of compromise. Monitoring can uncover attempts to enroll unauthorized certificates for important users (VIPs), such as system administrators, executives, and individuals with access to business critical data. Monitor for changes to critical accounts and changes to key security groups. Watch for anomalous behavior in certificate enrollment and have robust alerting capability when events are detected.

In the event that your PKI is compromised, it is important to understand the extent of the breach and the response options. Your response will be highly variable, depending on the level of access an attacker was able to obtain, along with the level of impact your organization is willing or able to bear on the business processes supported by the PKI.

Deploying and maintaining a secure PKI is not a trivial task. A PKI is a high value target for a determined attacker, and is often used as part of a broader campaign to obtain sensitive organizational data. A PKI that is not properly secured can lead to the compromise of most, if not all systems in your IT infrastructure. As your organization deploys PKI, ensure the proper investment is made to deploy it securely and implement adequate support for ongoing operations and monitoring.

# Introduction

Attacks against computing infrastructures, whether simple or complex, have existed as long as computers. However, within the past decade, increasing numbers of organizations of all sizes, in all parts of the world have been attacked and compromised in ways that have significantly changed the threat landscape. Cyber-warfare and cybercrime have increased at record rates. “Hacktivism,” which attacks are motivated by activist positions, has been claimed as the motivation for a number of breaches intended to expose organizations’ secret information, to create denials-of-service, or even to destroy infrastructure. Attacks against public and private institutions have become ubiquitous, with the goal of exfiltrating the organization’s intellectual property (IP).

When looking at your environment, an appropriate mindset is to assume that you have already been breached or a breach will occur at some point. No organization with an information technology (IT) infrastructure is immune from attack, but if appropriate policies, processes, and controls are implemented to protect key segments of an organization’s computing infrastructure, escalation of attacks from penetration to complete compromise might be preventable. The principles and recommendations provided in this document are intended to help secure your environment against external attackers and misguided or malicious insiders.

The information and recommendations provided in this document are drawn from a number of sources and derived from practices designed to protect Public Key Infrastructure (PKI) installations against compromise. Although it is not possible to prevent attacks, it is possible to reduce the attack surface and to implement controls that make compromise much more difficult for attackers. This document presents some of the most common types of vulnerabilities observed in compromised environments and most common recommendations to improve the security of PKI installations.

While many of the recommendations found in this document are specific to PKI installations that utilize Active Directory® Certificate Services (AD CS) as a technology platform, many of the recommendations are also vendor agnostic and can be implemented in any PKI deployment.

If you operate a PKI within an Active Directory® (AD) environment, it is strongly recommended that you also read [Best Practices for Securing Active Directory](http://aka.ms/bpsad). Since the security of the AD CS installation is in many respects directly dependent on the security of the Active Directory® installation, an understanding of the avenues to compromise and methods to secure Active Directory® will greatly enhance your ability to secure your PKI deployment.

## About this Document

### Content Origin and Organization

Much of the content of this document is derived from knowledge gained from years of PKI and Active Directory® assessments performed by numerous Microsoft information security professionals. Additional recommendations are gathered from extensive experience gained from operating numerous PKIs at Microsoft, both for external and internal uses. Although individual customer data was not used to create this document, the most commonly exploited vulnerabilities identified from our assessments and the recommendations to improve security of AD CS installations are utilized. Not all vulnerabilities are applicable to all environments, nor are all recommendations feasible to implement in every organization.

### PKI Assessments and Consulting

Within Microsoft, numerous organizations provide consulting services for PKI. Depending on the needs of the customer, an engagement may include PKI design, implementation, compromise response, or assessment for health and security. Microsoft provides customers with recommendations based on their organization’s unique characteristics, practices, and risk appetite. PKI assessments performed internally at Microsoft and those of our customers have contributed to the recommendations in this paper.

### Content Scope

This paper is not intended to address all potential security issues across PKI, but rather is focused on providing actionable content for areas in which Microsoft has seen deficiencies while performing assessments over several years with many customers. Specifically, this document does not attempt to provide recommendations for other Active Directory® Certificate Services roles such as Network Device Enrollment Services (NDES) or Online Certificate Status Protocol (OCSP).

## Introduction to PKI

### PKI Components

Within any PKI regardless of the technical implementation, a number of components and actors are present. This brief introduction will help provide context and definition of terms used throughout this whitepaper.

**Subscriber/End Entity** –The person or computer listed as the subject in a certificate. In the context of this whitepaper, an End Entity certificate is any certificate issued to a person or computer.

**Relying Party** –A user or system that consumes the certificates generated by the PKI. Examples of relying parties would be users of web browsers visiting an SSL protected web site, a VPN server authenticating a remote user, or a computer validating that an executable is signed correctly before running it.

**Certification Authority (CA)** – Broadly, a CA is a set of systems and practices used by an organization to issue and manage certificates. For the context of this paper, the CA will be the system responsible for signing certificates and running CA software such as Active Directory® Certificate Services.

**Registration Authority (RA)** – A CA may delegate some of the tasks related to verifying identities and processing certificate revocation requests to a Registration Authority (RA). An RA could be a person performing manual checks or a system automating the required validation checks. The CA will accept certificate requests processed by the RA and sign them, trusting that the RA has done the proper validation.

**PKI Repositories** – A PKI repository is a location accessible by relying parties where PKI information is stored. This information could include certificates, revocation information, or information regarding the policies of the PKI. For example, this information can be posted to internal or external web locations.

For a glossary of terms used in this whitepaper, refer to [Appendix D: Glossary of Terms](#_Appendix_D:_Glossary).

For basic information about PKIs, refer to [Appendix E: PKI Basics](#_Appendix_E:_PKI).

### PKI Governance

Maintaining the trust of relying parties is an integral component to running a PKI. If a relying party does not know the policies a PKI uses for governance, or the PKI has no formal policies, they cannot make an informed decision to trust that the certificates issued by the PKI are correct. To facilitate trust, a PKI must be operated with some level of oversight, and with policies, standards, and procedures in place to control how the PKI is managed. The level of oversight required will vary depending on the intended use. For example, a PKI trusted and used internally at a company to issue certificates for wireless network authentication does not need the same amount of oversight as a PKI used to issue SSL certificates trusted by default by all major web browsers. Governance is not a topic unique to PKI- other critical infrastructure systems should also be run with formalized change control and oversight.

### Business Drivers for PKI

Whenever a new piece of technology is introduced to an environment, it should be for a defined business reason and PKI is no exception. In many cases, we have observed a PKI introduced at a company in support of a specific solution or product. For example, a PKI could be constructed to provide end user certificates in support of secure remote access. In general, PKI can be thought of as an enabler for providing solutions for the following high-level business requirements:

* **Data Encryption** –PKI provides several solutions for securing data in transit and at rest. PKI is commonly included as a part of a solution to securely exchange data between partners, customers, or internal sites. Common implementations include using certificates to enable Transport Layer Security (TLS) for secure transmission of data, S/MIME certificates for sending encrypted email, and Encrypting File System (EFS) certificates to encrypt files on a workstation.
* **Authentication** – PKI is often used as a solution to provide high-assurance authentication credentials. Common implementations include issuing certificates for private keys that reside on a smart card for multifactor authentication and issuing certificates to computers and devices for network authentication.
* **Data Integrity** – PKI can be used to ensure that critical data has not been altered and that it comes from a trusted source. One common implementation is code signing, where files are signed by the developer when a software package is released. Another example is electronic document signing, where certificates can be used to allow a person to “sign” a document to show the document has not been altered since the signature took place.

### Elements of a Successful PKI

For a PKI deployment to be successful, several factors must be in place:

* **Understanding of Business Requirements and Objectives** – Like any investment made by an organization, the business requirements and value provided by the PKI should be well understood prior to implementation. Understanding how the PKI supports the business, what processes it supports or enables, and any externally imposed requirements allows you to make informed decisions on the level of risk you are willing to accept when designing the system. For example, an internal PKI supporting wireless LAN authentication will be designed and secured differently than a PKI built for issuing SSL certificates that are trusted by external organizations.
* **Risk Assessment** – A proper risk assessment and threat modeling should be performed prior to deploying a PKI. A risk assessment will determine the level of security and the investment that should be made in the PKI.
* **Executive Support** –As with any other large-scale IT project, support from executive management is crucial in the deployment of a PKI that meets large-scale needs. A properly implemented PKI often represents a significant investment, both in capital and human resources. Executive management needs to have a clear vision of why PKI must be designed and operated differently from commodity IT services, as well as have a clear understanding of the business requirements the PKI helps to satisfy.
* **Planning and Foresight** –A PKI deployment is often unique from deployments of other technologies, because more stringent security controls are required for the deployment to succeed. A PKI deployment also requires the development of rigorous operating procedures. For a PKI to succeed, careful planning must occur to ensure that the policy, procedures, and technical implementation meet the needs of the business, both now and into the future. If a PKI is configured incorrectly, frequently the only good solution is to start over and implement a new system. With proper planning, you will be able to avoid costly rework or compromise.

When planning a deployment, pay special attention to any shorter term (12-18 months) uses of the PKI. If you do not plan at the very beginning to accommodate known future uses, you may end up setting up parallel infrastructures in the future, or be forced to modify the existing infrastructure in ways that make it more complex and costly to operate.

* **Defined Policies** –Prior to implementing any certification authorities or issuing certificates, define and agree upon the policies which govern the use of the PKI. Applications either inside or outside your environment will be dependent on the PKI, and these policies will provide clear guidance on what to expect for topics such as certificate issuance, security, disaster recovery, etc. Policies do not need to be overly complex, but it is critical to develop and follow them.
* **Ongoing Governance and Oversight** –Governance plays a significant role in a successful PKI because a PKI is not a static system. There will likely be ongoing changes within your environment that will drive operational or security changes to your PKI. Proper governance ensures the risk of any changes introduced are well understood, carefully considered, and are well communicated to the community of relying parties.

### Determining the Level of Protection Required

As stated in the prior section, it is critical to perform a risk assessment and develop a threat model prior to deploying a PKI. The recommendations provided throughout this paper are not one-size-fits-all for every PKI deployment; each deployment is unique in its requirements. To assist in determining what recommendations may apply to your deployment, each recommendation has been categorized based on impact level, which (at a high level) is a measure of the impact a CA breach would have on the business it supports. The following impact levels are used to classify the recommendations:

* **High Business Impact – Customer or Partner Impacting** –This classification relates to CAs that, if compromised, could impact your customers or partners, as well as have an internal impact.
* **High Business Impact –** **Internal Impact** –This classification relates to CAs that if compromised, could impact internal operations of your company significantly. Examples include CAs that issue certificates used to protect proprietary data, or allow authentication to internal systems or networks. Many ADCS deployments, regardless of intended use, will fall into this category.
* **Moderate Business Impact** – This classification relates to CAs that are not widely trusted in your environment or do not support critical business processes.

Each CA should be analyzed to determine the impact of a breach. In most cases, all CAs in a hierarchy will have the same impact level. However, there are some cases where a subordinate CA may have a lower impact level than the root or other subordinate CAs because technical constraints may prevent it from being used in more critical use cases. For a complete list of recommendations in this paper along with the recommended impact level at which to implement them, refer to [Appendix F: List of Recommendations by Impact Level](#_Appendix_F:_List).

## Compromising PKI

Except in exceptional cases, compromise of a PKI is not the ultimate goal of an attacker. Rather, an attacker’s motivation is typically to acquire other desirable data such as trade secrets, payment card data, or other data. Compromising a PKI in an environment may be a necessary step for attackers to gain the credentials required to access the desired data. There are several compelling reasons why an attacker may attempt to compromise a PKI as part of an enterprise breach:

* **Elevation of privilege** – An attacker can leverage the PKI to gain credentials that will allow them to gain full access to desired systems or potentially all systems across the target environment.
* **Persistent access** – While many attacks begin with attackers using backdoors on systems to maintain access to an environment, compromising a PKI and obtaining credentials can allow attackers to use the “front door” and access the environment through legitimate means such as a Virtual Private Network (VPN) or DirectAccess (DA).
* **Impersonation** – Compromising a PKI can allow an attacker to create credentials of VIP users who have access to desired data.

In many cases, compromising a PKI and obtaining fraudulent certificates allows an attacker to traverse the network using methods that look very similar to normal user activity and are difficult to detect.

### How PKI Compromises Occur

PKI systems should be guarded as some of the highest security systems in an environment, so an attacker may require several steps to accomplish a compromise. In a typical breach, attackers gain a foothold into the environment by exploiting systems not current with security patches or outdated applications. For a more complete list of common attack vectors in an Active Directory® environment, refer to [Best Practices for Securing Active Directory](http://aka.ms/bpsad). Once the attacker gains some basic level of access within the network, the following are some of the common scenarios that result in a compromise of the PKI:

#### Misconfigurations

A common method of compromise is for attackers to leverage misconfigurations within the PKI to issue certificates for other users of systems for which the requesting user should not have rights to request, or certificate types that the user should not be able to request. Examples include misconfigurations in template permissions, such as overly broad enrollment permissions, or misconfigurations on the CA that allow users to request certificates with user-defined data. Misconfigurations in allowed certificate usages and constraints could also allow attackers to create subordinate CAs with arbitrary attributes.

#### Inadequately Secured Certification Authority Systems

In many cases, the CA servers are run and managed similarly to any other system in the environment. This includes the CA systems being built from the same image as other systems and includes many unnecessary applications that increase the attack surface of the CA. Operating a CA in this manner exposes it to many additional attack vectors, including attacks from other enterprise systems such as monitoring, update management, backup, inventory, etc. PKI attacks are sometimes opportunistic in the sense that an attacker may not have been planning on compromising PKI, but because they gained a credential for an account that had access to the PKI, they leverage it. Common vectors for attack in this scenario are passwords left with default build values, generic configurations of software that allow overly broad access, or excessive administrator rights on the CA.

In other cases, the CA is built using a custom “one off” configuration where adequate documentation does not exist and the configuration has not been thoroughly tested. While custom installations can lessen the attack surface when done correctly, they can also introduce misconfigurations, as standard processes are often not followed. Compromised CAs often do not adequately protect the CA keys, leaving them vulnerable to attackers either on the system or in backups.

#### Inadequately Secured RA Systems

Many PKI designs use a front end RA system to handle validating requests for certificates. While the back end CA may have good security, in some cases the RA systems are not secured in the same manner. If this is the case, attackers may be able to compromise the RA and use its credentials to issue the certificates they need to further their attack. Some examples of RA systems include smart card management software, Network Access Protection (NAP) Health Registration Authority (HRA), and Network Device Enrollment Services (NDES).

#### Social Engineering

Each CA should implement its own set of practices, both logical and process-driven to ensure the identity of the subject requesting a certificate is validated. If there are deficiencies in these practices, it is possible for an attacker to use social engineering techniques to have certificates issued to them. For example, an attacker could have an employee smart card shipped to them by pretending to be a remote employee if home addresses are not validated. An attacker pretending to be a web administrator could have an SSL certificate issued on behalf of a highly sensitive payroll system to perform a man-in-the-middle attack.

### Protecting a PKI Deployment

Whether you are operating an existing PKI or planning to deploy a new PKI in your environment, the recommendations provided in this paper are intended to mitigate many of the common insecurities and shortcomings addressed in the previous section. Beginning with recommendations to consider during the design phase, each section will cover an aspect of PKI systems and provide recommendations for specific approaches or general guidance to consider for the threats in your environment. Not all the recommendations provided in this paper will be applicable to every environment. Consider the [impact a breach of the CA](#_Determining_the_Level) would have on your business through a risk assessment and implement those controls which help to mitigate the known threats. A comprehensive list of all recommendations in the paper is provided in [Appendix F: List of Recommendations by Impact Level](#_Appendix_F:_List), along with the impact level at which you should apply the recommendation.

# Planning a CA Hierarchy

Certificate hierarchy planning is one of the most important aspects of PKI design because the design will affect how certificates are validated and used by PKI-enabled solutions. This section introduces a number of recommendations for designing a certificate hierarchy that can be used to meet today’s pressing business needs as well as future needs that may not yet be identified.

A PKI can be implemented either as part of the IT infrastructure or by using external, commercial CAs. In general, the following are the PKI design options:

* Implement a completely self-managed PKI within your organization that contains internal CAs chained to an internal root CA at the top of the chain
* Purchase a CA certificate from a commercial CA and issue certificates within the organization from internal, self-managed CAs that are chained to the external root CA
* Purchase certificates from a commercial CA that are chained to a public root CA (preferably a member of the [Microsoft Root Certificate Program](http://go.microsoft.com/fwlink/?LinkID=269988) that are automatically distributed to clients that use Microsoft Windows® platforms)

If the security solutions supported by the PKI do not require external parties to trust the issued certificates, and will not in the future, then you may opt for a self-managed PKI that uses an internal root CA as the trust anchor for the PKI. Using an internal root CA allows you to maintain direct control over its security policies and to align its Certificate Policy (CP) with the overall security policy. Therefore, you will use internal CAs for issuing certificates to end users, to computers, and to services. These internal CAs can be expanded to include additional functionality, such as support for new certificate types, at a lower cost than buying external certificates.

## CA Hierarchy Options

In a hierarchical PKI (a typical deployment), there are generally three types of hierarchies – one tier, two-tier, and three-tier.



Single/One-Tier Hierarchy

* **One-Tier Hierarchy** – Consists of one single CA. The single CA is both a root CA and an issuing CA. A root CA is the trust anchor of the PKI, so a root CA public key serves as the beginning of trust paths for a security domain. Any applications, users, or computers that trust the root CA also trust any certificates issued by the CA hierarchy. The issuing CA is a CA that issues certificates to end entities.

This one-tier hierarchy **is not recommended for any production scenario** because with this hierarchy, a compromise of this single CA equates to a compromise of the entire PKI. For security reasons, root and issuing CAs are normally separated because it is generally very difficult to quickly distribute a new root CA certificate to replace a compromised CA. This is especially true when the environment includes computers not joined to management domain or devices where a special process is required to provision the root CA certificate. Because of this, a one-tier hierarchy may be sufficient for only simple implementations where ease of manageability and lower cost outweigh the need for greater levels of security or flexibility.

A common finding in PKI assessments is that some organizations install a single CA in order to support a major project that may have required it. Perhaps this is an installation of System Center Configuration Manager, or wireless network protection, or some other PKI-consuming technology and one small line-item in the project’s plan is dedicated to the CA installation. Over time, this single CA begins to get a lot of use as it is leveraged more and more for purposes other than those originally conceived. Suddenly, there is a need for a proper PKI and administrators face some uneasy questions:

* Can I install multiple PKIs in my forest without them interfering with each other?
* How do I set up my new PKI properly so that it is scalable and manageable?
* How do I get rid of my old CA without causing an interruption in my business?
* Is the configuration of this CA posing a security risk to the enterprise?

So there are multiple security risks in using a one-tier hierarchy – your only CA is online and more susceptible to compromise, and you cannot revoke this CA if it was compromised. This hierarchy is also more difficult to expand to support other scenarios. If you are in the position to move to the recommended CA hierarchy design, refer to the [Moving Your Organization from a Single Microsoft CA to a Microsoft Recommended PKI](http://blogs.technet.com/b/askds/archive/2010/08/23/moving-your-organization-from-a-single-microsoft-ca-to-a-microsoft-recommended-pki.aspx) article.



**Three-Tier Hierarchy**

* **Three-Tier Hierarchy** – In a three-tier hierarchy, there is a root CA tier (offline), an issuing CAs tier (usually online), and an intermediate tier placed between them. The placement of this intermediate CA can be for several different reasons. The first reason would be to use the second tier CA as a policy CA. For example, one policy CA will issue certificates that requires that a user has to appear in person and another CA will issue certificates to any authenticated corporate users. In other words, the policy CA is configured to issue certificates to the Issuing CA that is restricted in the type of certificates it issues. The policy CA can also just be used as an administrative boundary. That is, 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 and not technical perspective.

Another reason to have the second tier added is 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 available. It should be noted that second tier CAs in this hierarchy should, 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 cost goes up.

Note that performance of certificate chain building on PKI solution clients is affected with an increase in the number of tiers because three-tier hierarchy clients need to verify certificate and certificate status information for both issuing CAs and policy CAs. Another consideration is policy or administrative boundary requirements because a three-tier hierarchy will increase operational costs. Also note that if you are not going to implement policy or administrative boundaries, then the middle tier will be unused and is unneeded. Because of this, three-tier CA hierarchies are usually not recommended (with the exception of a few unique cases). In fact, Microsoft IT changed its design to a two-tier CA hierarchy for its internal PKI. Refer to [Deploying and Managing PKI inside Microsoft](http://technet.microsoft.com/en-us/library/cc964304.aspx) for more information.



**Two-Tier Hierarchy**

* **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 so the private key of the root CA is better protected from compromise. The two-tier hierarchy also increases scalability and flexibility due to the fact that there can be multiple issuing CAs subordinate to the root CA. This allows CAs to exist in different geographical locations, as well as with different security levels. Manageability is increased since the root CA has to be brought online to sign CRLs. Capital cost is increased marginally because all you need is an additional server or a virtual machine. The two-tier hierarchy is the recommended design for most PKI solutions.

Another advisable idea is to restrict certificates of the subordinate issuing CAs to limit their impact on the CA hierarchy, so that subordinate CAs cannot issue “rogue” certificates that could be used for unintended purposes. This is important when you have more complex certificate hierarchies. The most obvious case occurs when an issuing CA is operated by different parts of an organization, but even for internal issuing CAs, it may make sense to restrict the scope of issued certificates. You want to limit certificates issued for a specific scenario (for example, server authentication) so one CA does not affect others (for example, issuing certificates for smartcards) in case of security breach. The best way to accomplish this task is to implement path length constraint and limit Extended Key Usages (EKUs) for the issuing CA’s certificate as described in the [Planning Certificate Algorithms and Usage](#_Planning_Certificate_Algorithms) section. Please note that there are many shared elements of multiple enterprise subordinate CAs in Active Directory® environment (for example, certificate templates), so this restriction should not be the only mitigation.

## Conclusion

For a complete list of the recommendations for planning a CA hierarchy, along with the level of [business impact](#_Determining_the_Level) at which you should consider implementing them, refer to [Appendix F: List of Recommendations by Impact Level](#_Planning_a_CA).

# Physical Controls for Securing PKI

In today’s threat landscape, physical security of hardware is often not a strong consideration when designing a system. When designing many systems, the physical security is assumed to be in place, and since the majority of attacks are occurring over the network, not much extra attention is given. When designing a PKI, additional consideration must be given to the physical security of the systems, as unauthorized physical access can lead to a complete compromise of the PKI, and subsequently lead to compromise of other critical systems that rely on it.

Designing physical security involves a substantial amount of planning before deploying a PKI because there are many aspects to consider, and there is not a one-size-fits-all solution. For example, the physical security for an organization wanting to deploy a simple [two-tier](#_CA_Hierarchy_Options_1) CA system for solving a single application needed in a low assurance situation is handled differently than the physical security for a PKI for a financial institution wishing to use their CAs for securing transactions.

The following information is not intended to be a checklist or “how-to” guide for building physical security for PKI. Instead, the concepts below should each be given consideration when implementing physical security controls for a PKI.

## Functional Considerations

The level of physical security a PKI requires depends on the functions it allows. Consider the following when defining physical security requirements for PKI:

* **Assurance Level** – The level of trust stated by the entity providing certificate services based on many different factors including the stringency of the method used to identify a person or entity receiving certificates, the criteria required for issuance, and the purpose of the certificates issued.
* **Function of the system** – The function or functions a system has in relation to the PKI. Functions include acting as a CA (online or offline), enrollment functions, and revocation verification hosting. Different functions may require different levels of physical security.
* **Form Factor** – The server operating system host type - virtual or physical.
* **Private Key Storage** – The method by which the private keys are stored whether it is on a CA Server file system or a Hardware Security Module (HSM) which is a dedicated hardware device for this purpose.
* **PKI Artifact Storage** – Storage of dependent components including HSM cards or tokens, backup drives, USB drives, smart card readers, or biometric devices. See [Protecting CA Keys and Critical Artifacts](#_Protecting_CA_Keys_1) for more information. Strong physical access controls should be in place for any sensitive PKI artifacts.
* **Business Continuity and Disaster Recovery** – Processes and procedures created to ensure a PKI is functionally available after a minimal amount of downtime after an event causing disruption of service.

## Operational Considerations

In addition to functional considerations, consider the following operational aspects when designing physical security for PKI. These include but are not limited to:

* **Environmental** – Ensure the availability of sufficient electrical power to operate servers, heating, ventilation, and air conditioning, logical security controls, and surveillance. Also ensure the ability to transition to secondary or facility-generated power in case of primary power source outage in primary or backup data center facilities.
* **Geographic Location** – Consider the location of primary and backup data center facilities and the associated risks due to climate, power sources, available area-based workforce, and other geopolitical considerations.
* **Structure Hardening** – Place controls in primary and backup data centers to mitigate the risk of unauthorized human entry as well as unwanted animal breach in addition to weather or environmental-related risks due to floods, tornadoes, earthquakes, or hurricanes. If necessary, place controls to prevent terror or wartime-related structural compromise.
* **Interior Climate Control** – Place temperature and humidity consistency controls to prevent overheating of the servers, condensation, and static electricity.
* **Asset and Personnel Safety Measures** – Ensure that appropriate heat and flame prevention and extinguishment are present. Also, ensure that safety of the personnel in the event of an emergency is accounted for in the design.

## Designing Physical Security

Most data centers already have some physical security in their design or implementation. Typically they have doors requiring a proximity card for access with another form of verification such as a PIN pad, biometric scanners, or a security guard checking identification. Most data centers have some form of closed-circuit surveillance inside and out. Some have dedicated cages for servers identified as high-value assets. Some or all of these controls do not necessarily mean a PKI contained inside is secured.

Not all organizations have the luxury of designing their data center with PKI in mind. However, organizations with existing data centers with some of these controls in place should take advantage of their placement and use the existing controls to help mitigate risk. The following recommendations should be considered when designing PKI physical security.

### Track and Audit Physical Access Requests

Consider implementing processes that allow all access requests to sensitive areas be tracked and have an audit trail. Consider having all access to a sensitive area require an approval workflow be completed prior to access being granted. For example, a process could be defined that requires data center operations staff to review all requests for access and ensure they are originating from known individuals with a legitimate need for access prior to temporary access being granted. Using an access approval process is a preventive control that allows for additional verification, where personnel with persistent access may misuse their credentials and access may not be detected until after the fact.

Periodically audit physical access to sensitive areas to ensure no unapproved access has occurred. Consider comparing physical access logs with known work orders that were executed and verifying that only trusted personnel were allowed into the sensitive area.

### Consider Using Biometrics

Proximity access cards and keys can be easily stolen. Identification cards can be counterfeited. It is far more difficult to fake a person’s biometric signatures such as hand geometry, fingerprints, or retina data. Consider using biometric data as an authentication mechanism to access building areas where sensitive PKI assets are stored.

A key aspect of using biometrics is ensuring identity verification is performed during biometric enrollment, especially when the data is paired with a device such as a proximity card. In addition, it is important to ensure any biometric enforcement controls are configured to reject two persons with identical data in biometric enrollment. A person could conceivably have a valid stolen proximity card and pair it with their own biometric data to access an area as the owner of the stolen card.

### Use Multi Person Control

For sensitive areas where PKI assets are stored, consider requiring multi person control to enter the area. Multi person control ensures that no single person can gain access to sensitive assets. This prevents a malicious insider from acting alone, forcing them to collude with another trusted individual to gain unauthorized access. In the case of physical breaches, it may be more likely that the breach is performed by an insider rather than an external attacker. With multi person control in place, it is more difficult for an attacker to obtain multiple credentials, or put multiple trusted people under duress to perform an action against their will.

Consider implementing technical controls that enforce multi person access, ideally with representation by persons in differing roles or organizations. Examples include door readers that require multiple distinct persons to present keys prior to allowing access, alarms that require two codes to disable, or safes that require multiple combinations to unlock. Further enforcement mechanisms of multi person control are discussed in the [Protecting CA Keys](#_Protecting_CA_Keys_3) section.

### Eliminate Tailgating to Sensitive Areas

Access to sensitive areas should be limited only to authorized personnel. “Tailgating”, or allowing a user to enter behind another user based on their access should be prohibited. Consider implementing a man trap that only allows one user to enter the sensitive area at a time. Alternatively, a guard can control access by authenticating each individual as they enter the sensitive area. In cases where visitors need to access a sensitive area where they do not have access, consider implementing an override procedure where multiple authorized persons are required to override the system and allow the visitor to have access.

### Use Alarm Systems as a Detective Control

Consider implementing alarm systems to detect unauthorized access to sensitive assets. For example, a server rack can be configured to trigger an alarm if it is opened without prior knowledge of the facility security team. Alarms can be used as a control to ensure that all access attempts come to the attention of the facility security team. For example, if the facility security team is required to disable the alarm prior to access, this would alert them that work is being performed and to be on heightened awareness since there are people in the sensitive area.

### Use Cameras as a Detective Control

Consider using surveillance cameras in sensitive areas where access is limited to ensure that unauthorized access is detected. When cameras are used, ensure that they are placed such that they capture all entry/exit to the secure location and have a good view of the sensitive assets. Ensure that there are processes in place to ensure that access events are reviewed in a timely manner and that recordings are stored securely. According to policy in many organizations, key ceremonies are recorded. If you are able to utilize surveillance cameras to record or design the data center to allow line-of-sight surveillance to view artifact extraction, handling, storage, and inventory or even show screen access or use, an additional camera may not need to be used.

### Geographically Separate Primary and Backup Sites

In case of primary site failure due to an act of nature such as an earthquake, hurricane, or flood, as well as acts of terror or war, the backup location should be geographically distinct and not susceptible to the same acts of nature or terror-related damage as the primary site. Additionally, separate staff should be available for primary and backup facilities. Where backup facilities are hosting sensitive hardware such as active HSMs, ensure that the physical controls present are equivalent to the controls at the primary location, and meet the defined corporate policy.

### Use Security by Obscurity Carefully

Security through obscurity can be used to your advantage or to your detriment. Do not place CAs alongside servers administered by an organization different from the organization responsible for PKI. However, it can be to your advantage to discreetly name or tag systems to not immediately disclose the purpose or criticality of the system. Also, disseminate information regarding sensitive assets (location, purpose, etc.) on a need to know basis. Refrain from tracking information on the company intranet that would make it easy for an attacker to know the exact location of sensitive equipment.

## Conclusion

Physical security controls help to provide assurance that the PKI will not be compromised by attacks requiring physical proximity to the PKI systems. Controls can help mitigate impersonation of authorized users, hardware attacks, and introduction of unauthorized software into offline environments. Implementing strong physical controls can also help mitigate risks associated with insider attacks including rogue or administrators under duress performing unauthorized actions.

For a complete list of the recommendations for planning physical controls for securing PKI, along with the level of [business impact](#_Determining_the_Level) at which you should consider implementing them, refer to [Appendix F: List of Recommendations by Impact Level](#_Physical_Security).

# PKI Process Security

As mentioned above in the [Introduction to PKI](#_Introduction_to_PKI) section, maintaining the trust of relying parties is an integral component to managing a PKI. If the PKI has no formal policy defined, the relying parties cannot make an informed decision whether to trust certificates issued by the CAs. This is especially important when certificates are distributed to external third parties. Otherwise, there is no clear understanding of how the PKI is managed.

To facilitate this trust, you should deploy and operate a PKI with some level of oversight, and with policies, standards, and procedures in place to manage the PKI. The level of oversight and the number of controls required will vary depending on the intended use and security impact of issued certificates. For example, a CA trusted and used internally to issue certificates for wireless network authentication does not need the same amount of oversight as a CA that issues SSL certificates to customers. In both cases, define and set the security bar for the PKI from the start.

## PKI Policy

Prior to deploying any CA or issuing a certificate, define and agree upon the policy which governs the use of the PKI. A policy usually takes into consideration regulatory and industry requirements as well as unique requirements for your company. The policy may also specify technical aspects of the PKI such as the cryptographic algorithms that must be used as well as operational controls for the CAs.

It is likely that other services either inside or outside of your environment will depend on the PKI, and the PKI policy should provide a clear guidance on what can be expected for certificate issuance, security, disaster recovery, etc. The policy does not need to be overly complex, but it is critical to develop and follow it from the beginning to have a level of assurance in the operated PKI.

In addition to PKI policy, you may need to develop CA-specific policies before implementing the PKI. These policies may be expressed differently depending on the required level of assurance. They can be expressed either as documented statements about certificate usage and issuance controls for a simple internal CA or as a formal Certificate Policy/Certification Practice Statement that follow IETF Public Key Infrastructure X.509 Certificate Policy and Certification Practices Framework ([RFC 3647](http://tools.ietf.org/html/rfc3647)) with accompanying standard operating procedures.

### Certificate Policy

As defined by IETF, a Certificate Policy (CP) is “a named set of rules that indicates the applicability of a certificate to a particular community and/or class of application with common security requirements.” That is, a CP defines the expectations and requirements of the relying party community that will trust the certificates issued by its CAs. For example, the CP explains what methods are used to establish the identity of a subject before issuing a certificate. A Certification Practice Statement (CPS) outlines the operation of the PKI from a security perspective and must be followed from the initial deployment of a company’s CAs. These formal documents may be overkill for a typical enterprise PKI deployment, but they provide a good structure to think about for your PKI and risk assessment.

It is quite common to delay creating a CP/CPS until after some CAs are already deployed in the environment. However, it is recommended to not begin server deployment or even CA hierarchy design without clear definition of your PKI policy based on your own risk assessment and with approval from all affected parties within the company. Consider getting input from both the legal and audit departments while defining the policy if it affects external parties. The existence of a comprehensive and accurate policy and certification practices is critical in creating a reliable PKI.

### Certification Practice Statement

The CPS sets the security standard for the PKI solution and is used as the source for the PKI security requirements. Note that you may choose to not create this formal document, but it is quite useful to follow the CPS structure to develop and document your own policy. A CPS is important when certificate subscribers exchange or use certificates with partners and entities outside of the company's network. When external trust is implemented, you need to align PKI policies and practices as part of external contract terms and a CPS is a good way of achieving this.

The CPS basically translates CPs into operational procedures on the CA level. The CP focuses on the make-up and uses of a certificate; the CPS focuses on a CA.

It is important to make conscious decisions about what practices will be used for certificates. You should include both the decision and reasoning behind it in the documentation. By doing this, you are able to state what your practices are so that these practices can be tested and evaluated in production. When setting up monitoring, use this documentation to specify what should trigger an alert and what is acceptable per certificate issuance best practices.

Effective key management controls and practices expressed in the CPS or similar documentation within your company are essential to the trustworthiness of the PKI. These practices should cover CA key generation, CA key storage, backup and recovery, CA public key distribution, CA key usage, CA key destruction, CA key backup, and the management of CA cryptographic hardware through its life cycle.

The certificate lifecycle is at the core of the services provided by the CA. The user certificate lifecycle usually includes the following:

* Registration (the identification and authentication process related to binding the End Entity to the issued certificate)
* Renewal/rekey of certificates
* Revocation of certificates
* Timely publication of certificate status information

Effective controls over the registration process are essential because poor identification and authentication controls jeopardize the ability of clients and relying parties to rely on the certificates issued by the CA. Effective revocation procedures and timely publication of certificate status information are also critical elements because it is critical for relying parties to know when they are unable to rely on certificates that have been issued by the CA.

The establishment and maintenance of a trustworthy CA environment is essential to the security and reliability of the CA’s processes. The CPS (or similar documentation) must describe how logical and physical access to the PKI and data is restricted to authorized individuals, how the continuity of key and certificate management operations is maintained, and how CAs development, maintenance and operations are properly authorized and performed to maintain PKI integrity. Without strong CA environmental controls, strong key and certificate life cycle management controls are severely diminished in value. CA environmental controls must include:

* PKI policy management
* Security policy management
* Security management
* Asset classification and management
* Personnel security
* Physical and environmental security of the PKI facilities
* Operations management
* System access management
* Systems development and maintenance
* Business continuity management
* Monitoring and compliance
* Auditing

## PKI Governance and Oversight

This section is applicable to big enterprises or government organizations because of the complexity of their environments. A common error for organizations during PKI solutions deployment is the lack of PKI governance or oversight. The goal of governance is to ensure the consistent management of the policy including its formulation, the applicability of policy, and measurement of compliance with policy. Governance plays a significant role in a successful PKI because a PKI is not a static system. There will likely be ongoing changes within the environment that will drive operational or security changes to the PKI. For example, ongoing risk assessments will uncover new risks which the governing body of the PKI should take into account when developing the policy. The governance structure for the PKI policy is usually known as the Policy Authority. The Policy Authority is responsible for identifying the appropriate set of requirements for a given community and oversees the CAs that issue certificates for that community. Proper governance will ensure that any changes introduced are well understood, carefully considered, and are well communicated to the community of relying parties. The following are some recommendations s for ensuring proper governance and oversight:

* The Policy Authority should use existing policy structure
* If a policy team exists already, get them involved
* Whenever a PKI may affect external customers or partners, involve the legal department in the work of the Policy Authority
* Formalize the work that the Policy Authority does. For example, if members of the Policy Authority approve changes, ensure that the approval process is documented and tracked. Take formal votes and keep meeting notes.
* Establish regular meetings to review and update the PKI policy

## Roles and Responsibilities

PKI process security relies on trustworthy personnel to deploy and operate the PKI. Personnel security plays a critical role in the PKI’s overall security. Design personnel security to prevent both unauthorized access to the PKI facility and CAs and compromise of sensitive CA operations by CA personnel. Inadequate personnel security procedures or negligent enforcement of personnel security policies can pose potentially serious threats to security. These threats can include unauthorized access, data loss and corruption, denial of service, and even facility sabotage or terrorism. Such events can erode or destroy confidence in the PKI.

While planning for CA deployment or operations, clearly define and assign individuals to trusted roles. A trusted role is one whose incumbent performs functions that can introduce security problems if not carried out properly, whether accidentally or maliciously. It is essential that the people selected to fill trusted roles be held accountable to perform designated actions correctly because if they fail to do so, the integrity of the CA and the PKI is weakened. The functions performed in these roles form the basis of trust in the CA. There are two approaches to take in order to increase the likelihood that these functions can be successfully carried out. The first approach is to minimize the number of trusted roles and ensure that the people filling those roles are trustworthy and properly trained. The second is to enforce the concept of least privilege and distribute the functions of the roles among several people, so that any malicious activity requires collusion (multi-party control).

Trusted roles include the following responsibilities:

* Overall responsibility for administering the implementation of the CA’s security practices
* Approval of the generation, revocation, and suspension of certificates
* Installation, configuration, and maintenance of servers that operate the CA
* Day-to-day operation of the servers
* CA backup and recovery
* Maintenance and review of audit records
* Cryptographic key life cycle management functions (for example, key component custodians)
* Development and validation of the CA

The mandatory trusted roles usually defined are the CA administrator, certificate manager (or security officer), the CA operations staff, and security auditors. Multiple people may hold the same trusted role with collective privileges sufficient to fill the role. Maintain lists for those who act in trusted roles and define the number of persons required per task. When multi-party control is required, all participants shall hold a trusted role. Do not achieve multi-party control using personnel that serve in an auditor role with the exception of audit functions.

The following tasks usually require two or more persons:

* Generation, activation, and backup of CA keys
* Performance of CA administration or maintenance tasks
* Archiving or deleting CA audit logs. At least one of the participants must be in an auditor role.
* Physical access to CA equipment
* Access to any copy of the CA cryptographic module
* Processing of third party key recovery requests

Some roles require separation of duties. For example, individuals serving as auditors shall not perform or hold any other trusted role. Therefore, only an auditor may perform internal auditing functions, with the exception of those security audit functions (configuring, archiving, deleting) that require multi-person control. Enforce roles separation by requiring that an individual who performs any trusted role have only one identity when accessing CA equipment. The way to achieve this in AD CS is described in the [Role Separation](#_Role_Separation) section.

Personnel considered to fulfill trusted roles should present some proof of the requisite background, qualifications, and experience needed to perform their prospective job responsibilities competently and satisfactorily. In some cases it might be wise to ask persons fulfilling trusted roles to pass a comprehensive background check (in accordance with local privacy laws) and ensure that they periodically undergo background checks.

All personnel performing duties with respect to the operation of CAs should receive training to perform their duties. This training could be formal or informal, and is like the training for other IT systems. In some higher security environments, it is beneficial to formalize the training and track completion prior to granting access. Training should cover the following topics:

* PKI security principles and mechanisms
* All PKI software versions in use on the system
* All PKI duties they are expected to perform
* Disaster recovery
* Business continuity procedures
* Stipulations of established PKI policy

## Key Generation Ceremonies

As mentioned previously, secure CA key generation is essential to the trustworthiness of the CA and PKI. An important portion of secure CA key generation is to ensure that CA key pairs are generated in accordance with the PKI’s established policy. This is achieved by following predefined procedures specified within detailed key generation ceremony scripts. As mentioned before, plan the key generation ceremony in advance and include relevant parties in its approval.

The CA key pair generation must create a verifiable audit trail demonstrating that the security requirements for procedures were followed. The ceremony must be documented in sufficient detail to show appropriate multi-party control and role separation were used. Depending on the established security policy, generation of CA keys should be witnessed by an independent party and/or videotaped and all CA key generation activities must be logged.

Develop a ceremony document that contains a list of the materials used, the trusted roles and responsibilities, the ceremony roles list, and a detailed step-by-step script for HSM setup, CA deployment, and CA configuration. The ceremony script must contain fields for signatures of operator and witness, and be specific on produced ceremony artifacts. Below is an example of a ceremony script:

| **Step** | **Description** | **Operator Initials** | **Witness Initials** |
| --- | --- | --- | --- |
| 1 | Prepare Hardware Resources  Confirm that the appropriate hardware resources are present and inspect each for tampering:   * Private network switch * Monitor, Keyboard, Mouse * HSM * Transport media * Tamper-evident security bags   Confirm the hardware is not connected to any external systems or network. |  |  |
| 2 | Server Configurations |  |  |
| 3 | Pre-Generation Configurations |  |  |
| 4 | CA Installation   * Connect to HSM * Begin CA installation * Copy artifacts to the transport media |  |  |
| 5 | Request CA Certificate |  |  |
| 6 | Install CA Certificate |  |  |
| 7 | Post-Issuance Configurations |  |  |
| 8 | Backing Up server |  |  |

**Sample Key Generation Ceremony Script**

## Conclusion

Defining a Certificate Policy/Certification Practice Statement, regardless of how formal you choose to be, provides a baseline expectation of what relying parties can expect from your service. Creation of these documents also act as a forcing function to ensure that all aspects of your PKI management have been given some consideration. Enforcing the concept of trusted roles helps ensure those operating the PKI meet a minimum standard of training and experience prior to being granted access to high value assets. Performing key signing ceremonies provide assurance that proper steps were followed, which mitigates risks associated with keys being created or managed insecurely.

For a complete list of the recommendations for creating PKI processes and documentation, along with what level of [business impact](#_Determining_the_Level) at which you should consider implementing them, refer to [Appendix F: List of Recommendations by Impact Level](#_PKI_Process_Security).

# Technical Controls for Securing PKI

While a large percentage of the work required to operate a successful PKI is in the creation of the correct policies, standards and procedures, the work required to implement a secure design should not be discounted. This section introduces a number of technical recommendations for implementing a secure design. Implementing strong technical controls will introduce barriers that will make successful exploitation very difficult or cost prohibitive. Many of the recommendations are specific to AD CS-based online PKI deployments, although the concepts are universally applicable. Not all recommendations apply to all environments. Each recommendation is [provided in the appendix](#_Technical_Controls_for) along with the recommended [impact level](#_Determining_the_Level) to which it applies.

## Securing the CA Operating System

The following are recommendations for securing the CA operating system.

### Creating a Baseline Configuration for all CAs and RAs

Critical systems such as CAs should be locked down from the moment they are introduced onto the network. Several freely available tools exist (discussed below) that either ship with Microsoft Windows® or can be downloaded to assist in creating a baseline and then deploying it via Group Policy Objects (GPO) to all domain-joined certification authorities.

### Microsoft Security Compliance Manager

[Microsoft Security Compliance Manager](http://technet.microsoft.com/en-us/library/cc677002.aspx) (SCM) provides comprehensive security baseline recommendations for Microsoft operating systems and server roles. Use SCM to create a detailed baseline that can be deployed and enforced on all domain-joined CAs via GPO.

### Microsoft Security Configuration Wizard

Microsoft Security Configuration Wizard (SCW) is a guide for the process of creating, editing, applying, or rolling back a security policy. In conjunction with SCM, use it to create a baseline configuration that can be applied across other similar servers via GPO. SCW is included with Microsoft Windows Server®. For more information on SCW, refer to the links below:

**Microsoft Windows Server 2008®**: [Security Configuration Wizard](http://technet.microsoft.com/en-us/library/cc771492(v=WS.10).aspx)

**Microsoft Windows Server 2008 R2® and Microsoft Windows Server 2012®**: [Security Configuration Wizard](http://technet.microsoft.com/en-us/library/cc754997.aspx)

### Online CA Hardening Recommendations

Below are several recommendations to consider when creating a secure baseline for an online CA. This list is not complete, and the recommendations provided should be extensively tested before deploying in a production environment.

* Disable CD-ROM Autoplay
* Rename administrator and guest accounts
* Disable local administrator and guest accounts
* Use a distinct password for the local administrator account that is not used on other systems
* Enable the Microsoft Windows® Firewall with Advanced Security and configure it to allow only required traffic. Refer to the [Network Isolation](#_Network_Isolation_1) section for more segregation recommendations.
* Disable services that are not required for the CA to function. SCM contains service recommendations from Microsoft for the CA role.
* Disable [LM and NTLMv1](http://technet.microsoft.com/en-us/library/jj865674(v=ws.10).aspx) authentication protocols
* Only install software that is necessary for the CA to perform its function
* Disable [Direct Memory Access (DMA) devices](http://support.microsoft.com/kb/2516445)
* Disable Remote Desktop Services

### Additional Roles on Certification Authorities

A common misconfiguration Microsoft sees during PKI assessments is an enterprise root or enterprise subordinate CA being run on the same system as a domain controller. Running a CA on the same system where other roles are hosted exposes the CA to a broader attack surface that introduces potential problems with performance and troubleshooting. Additionally, this may introduce issues when attempting to upgrade the environment in the future, as there may be requirements to run different components at different operating system levels. A CA system should run with only those roles and features installed that are required for its operation. Another common role installed on a CA is Internet Information Services (IIS) to support the CA web enrollment pages. Do not install the web enrollment pages or IIS as part of a standard CA install unless there is a known business requirement.

### Alternate Administrative Accounts

Administrators managing the day-to-day operations of the PKI should not use the same accounts used on personal productivity workstations to check email and browse the Internet. Instead, they should use dedicated alternate accounts with the required permissions necessary to manage the PKI.

### Updating Online Certification Authorities

Although it may seem counterintuitive, consider updating CAs and other critical infrastructure components separately from the general Microsoft Windows® infrastructure. If an organization leverages enterprise configuration management software for all computers in the infrastructure, compromise of the systems management software can be used to compromise or destroy all infrastructure components managed by that software. By separating updates and systems management for online certificate authorities from the general system population, the amount of software installed on CAs is reduced, and their management more tightly controlled.

### Internet Access from Certification Authorities

Launching web browsers on CAs should be prohibited not only by policy but by technical controls, and CAs should not be permitted to access the Internet except to validate CRLs. Although detailed configuration instructions are outside the scope of this document, there are a number of controls to implement in order to restrict the misuse or misconfiguration and subsequent compromise of CAs.

### Local Administrators Group Membership

In many organizations the baseline system configuration includes a large number of groups or user accounts included in the local administrators group of the system. With highly secure systems such as CAs, the number of accounts that are members of the local administrators group should be kept to a minimum. In an AD CS deployment, if an attacker gains access to an account with administrative access to the CA, there is a high likelihood they will be able to create certificates that will allow them to gain privileged access to the Active Directory®. For online CAs, consider limiting administrative access to only dedicated accounts used for management of the PKI, and enforce the members of the local administrators group via GPO. Enterprise Admins and Domain Admins can be removed from the local administrators group.

Accounts that are of particular interest to attackers are accounts with wide and/or deep access across an environment. Often these are accounts that perform an important function, such as security scanning, update management, backup, inventory, etc. and require administrative rights to operate. Where possible, remove these accounts from CAs and consider how the function could be performed without requiring administrative rights on the CA. Additionally, eliminate or limit the number of system/service accounts that are permanent members of the local administrators group. Actions that are performed on a CA should be traceable back to the person that performed the action.

### Application Whitelisting

Use AppLocker or a third-party application whitelisting tool to configure services and applications that are permitted to run on CAs. These permitted applications and services should be comprised of only what is required for the computer to host AD CS plus any system security software such as antivirus software. Whitelisting permitted applications on CAs adds an additional layer of security so that even if an unauthorized application is installed, the application cannot run.

A CA is an excellent candidate for AppLocker because the list of software required to run on a CA should be minimal and relatively static. Testing is vital to an AppLocker deployment. Prior to deployment, test a list of rules in a test environment and then migrate the rules to a production server and run using **Audit only** enforcement to profile the CA. Once the rule set is established, enforce the rules. For more information on deploying AppLocker, refer to the [AppLocker Overview](http://technet.microsoft.com/en-us/library/hh831409.aspx).

### Securing Remote Management Tasks

For highly secure CAs, it is very common for remote access to management tasks be very limited or disallowed by policy. Remote management should only originate from authorized users and systems. This can be accomplished through a combination of user rights settings and Microsoft Windows® Firewall with Advanced Security settings. A recommendation to consider when creating a remote access design is to use secure administrative hosts or jump hosts, as described in the [Best Practices for Securing Active Directory](http://aka.ms/bpsad) whitepaper. While the whitepaper discusses in detail the approaches for securing domain controllers, the same strategies can be applied to other highly sensitive systems such as CAs.

Microsoft Windows Server 2012 R2® and Microsoft Windows 8.1® introduce a [new feature](http://blogs.technet.com/b/kfalde/archive/2013/08/14/restricted-admin-mode-for-rdp-in-windows-8-1-2012-r2.aspx) in **mstsc.exe** called “Restricted Admin Mode”. If **mstsc.exe** is started with the **/restrictedAdmin** parameter, the credentials used to authenticate will not be sent to the remote computer, which limits the ability of attackers to steal and reuse credentials. In addition to restricting access via Remote Desktop Protocol (RDP), control access to the CA through other channels as well. If you use physical hardware to host the CA, there is a high likelihood that the hardware contains a Remote Management Board (RMB) that can be used to access the system. Account for access via the RMB and any other channels (Microsoft Windows PowerShell® Remoting, DCOM, SMB, etc.) when designing a firewall policy. In high security deployments, consider disabling RMBs.

After defining the acceptable methods of access, implement the controls using GPOs to apply them consistently. Consider using a dedicated Organizational Unit (OU) in Active Directory® to manage the application of GPOs to PKI systems. Many of the recommendations provided throughout this whitepaper can be applied through the use of GPOs.

### Multi-factor Authentication for Certification Authority Access

A recommendation for implementing a secure design is the implementation of multifactor authentication such as smart cards for online CA access. Smart cards implement hardware-enforced protection of private keys in a public-private key pair, preventing a user’s private key from being accessed or used unless the user presents the proper PIN, passcode, or biometric identifier to the smart card. Even if a user’s PIN or passcode is intercepted by a keystroke logger on a compromised computer, the card must also be physically present for an attacker to reuse the PIN or passcode.

For cases in which long and complex passwords have proven difficult to implement because of user resistance, smart cards provide a mechanism by which users may implement relatively simple PINs or passcodes without the credentials being susceptible to brute force or rainbow table attacks. Smart card PINs are not stored in Active Directory® or in local SAM databases, although credential hashes may still be stored in LSASS protected memory on computers on which smart cards have been used for authentication.

A common misconception when requiring smart cards for interactive access for a CA is that if there is a problem with the PKI used for the smart card certificates, the CA will be inaccessible to resolve the problem because it requires smart card logon. This is untrue because it is possible to continue to use the local administrator account in the case of an emergency. Even if the local administrator account is disabled, the system can still be booted to recovery mode to enable the account, or if GPOs can be edited, the account can be enabled via GPO to perform the tasks required.

## Securing Offline Certification Authorities

For highly secure CAs that issue very few certificates, a strong preventive control is to keep the CA offline. The lack of network connectivity provides a boundary for potential attackers and exploits. The purpose for adding an additional security boundary and removing root and policy CAs from the network is that a compromise of a root CA has broader impact because it can be used to sign additional issuing CAs valid for any use cases and are inherently trusted. Additionally, root and policy CAs typically have very little use that would even require them to be powered on. However, keeping a CA offline introduces some new challenges, such as updating, maintenance and access.

Offline CAs are often one of the most undervalued assets of an organization. If an attacker gains control of an offline CA that subordinates to an enterprise CA in Active Directory®, this could lead to full compromise of the directory by taking advantage of the inherited trust relationship. If an attacker gains control of an offline CA that subordinates to a CA used to issue certificates for financial transactions, intellectual property, or critical communication between partner organizations, this could jeopardize the business partnership or lead to regulatory penalties. Consider the following recommendations when designing and managing offline certification authorities.

### Protect CA Private Keys

The most important logical piece of data is the CA private key. Every time a CA performs a signing of a certificate or CRL, the CA private key is being used. If the CA private key were compromised, the attacker could perform operations as the CA, undermining any other security controls. More detail regarding protecting CA private keys can be found in [Protecting CA Keys and Critical Artifacts](#_Protecting_CA_Keys_2).

### Offline CAs Should Be Truly Offline

Offline CAs are called offline for a reason - the absence of network connectivity provides a boundary for potential attackers and exploits. They are only accessible physically and never connect to a network. If the offline CA is installed on a physical server chassis, a network cable should never be plugged into the server. Ideally the server would be built without a network card, or the network card disconnected from the motherboard, disabled in the BIOS, or at minimum disabled logically in the operating system. Offline CAs can also be virtualized; refer to the [Virtualizing Certification Authorities](#_Virtualizing_Certification_Authorit_1) section for more information.

Regardless if the CA is physical or virtual, when an offline CA is not in use, the systems and dependent components should be shut down completely. This includes host computers for virtualized offline CAs and HSMs.

### Managing Data Transfer

With an offline CA, typical data transfer techniques such as file shares are not available. However, data will still need to be transferred to and from the system periodically. It is essential to scan USB or other transfer media for malware and only use authorized devices for file transfer or updating the server. Consider using a dedicated USB, SD card, or other removable media to transfer data to and from the offline system.

### Updating Offline Certification Authorities

With strong processes in place to control the data introduced to the system, monthly security updates could be considered optional. For offline CAs, consider updating the operating system with service packs and any updates that affect the logical operation of the CA. This includes CA software updates, and updates for changes to time zone boundaries or Daylight Savings adjustments. Additional updates may be necessary to ensure supportability in case of a functional problem or for compliance reasons. If an HSM is used, ensure that updates to the HSM software and hardware are applied as appropriate. HSM vendors will provide updates that address security issues as well as additional functionality.

### Account Management

If you need built-in auditing capacity for tracking purposes, it may be necessary to create and assign separate local accounts for administration. However, if accessing the CA is protected with entry auditing and surveillance, extra accounts may not be necessary and the standard built-in administrator account can be used. In either case, it is recommended that any activity performed on an offline CA can be attributed back to the individual who performed the activity. If an HSM is not used, additional care is needed for administrative accounts as discussed in the [Protecting CA Keys](#_Protecting_CA_Keys_3) section.

## Virtualizing Certification Authorities

Virtualization of online or offline CAs may make sense in some scenarios. Virtualizing an AD CS CA in a Microsoft virtual server environment is a supported configuration. Refer to the [Microsoft Virtual Server support policy](http://support.microsoft.com/kb/897613) for more information.

This section provides guidance for securely implementing offline or online CAs using virtualization technology.

### Offline Certification Authorities

Before virtualizing offline CAs (root, policy), consider the following recommendations:

* **Decouple the CA from the Host Hardware** – Use removable media to store the virtual machine hard disk files, regardless of the host hardware. Doing so permits independence of any host hardware, preventing problems if the hardware fails or needs to be replaced. Use a removable USB attached hard drive to store the virtual machine hard disk files. Securely store the removable media. Access to the CA hard disk files is equivalent to access to the CA.
* **Use a Secure Host Machine** – The host hardware used to create the virtual machine and bring it online for routine maintenance should be dedicated hardware used for this purpose. Physically secure this hardware in the same manner as a physical Root CA. Use server hardware, or laptop hardware. If you use server hardware, keep the hardware locked in a secure cage under the same controls as other offline CAs. If you use laptop hardware, keep the laptop locked in a safe, handled similarly to storage of backup data or other items discussed in the [Artifact Storage and Chain of Custody](#_Artifact_Protection_and) section. If dedicated hardware is not a feasible option, consider building a clean computer not connected to the network when powering on an offline CA. In that case, build the computer, perform the needed activities, and then wipe the computer securely. In any case, do not attach the host hardware to the network. The CA should not have any portion of the infrastructure connected to the network, not even to use SAN attached storage. Remember, offline continues to mean offline. Periodically evaluate the condition of the hardware and replace it as needed.
* **Use an HSM** – By using an HSM, even if a virtual disk image is compromised, the keys are not. By not using an HSM, the CA keys are compromised if the virtual machine disk image ever leaves your immediate control. If there is already a network HSM for the online CAs, it may be possible to utilize it for offline infrastructure as long as you never connect the CA to any routable network. Many network HSMs come with multiple network ports, so it is possible to connect the host computer to the HSM using a crossover cable. More details on utilizing HSMs is explained in the [Protecting CA Keys and Critical Artifacts](#_Protecting_CA_Keys_2) section below.
* **Securely Build the Virtual Machine** – When creating the virtual machine for the offline CA, use the same controls that are used for a physical offline CA. Do not build the Virtual Machine (VM) on another computer connected to the network and then migrate it to a more secure computer. At no time during the build or maintenance process should you expose an offline CA virtual machine to a network attached host. Follow the same secure processes used to build a physical offline CA, including multi person control and building in a secure environment.
* **Perform Regular Backups and Updates** – When performing regular maintenance on the CA, you should make a backup of the virtual hard disk files. Make multiple copies, ensure that they work, and securely store them to offsite facilities. Store the backup copies with the same precautions and controls as the primary copy. As part of the backup, include the virtualization software needed to bring the CA online so that a support team has all the data needed to use the backup and recover the CA. Ensure that the virtualization formats used (e.g. VHD/VHDX, etc.) are kept up to date to reduce reliance on out-of-support software or configurations.
* **Verify the System Time** –Prior to performing any operations with an offline CA image, verify that the system clock is correct. Inaccurate time will cause certificates and CRLs to potentially be invalid and cause service interruptions.
* **Disable Remote Management Capabilities on the Host** – If the host computer has a Remote Management (RMB) board, ensure that it is disconnected from the network.

### Online Certification Authorities

When considering using virtualization for online CAs, consider the following recommendations:

* **Control Administrative Access to the Host Operating System** – Users and processes that have administrative access to the host operating system ultimately have access to the virtual machine hard disk files, which can result in the equivalent of physical access to the virtual CAs. If an administrator should not have access to a physical CA, they should not have access to the host operating system on which the virtualized CA is running. When determining access controls, consider all potential paths that might allow a user to gain access to the CA hard disk files and administrative access to the host OS. Consider also the access to virtual machine management consoles, access to snapshots, access to back end storage arrays, etc. In high security environments, consider disabling RMB access to the host hardware if multi- person control is a requirement.
* **Use Network Attached HSMs** – By using an HSM, even if a disk image is compromised, the keys are not. By not using an HSM, the CA keys are compromised if the virtual machine disk image ever gets out of your control. When using network attached HSMs, consider deploying redundant HSM hardware, because the HSM now becomes a single point of failure affecting multiple CAs relying on its services.
* **Use CA Database Backups** – Virtual machine snapshots can provide an instant recovery to a known good state. Snapshots will restore the CA database to the state it was in when the snapshot was taken, similar to a CA backup. If any certificates have been issued since, they will not be in the database. If certificates have been revoked, they will no longer show as revoked and would be removed from the CRL at the next publication. For virtualized CAs, continue to take regular CA database backups so if a snapshot is used, you also have a clean backup of the CA database in the case of corruption or unforeseen issues.

## Delegating PKI Tasks

Managing an Active Directory®-based AD CS CA deployment requires account permissions for a number of common activities. Generally speaking, the activities can be broken down into two major categories:

**Certification Authority Management**

These are infrequent configuration tasks that you may only perform once or a few times over the life of a CA:

* Installation of a new CA
* Renewal of a CA certificate
* Managing Active Directory® certificate containers (NTAuth, CAs, AIA, CDP, etc.)

**Certificate Management**

These are more frequent operational tasks that regularly occur over the life of a CA:

* Creating and managing certificate templates (new templates, updated enrollment permissions, etc.)
* Certificate lifecycle management (issuance, revocation, renewal)
* Maintenance of the CA (software or security updates, backups, etc.)
* Managing and verifying CRL and OCSP availability

By default, after installing a CA, ongoing operations require at least the occasional use of an account in the domain administrators or enterprise administrators groups. In some organizations it may be desirable to delegate the rights required to perform common PKI tasks to a separate group. It may also be desirable to delegate the rights required to perform infrequent operations to a separate team or set of accounts. If an organization operates a large number of CAs or the organizational structure is such that it makes sense to delegate these rights, refer to [Appendix C: Delegating Active Directory PKI Permissions](#_Appendix_C:_Delegating) for details on what permissions must be delegated.

## Role Separation

An AD CS CA offers the option to enforce Common Criteria (CC) role separation, which is used to separate CA support into predefined CA roles. Each role is eligible to perform a specific subset of CA functionality. Users can be assigned to only one role, and if they are assigned to more than one role, they are unable to perform any CA-related activities. The table below describes the different roles available that are subject to role separation:

| Roles and groups | Security Permission | Description |
| --- | --- | --- |
| CA administrator | Manage CA | Corresponds with CC *Administrator* role.  Configure and maintain the CA. This is a CA role and includes the ability to assign all other CA roles and renew the CA certificate. These permissions are assigned by using the CA snap-in. |
| Certificate manager | Issue and Manage Certificates | Corresponds with CC *Officer* role.  Approve certificate enrollment and revocation requests. This is a CA role. This role is sometimes referred to as CA officer. These permissions are assigned by using the CA snap-in. |
| Backup operator | Back up file and directories  Restore file and directories | Corresponds with CC *Operator* role.  Perform system backup and recovery. Backup is an operating system feature. |
| Auditor | Manage auditing and security log | Corresponds with CC *Auditor* role.  Configure, view, and maintain audit logs. Auditing is an operating system feature. Auditor is an operating system role. |

Role separation offers some benefits, but it also introduces some challenges that should be considered when evaluating its usefulness for your environment. Separating roles allows for a stronger separation of responsibilities for individuals or teams, and can provide a clear technical separation for systems that are subject to compliance requirements that require separation of duties. However, implementing role separation does require a sizable support staff. To ensure that there is adequate coverage for all critical roles, an organization would require multiple individuals for each role, which is often not possible.

Give careful consideration to the operational impact enabling role separation may have before enabling it in your environment. If you use role separation, ensure that its configuration is monitored for changes, as it can be easily disabled by someone with administrative rights on the CA. Refer to the [Monitoring Public Key Infrastructure](#_Monitoring_Public_Key_2) section for more information. For more details on role separation refer to the following resources:

[Implement Role-Based Administration](http://technet.microsoft.com/en-us/library/cc732590.aspx)

[Role Separation](http://technet.microsoft.com/en-us/library/cc773161(v=WS.10).aspx)

[Defining PKI Management and Delegation](http://technet.microsoft.com/en-us/library/cc755614(v=WS.10).aspx)

## Protecting CA Backups

When performing a backup of a CA, there are three items necessary to fully recover:

1. CA certificate(s) and private key(s)
2. CA registry information
3. CA database backup

Several options exist for backing up a CA. If you are using an HSM, consult the HSM vendor documentation for details on what is required to back up and restore HSM protected keys. CA backup options include:

* Perform a system state backup, which will include the CA database, registry settings, and CA key information (including the private key if you are not using an HSM)
* Manually back up the CA using the CA snap-in. This does not include the CA registry or any files required to restore HSM protected keys
* Use certutil.exe or Windows PowerShell® [CA Backup and Restore Windows PowerShell cmdlets](http://technet.microsoft.com/en-us/library/dn535774.aspx) to create a regularly scheduled backup script that backs up the database, registry settings, and required private key files

A common issue Microsoft finds in many PKI assessments is that once a backup of a CA is taken, the same level of protection is not always provided to the backup that exists on the CA. If you are not utilizing an HSM and you are performing regular backups that include the private key, the private key and certificate are stored in a PKCS#12 (PFX) file. If an attacker is able to gain access to the PKCS#12 file, they have the opportunity to brute force the password on the file and gain access to the CA key. If the password can be cracked, the attacker has compromised your PKI and can create certificates of their choosing. The same applies when performing system state backups. If an attacker gains access to a system state backup, they can restore it and gain access to the private key(s). When designing a backup strategy for the CA, consider the following recommendations:

* If an HSM is not used, perform a separate backup of the CA key and certificate to a PFX file and [store the file securely](#_Artifact_Protection_and) where it can be retrieved in the event a restore is required. Store the backup in a tamper-evident bag and place it in a safe with limited access, where the access is monitored and audited. When performing regularly scheduled backups, do not include the CA key as part of the backup. Continue to take new backups as CA certificates are renewed or new keys are generated and set a strong passphrase on the PFX file. To initiate a backup that includes only the CA database and not the CA key, use **certutil –backupdb** rather than **certutil –backup** if initiating the backup from the command line or a script. Beginning with Windows Server 2012 R2®, a backup can be performed with the PowerShell cmdlet **Backup-CARoleService**. When used with the **–DatabaseOnly** argument, the CA certificate will not be included in the backup.
* If an HSM is used, examine how you back up the files and HSM data required to use the keys. If you have multiple HSMs operating within the same security boundary it may be possible to obtain HSM files and use the CA keys on other computers. Consider backing up any HSM files separately and storing them securely to prevent any unauthorized use of the CA keys from other computers with initialized HSMs.
* Consider backing up the CA to another secure location that interfaces with backup systems rather than having backup systems connect directly to the CA. Backup systems typically have the ability to connect to large numbers of systems in the enterprise, and a compromise of the backup system could then lead to a compromise of the PKI.

**Note:** In Microsoft Windows Server 2008® and Microsoft Windows Server 2008 R2®, private keys were not included in the system state backup. A [hotfix was released](http://support.microsoft.com/kb/2603469) that addressed this issue and private keys are included with the system state backup image if the hotfix is applied.

## Network Isolation

CAs should only be accessible by the users and systems that require access to them. There are many deployment scenarios for a CA and many front end systems that may require access to a CA. Supporting some out-of-box scenarios such as auto enrollment of user or computer certificates requires broad access to the CA from most, if not all, domain joined client computers and users on the internal network. Other deployments, such as deploying with a RA such as Forefront Identity Manager Certificate Management, may only require the RA system to interact directly with the CA.

When developing the security design for PKI, consider the following recommendations:

* Isolate certificate systems away from other systems on the network. If a system does not have a legitimate need to connect to a certificate system, do not allow the connection.
* Implement security “zones” to isolate the certificate systems based on their criticality or relationship to each other
* Only allow the required inbound and outbound connections that are identified as necessary for the CA and supporting systems to function
* If you utilize network attached HSMs, restrict access to those devices to only the CAs or other systems that utilize them
* Restrict management access to originate from a limited set of administrative hosts. Refer to the [Securing Remote Management Tasks](#_Securing_Remote_Management) section for more information.

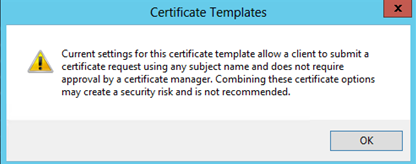
## Securing Certificate Templates

Attackers will take the path of least resistance when attempting to compromise an environment. If a simple attack vector is available, attackers will use it rather than using exploits that are more difficult to execute or more difficult to detect. One method attackers use to compromise environments is to employ misconfigured certificate templates to get credentials that can then be used to access additional systems or sensitive data. The following are several recommendations to consider in order to secure certificate templates:

* **Remove Overly Broad Enroll or Autoenroll Permissions** – Rather than determining the exact population of users that require a specific certificate type and explicitly granting them enrollment permissions, occasionally “Authenticated Users”, “Domain Users” or other broad groups are granted enroll or autoenroll permissions. This can lead to accounts that have no need for a certificate to be eligible to enroll. Avoid granting overly broad enrollment permissions for certificates. Instead, carefully consider which accounts need permissions, and explicitly deny enrollment rights for users or groups of users that should not be eligible for enrollment.
* **Remove Unused Templates from Certification Authorities** – When a Certificate Template no longer has a business need, remove it from any certification authorities that issued it. By default, several templates are included as part of the installation of an enterprise CA. If those templates are not required, they should be removed. Alternatively, on Microsoft Windows Server 2008® and newer you should install the CA with no default templates by including **LoadDefaultTemplates=0** in the **[certsrv\_server]** section of the **CAPolicy.inf** file used during setup of the CA.

Additionally, some high value templates that are issued relatively infrequently do not need to be available on the CA all the time. Examples include Enrollment Agent, Key Recovery Agent, and EFS Data Recovery Agent. Consider removing these templates from issuing CAs except during active use, and monitor for attempts to add these templates to a CA. Refer to the [Monitoring Changes to Certificate Templates](#_Monitoring_Changes_to) section for more information.

* **Secure Templates that Allow You to Specify the Subject in the Request** – Certificate templates provide for two main mechanisms for generating the subject name(s) in a certificate:

1. **Supply in request** – All subject information is provided by the requestor. If you use the default CA policy module, no additional checks are done to confirm the mapping of the subject information to a user account in Active Directory®. If a certificate template is configured to use “supply in request” without additional approvals, the following dialog box displays:  
     
   
2. **Build from this Active Directory® information** – The subject for the certificate is determined by the CA by performing an Active Directory® lookup of the user performing the request. The applicable attributes from the Active Directory® are used to populate the certificate.

When using “supply in request”, a user can request a certificate that would potentially allow them to authenticate as another user if no other security mechanisms are in place. When using “supply in request” templates, ensure that one or more of the following controls are in place to prevent misuse:

* **Implement additional signatures on requests** – A common approach is to require at least one signature from a user or system with an enrollment agent certificate. Consider requiring the enrollment agent private key to be stored in an HSM.
* **Implement certificate manager approval** – Certificate manager approval will set any incoming requests for the template into a pending state, where it must be approved by a person (or system) that holds the “issue and manage certificates” role on the CA before it is issued.
* **Implement monitoring of certificates issued by the template** – Configure monitoring and alert if certificates are issued to VIP user accounts or other accounts that are deemed critical.

## Controlling User Added Subject Alternative Names

An Active Directory® Certificate Services CA offers several methods to add subject alternative names (SANs) to a certificate:

* **Add from known AD object attributes** – The CA can add alternative names from a defined subset of attributes when you choose to add the subject information from Active Directory®. The CA performs this addition, and the data is not specified by the user. Manipulation would require an attacker to be able to manipulate the values of attributes for a user in Active Directory®.
* **Add as an extension in the certificate request** – If the template is configured for “supply in request”, the extensions requested will be honored by the CA if supported. The alternative names are provided by the requestor.
* **Add as an attribute that accompanies the certificate request** – Requires the CA to allow user-specified alternative names via the **EDITF\_ATTRIBUTESUBJECTALTNAME2** flag. If this flag is set on the CA, any request (including when the subject is built from Active Directory®) can have user defined values in the subject alternative name.

Allowing users to define arbitrary alternative names poses risk to the PKI if it is not implemented with proper controls. Anytime you allow a user to define SANs, implement the following additional controls:

* Requests that may contain user-defined alternative names should be set to “pending” when submitted and reviewed by a Certificate Manager prior to issuance
* Do not allow a single person to have the ability to both add SANs and approve the request

It is strongly recommended not to enable the **EDITF\_ATTRIBUTESUBJECALTNAME2 flag** on an enterprise CA. If this is enabled, alternative names are allowed for any Certificate Template issued, regardless of how the subject of the certificate is determined according to the Certificate Template. Using this feature, a malicious user could easily generate a certificate with an alternative name that would allow them to impersonate another user. For example, depending on the issuance requirements, it may be possible for a malicious user to request a new certificate valid for smart card logon and request a SAN which contains the UPN of a different user. Since smart card logon uses UPN mapping by default to map a certificate to a user account, the certificate could be used to log on interactively as a different user, which could be a domain administrator or other VIP account. If this flag is enabled, the CA should be limited to require Certificate Manager approval or limit enrollment permissions to only trusted accounts.

To see if **EDITF\_ATTRIBUTESUBJECALTNAME2** is enabled on the CA, run the following command:

**certutil –getreg policy\EditFlags**

If **EDITF\_ATTRIBUTESUBJECTALTNAME2** is included, it is turned on. To disable the setting, run the following command:

**certutil –setreg policy\EditFlags –EDITF\_ATTRIBUTESUBJECTALTNAME2**

Then restart the CA service:

**net stop certsvc && net start certsvc**

For more information on subject alternative names, refer to [How to Request a Certificate With a Custom Subject Alternative Name](http://technet.microsoft.com/en-us/library/ff625722(WS.10).aspx).

## Conclusion

Implementing strong technical controls can mitigate many of the common attack vectors used to compromise an ADCS PKI installation. This section has detailed some of the common misconfigurations that can lead to compromise, including securing access to certificate templates, and template enrollment options that lead to the issuance of unauthorized credentials. Limiting access to PKI systems through network controls and treating PKI systems as high value assets that are not managed like common infrastructure helps mitigate the risk of the PKI being compromised through supporting systems and overly broad access. Strong key protection help mitigate the threat of a CA key being exported and used outside of authorized hardware by an insider threat or an attacker.

For a complete list of the recommendations for technical controls, along with the level of [business impact](#_Determining_the_Level) at which you should consider implementing them for, refer to [Appendix F: List of Recommendations by Impact Level](#_Technical_Controls_for).

# Planning Certificate Algorithms and Usages

Two key factors in implementing a secure PKI are the choices of cryptographic algorithms used throughout the PKI, and determining what the resulting certificates can be used for. Advances in computing capabilities and cryptographic research continue to show that cryptographic algorithms have a finite lifespan. Selection of proper algorithms will ensure that the data and processes they protect have the best possible chance of remaining secure for their useful life.

## Cryptographic Algorithms, Key Lengths, and Validity Period

The proper selection of cryptographic algorithms and key lengths is essential to the effective use of certificates. The security of information protected by certificates depends on the strength of the keys, the effectiveness of mechanisms and protocols associated with keys, and the protection afforded to the keys. CAs are presented with many choices of cryptographic mechanisms. Inappropriate or inconsistent choices may result in less security for the certificate subscribers. This section provides recommendations for selecting algorithms, key lengths, and certificate validity period for CAs. Note that recommendations provided in this document are current for the publishing date of the document, but you may need to revisit them as computing capabilities and cryptographic research advance.

### Selecting Algorithms and Key Lengths

When designing certificate hierarchy, use only secure cryptographic algorithms and associated key lengths in PKI CAs. Strictly avoid the use of weak cryptographic algorithms (such as MD5) and key lengths. Due to a great deal of attention in cryptography and PKI in recent years, even if you currently employ widely-used cryptographic algorithms (such as RSA/SHA-1 because hash collisions are computationally feasible for MD5 and SHA-1 algorithms which effectively “breaks” them), consider employing new algorithms such as those based on elliptic curve cryptography (ECC).

NIST Special Publication 800-57 [Recommendation for Key Management Part 1 (Revision 3)](http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57_part1_rev3_general.pdf) and ENISA’s [Algorithms, Key Sizes and Parameters Report – 2013 Recommendations](https://www.enisa.europa.eu/activities/identity-and-trust/library/deliverables/algorithms-key-sizes-and-parameters-report) provide detailed recommendations for algorithms, key lengths, and signature schemes. Both documents contain some key lengths comparison for different algorithms and consider 128-bit security level to be the minimum requirement for new systems being deployed. However, take into account the length of time data needs to be kept secure. This is where CA certificate validity period plays its role. The validity period defines how long CA certificates will be trusted because the key length for CA certificates relates to both the security level that needs to be provided and the required duration of the key’s validity. With a longer validity period, plan for a higher security level of crypto algorithms.

With these considerations in mind, the recommended subordinate CAs key length must be at least 2048 bits for RSA and ECC-based subordinate CA keys must use one of the following curves: P-256, P-384, or P-521. For any CA that has certificate expiration more than 15 years in the future, the CA key length that uses RSA must be 4096 bits or greater or, if the CA key uses ECC, the CA key must use either the P-384 or P-521 curve.

The SHA-2 family of hash algorithms is currently the only recommended family of cryptographic hash algorithms. Microsoft recently announced a new [policy](http://social.technet.microsoft.com/wiki/contents/articles/1760.windows-root-certificate-program-technical-requirements.aspx) for CAs that are members of the [Windows Root Certificate Program](http://social.technet.microsoft.com/wiki/contents/articles/3281.introduction-to-the-microsoft-root-certificate-program.aspx) that deprecates the use of the SHA1 algorithm in SSL and code signing certificates in favor of SHA2. The recommendation to discontinue use of SHA-1 is also published as [Security Advisory 2880823](http://blogs.technet.com/b/srd/archive/2013/11/12/security-advisory-2880823-recommendation-to-discontinue-use-of-sha-1.aspx). The recommendation is to ensure that cryptographic keys have a limited lifetime to mitigate the risk of future advances in the capabilities of cryptographic attacks.

To ensure the effective use of certificates, use the following secure certificate signature scheme and hash algorithm combinations:

* RSASSA-PKCS-v1.5 signature scheme as defined in [PKCS #1 RSA Cryptography Standard v2.1](http://www.ietf.org/rfc/rfc3447.txt) with SHA-2 hash algorithms
* RSASSA-PSS signature scheme as defined in [PKCS #1 RSA Cryptography Standard v2.1](http://www.ietf.org/rfc/rfc3447.txt) with SHA-2 hash algorithms (while the RSASSA-PSS signature scheme is considered more secure than the RSASSA-PKCS-v1.5 signature scheme, it is not widely supported)
* The ECDSA signature scheme with SHA-2 hash algorithms

Most PKI deployments today use the RSA/SHA-1 algorithms rather than ECC/SHA-2 due to limited support for elliptic curve cryptography (ECC) by replying parties. Fortunately, Microsoft Windows Vista® and, Microsoft Windows Server 2008® and later versions support advanced cryptographic algorithms including Elliptical Curve Cryptography (ECC) and Secure Hash Algorithm (SHA) version 2. It is important to perform adequate testing to ensure compatibility with relying party applications. Note that with any choice of cryptographic algorithms and key lengths you should pay significant attention to application compatibility and integration testing as application capabilities may differ. Refer to [Common Questions about SHA2 and Windows](http://blogs.technet.com/b/pki/archive/2011/02/08/common-questions-about-sha2-and-windows.aspx) for answers to common questions about support of SHA-2 on Microsoft Windows® operating systems.

### Certificate Validity Periods

During planning and design of your PKI, give consideration to the validity period for each certificate and key in the PKI. When a certificate is checked for expiration, every CA certificate in the chain must be checked. A CA should not issue certificates that have a validity that extends beyond the validity of its own certificate. A Windows Server® CA will not issue certificates beyond the validity of its CA certificate. If you do not plan validity periods correctly, certificate lifetime can become truncated. For example, a certificate that should have a two year validity only has eighteen months because the Issuing CA certificate is due to expire in eighteen months.

As a general rule, the validity period of CA certificates should be at least twice as long as the maximum validity of the certificates issued by the CA. For example, if a CA issues certificates that are valid for two years, the validity period of the CA certificate should be at least four years. A CA certificate is usually renewed in the middle of its lifetime so that it can keep issuing certificates during the full validity period. The recommendation is to renew the CA certificate once while keeping the same key pair and to renew it again while changing the key pair.

Note that special consideration must be given when a CA issues certificates for multiple applications that have different validity periods. For example, an Issuing CA could issue both two year SSL certificates and one year code signing certificates. Each type of certificate will have a different date past which it will not be able to obtain a certificate for the full validity period.

Several reports, such as [ENISA Algorithms, Key Sizes and Parameters Report (2013 recommendations)](https://www.enisa.europa.eu/activities/identity-and-trust/library/deliverables/algorithms-key-sizes-and-parameters-report), [NIST Special Publication 800-57 Part 1 Rev. 3](http://csrc.nist.gov/groups/ST/toolkit/key_management.html), or [BSI Algorithms for Qualified Electronic Signatures](http://www.bundesnetzagentur.de/SharedDocs/Downloads/EN/BNetzA/Areas/ElectronicSignature/PublicationsNotifications/SuitableAlgorithms/2013algokatpdf.pdf?__blob=publicationFile&v=4) provide guidelines for choosing strengths of cryptographic algorithms based on the algorithm security lifetime. For a comparison of the various proposed models, refer [here](http://www.keylength.com/) to define the validity period based on your risk assessment.

Be aware of applications that have long-term PKI use cases. For instance, data stored with its original encryption or documents stored with their original digital signatures may require special processing because all of the keys and certificates originally used will have expired or exceeded their initial period of usefulness, as is the case with code signing. It is important to pay attention to persistence and availability of the certificate revocation information with a goal to enable applications to verify original signature, usually with help of [time stamping](http://msdn.microsoft.com/en-us/library/windows/desktop/bb931395(v=vs.85).aspx). If you use such applications, ensure that those applications can use expired certificates, or manage the storage of their data such that the original signatures and encryptions are not used.

#### End Entity Certificate Expiration

Certificate expiration raises the potential for service outage if a certificate is not replaced before it expires. Starting with Microsoft Windows Server 2012® and Microsoft Windows 8®, certificates in the Computer and User Personal certificate stores (also known as the “My” certificate stores) have [lifecycle notifications](https://social.technet.microsoft.com/wiki/contents/articles/14250.certificate-services-lifecycle-notifications.aspx) generated, including expiration and near-expiration events. The notification feature allows a script or an executable to launch in response to notifications gathered from the Certificate Enrollment API and Microsoft Windows PowerShell®. Administrators can utilize these notifications to help with managing certificate expiration. Events are written to the CertificateServicesClient-Lifecycle-System or CertificateServicesClient-Lifecycle-User logs. The table below lists the expiration-specific events and their sources.

| Event | ID | Level | Detail | Sources |
| --- | --- | --- | --- | --- |
| Close to expiration | 1003 | Warning | •  Template •  Subject names •  EKUs •  Thumbprint •  Store •  Expiration | Autoenrollment, when Renew expired certificates, update pending certificates, and remove revoked certificates is NOT selected. |
| Expired | 1002 | Error | •  Template •  Subject names •  EKUs •  Thumbprint •  Store •  Expiration | Autoenrollment, when Renew expired certificates, update pending certificates, and remove revoked certificates is NOT selected. |

### Mixing Cryptographic Algorithms

A common error when planning to support new cryptographic algorithms is to introduce the new algorithm into the existing certificate hierarchy. For example, you may consider establishing a CA that issues ECC based certificates in an existing CA hierarchy that uses RSA. This will not alleviate all security concerns because the CAs in the certification chain are still vulnerable to potential RSA weaknesses. The same applies to the choice of hash algorithms for issued certificates. The recommendations for cryptographic algorithms are:

* All keys certified within CA hierarchy use the same asymmetric cryptographic algorithm family. For example, a CA that uses an RSA key should not certify a subordinate CA that uses an ECC key.
* The key length of a CA key is the same or longer than the key lengths of the keys certified by the CA
* The strength of the hash algorithm used by a CA to sign certificates is comparable to the strength of the CA key. For example, a CA that has a P384 ECC key should use SHA-384 to sign certificates.
* The strength of the hash algorithm used by a CA to sign certificates is at least as strong as the hash algorithm used by its subordinate CAs. For example, if a CA uses SHA-256 to sign a subordinate CA certificate, then that subordinate CA must not use SHA-384 to sign certificates it issues.

[NIST Special Publication 800-57 “Recommendation for Key Management Part 1 (Revision 3)”](http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57_part1_rev3_general.pdf) provides suggested crypto periods for key types and comparable strengths of crypto algorithms:

| **Bits of security** | **Symmetric Key Algorithms** | **FFC (e.g., DSA, D-H)** | **IFC (e.g., RSA)** | **ECC (e.g., ECDSA)** |
| --- | --- | --- | --- | --- |
| 112 | 3TDEA | L = 2048  N = 224 | k = 2048 | f = 224-255 |
| 128 | AES-128 | L = 3072  N = 256 | k = 3072 | f = 256-383 |
| 192 | AES-192 | L = 7680  N = 384 | k = 7680 | f = 384-511 |
| 256 | AES-256 | L = 15360  N = 512 | k = 15360 | f = 512+ |

While planning a PKI deployment, ensure that the CA hierarchy uses consistent asymmetric cryptographic algorithms and key lengths when issuing certificates. When you decide to move to stronger algorithms, consider building a parallel infrastructure built entirely with new algorithms to which you can migrate.

## CA Key Usages and Certificate Extensions

Most certificates in use today, including those issued by a CA running Microsoft AD CS, conform to the X.509 v3 standard. This standard provides extensible methods to include additional information using certificate extensions. The extensions defined for X.509 v3 certificates provide methods for associating additional attributes with users or public keys and for managing relationships between CAs. For example, a certificate can be marked as valid for usages such as “Client Authentication” or “Smartcard Logon”, or can list a specific policy that applies to the certificate. Applications that accept certificates can then be configured to only accept a certificate if the extensions match what it is expecting.

### Key Usage

The intended scope of usage for a private key is specified through certificate extensions, including the Key Usage and Extended Key Usage (EKU) extensions in the associated certificate. The cryptographic use of a specific key is constrained by the Key Usage extension in X.509 certificates. All certificates should include key usage as a critical extension. Note that you usually should not change key usage in the end-entity certificates, and AD CS CA will take care of setting the right bits in the keyUsage extension.

The following table summarizes key usages for user certificates:

|  |  |  |
| --- | --- | --- |
| User certificates | Signature public keys | digitalSignature bit (0) |
| RSA public keys to use for key transport | keyEncipherment bit (2) |
| ECC public keys to use for key agreement | keyAgreement bit (4) |

Public keys in CA certificates must be used only for signing issued certificates and status information (for example, CRLs). The following table summarizes key usages for CA certificates:

|  |  |  |
| --- | --- | --- |
| CA certificates | Public key to use to verify other certificates | keyCertSign bit (5) |
| Subject public key to use to verify CRLs | cRLSign bit (6) |
| Subject public key to be used to verify Online Certificate Status Protocol (OCSP) responses | digitalSignature bit (0) |

Public keys that are bound into device, applications, and service certificates may be used for digital signature (including authentication), key management, or both. The following table summarizes key usages for device certificates:

|  |  |  |
| --- | --- | --- |
| Device certificates | Public keys to use for digital signatures | digitalSignature bit (0) |
| RSA public keys to use for key transport | keyEncipherment bit (2) |
| ECC public keys to use for key agreement | keyAgreement bit (4) |
| RSA keys to use both for digital signatures and key management | digitalSignature and keyEncipherment bits |
| ECC keys to use both for digital signatures and key management | digitalSignature and keyAgreement bits |

### Extended Key Usages

The other important certificate extension that controls what a certificate is trusted for is the Extended Key Usage (EKU) extension. The Key Usage Extension has an indirect dependency with the EKU extension, so these two extensions need to align. These extensions are usually populated according to [RFC 5280](http://tools.ietf.org/html/rfc5280) and corresponding certificate usage recommendations - for example, in the Transport Layer Security protocol or for smart card logon.

EKU defines what a certificate will be used for and may contain multiple purposes. Defining the types of certificates supported by the PKI by determining EKUs is a critical part of the solution design process. A common misconfiguration is to [leave unnecessary default templates installed](#_Securing_Certificate_Templates) that offer a wide variety of EKUs. You should determine the required capabilities of a certificate before it is issued, and carefully plan their EKUs.

Every PKI-enabled solution should have the type of certificate that matches its trust level and key usage requirements. This could be accomplished by using its own type of certificate that has its own attributes, enrollment process, and target audience. However, it is likely that some solutions can share a certificate for different purposes if the solutions happen to have the same characteristics. For example, a certificate stored on a smart card may be used for workstation logon, VPN access, or for authentication to restricted web sites.

It is important to evaluate whether to use single-purpose or multipurpose certificates. The following table summarizes the benefits of both approaches.

| **Benefits of Single-Purpose Certificates** | **Benefits of Multipurpose Certificates** |
| --- | --- |
| Granular control of certificate usage. Certificates can be individually revoked without affecting other use cases. Certificate usage is limited, so risky scenarios (for example, code signing) are not enabled unintentionally. | Greater usability. Users find multipurpose certificates easy to handle because fewer of them exist. |
| Limited key usage. Different keys are used for different PKI-enabled services. | A smaller accumulated size. Multipurpose certificates are advantageous when hardware tokens have limited storage capabilities. |
| Multiple sources. Certificates can be issued through different CAs when different certificate policies, application policies, or issuing policies apply. | Less network overhead. Less demand is placed on the infrastructure because a lower number of certificates are in use. |
| Archival and Recovery. Certificates that are good for signing and encryption should not be archived. Single purpose certificates allow encryption keys to be archived while allowing signature keys to only exist where they were created. | Less administrative overhead. The smaller number of enrolled certificates reduces the overall management burden placed on the certificate managers. |

The EKUs of each certificate type are typically defined as part of certificate profiles (expressed in the CPS). For AD CS enterprise installations, the properties of each certificate type are configured in certificate templates stored in Active Directory®. Microsoft Windows® operating systems come with built-in templates designed for the most common applications. You can use them as is or as the basis for custom certificate templates. Refer to [Certificate Template Versions and Options](http://social.technet.microsoft.com/wiki/contents/articles/13303.windows-server-2012-certificate-template-versions-and-options.aspx) for additional information.

#### Object Identifiers

EKUs are identified in a certificate by object identifiers (OIDs). An OID is a sequence of integers that uniquely identifies an object, and OIDs are used in the X.509 v3 standard to represent policies, extensions, and attributes in digital certificates. Preferably, the OIDs should be globally unique, especially if the PKI will be used externally.

Quite often customers do not consider OIDs before the PKI deployment starts, but it is not complex to obtain one. If you need to define your own EKU and your PKI will be used across multiple organizations, the EKU needs to have a unique OID assigned either by the Internet Assigned Numbers Authority (IANA) or by a national standards organization. However, if you do not foresee using the OIDs in conjunction with other organizations, the option exists of using private OIDs within the Microsoft IANA-assigned tree. These OIDs are based on the globally unique identifiers (GUIDs) of Active Directory® forests. Such an OID can be obtained by running Microsoft Management Console (MMC) and using the Certificate Template snap-in. In the snap-in, right-click **Certificate Templates**, and then select **View Object Identifiers**. The OIDs are in the 1.3.6.1.4.1.311.21.8.a.b.c.d.1 format, where a.b.c.d is a unique string of numbers based on the AD forest’s GUID.

### Critical Extensions

Marking an extension as critical is a powerful concept because it enforces common understanding of important certificate fields during certificate chain validation. If a PKI-enabled application does not understand the extension marked as critical, it should not process the certificate. This may affect third party applications or browsers that will use issued certificates. Refer to [How Certificates Work](http://technet.microsoft.com/en-us/library/cc776447.aspx) and [How CA Certificates Work](http://technet.microsoft.com/en-us/library/cc737264(v=WS.10).aspx) for additional information. Unless you have a full understanding of how your certificates will be used by relying parties, consider limiting the use of the critical flag in End Entity certificates.

## CA Certificate Constraints

The goal of stating constraints in a CA certificate is to restrict the scope of the CA and limit the impact of a CA’s security breach on other CAs in the certificate hierarchy. [RFC 5280](http://tools.ietf.org/html/rfc5280) defines multiple way to express constraints: basic constraints, name constraints, policy constraints, and EKU. The use of these constraints for qualified subordination is explained in [Planning and Implementing Cross-Certification and Qualified Subordination Using Windows Server 2003](http://technet.microsoft.com/en-us/library/cc787237.aspx).

Unfortunately, support for most of these constraints varies from platform to platform, and it differs between applications (depending on the chain building library used) and kernel-mode/user-mode. These limitations leave only a limited number of practical options to restrict subordinate CA certificates: basic constraints and EKU. By enforcing these CA certificate extensions for all the issuing CAs, you can protect the PKI from an attacker that may use a single compromise of an Issuing CA signing key to issue another subordinate CA certificate in the CA hierarchy and effectively work around controls established in the PKI.

### Basic Constraints

The basic constraints extension identifies whether the subject of the certificate is a CA and the maximum depth of valid certification paths that include this certificate. The path length constraint within a basic constraint for CA certificates specifies how many levels of CAs can be subordinated to a CA. This constraint is used to ensure that issuing CAs will not enroll additional subordinated CAs and that non-CA certificates will not be used to sign other certificates. The path length constraint is specified during CA installation and cannot be changed without reissuing the CA certificate.

The Root CA usually has no defined path length constraint (pathLenConstraint=none) that allows you to extend the CA hierarchy in the future. However, if you have a clear understanding that your issuing CAs will issue certificates only to End Entities, then you should limit issuing CAs by specifying the path length constraint and set this constraint to zero. This prevents the Issuing CA from issuing any CA certificates and will be enforced by PKI clients during certificate chain validation. Without this constraint an attacker may use a single compromise of an Issuing CA signing key to issue another subordinate CA certificate in the CA hierarchy and effectively work around controls established in the PKI.

It is not recommended to use [BasicConstraintsExtension] section in the CAPolicy.inf file to specify the path length constraint for a subordinate CA in AD CS. To add this constraint to a subordinate CA certificate if the parent CA has no PathLength constraint in its own certificate, set the CAPathLength registry value on the parent CA. For example, to issue a subordinate CA certificate with a PathLength constraint of 0, use the following command to configure the parent CA.

**certutil –setreg Policy\CAPathLength 1**

Setting this value causes the CA to behave as though its own certificate had a PathLength constraint of whatever number you specify. Any subordinate CA certificate issued by the parent CA will have a PathLength constraint set appropriately in its Basic Constraints extension. You must restart AD CS CA for this change to take effect. Note that there is no easy way to undo this change and you may need to reissue subordinate CA certificates. Nevertheless, with clear understanding of your CA hierarchy basic constraints together with extended key usage constraints is the powerful mechanism to limit your issuing CAs.

### Extended Key Usage Constraints

As mentioned in the [Extended Key Usages](#_Extended_Key_Usages) section, an EKU extension indicates one or more purposes for which the certified public key may be used, in addition to or in place of the basic purposes indicated in the Key Usage Extension. In general, this extension will appear only in End Entity certificates, but if a CA includes EKUs to state allowed certificate usages, then it EKUs will be used to restrict usages of certificates issued by this CA. For example, if only the EKU for “Server Authentication” is included in the CA certificate, a certificate issued by that CA would not be valid for “Client Authentication.” In this case, EKU extension must not be marked as critical and anyExtendedKeyUsage must not be included in the extension.

Note that you may hit the limitations of software in validation of issued certificates, so in all cases you must test your target scenarios to verify the design decisions.

### Constraining CA Certificates

If you intend to restrict subordinate CA certificates, consider the following recommendations:

* For subordinate CA certificates, the Basic Constraints extension should be present and marked as critical
* The cA field should be set to TRUE
* The pathLenConstraint field should be set to the minimum value required to enable the business scenario (i.e. 0 if that CA will issue certificates only to End Entities)
* The EKU extension should be present and contain the minimum set of EKU object identifiers (OIDs) to enable the business scenario. Furthermore, the anyExtendedKeyUsage OID (2.5.29.37.0) should not be specified.

## Conclusion

Using strong cryptography throughout your PKI helps mitigate threats introduced when a cryptographic algorithm becomes weak and susceptible to failure, which can lead to a full loss of trust in your PKI. Utilizing constraints on End Entity certificates and CA certificates helps to mitigate the risk of having a certificate used for unintended use cases, which may allow access to resources or data that should not be allowed. Constraints also help mitigate the threat of an attacker creating their own subordinate CA in your PKI hierarchy after a compromise, allowing them to create certificates of their choosing.

For a complete list of the recommendations for planning certificate algorithms and usages, along with the level of [business impact](#_Determining_the_Level) at which you should consider implementing them, refer to [Appendix F: List of Recommendations by Impact Level](#_Planning_Certificate_Algorithms_1).

# Protecting CA Keys and Critical Artifacts

A primary security control in a PKI is how private keys are stored and managed, particularly for certification authorities. A strong key protection strategy along with other physical and logical controls can provide defense in depth to prevent external attackers or insider threats from compromising the integrity of the PKI. Additionally, a well-run PKI requires the secure storage of several artifacts, such as HSM activation cards or tokens, backup files, and documents. Improper storage of these artifacts can lead to the compromise of the entire PKI, without an attacker ever having to compromise a CA system. This section provides a set of recommendations for securing the keys used by CAs or other critical systems, and recommendations for securely storing other PKI artifacts.

## Protecting CA Keys

When designing a new PKI, a key design decision is whether or not to use an HSM or create keys stored in software. HSMs come in a number of form factors including rack-mounted appliances which can be attached using a serial or network connection, as well as smaller PCI or USB-connected devices. HSMs have a number of tamper-evident and self-destructing features. They are also available with multiple performance and compliance options. However, purchasing HSMs for PKI can represent a significant capital investment. For all CAs used in a production environment to secure corporate resources, a recommendation is to implement an HSM as part of a secure PKI. Below are a number of the benefits gained when using an HSM:

* HSMs provide the capability of enforcing additional controls whenever the CA key is used, such as enforcing [multi person control](#_Enforcing_Multi_Person)
* Implementing HSMs ensures that the private key cannot be used without access to a properly configured HSM. In the event of a compromise, the damage done can be limited. With a software-based key, any number of copies of the key can exist and be used from any computer.
* HSMs perform cryptographic operations on secure hardware. Even if a computer is compromised, the cryptographic keys are not available in the computer’s memory to be captured.
* HSMs can provide tamper evidence or tamper resistance for physical attacks

**Note:** Marking a software-based key as “not exportable” is not considered a security feature and should not be relied upon to prevent attackers from obtaining the private key.

In a default Active Directory® Certificate Services installation without an HSM present, the CA private key is stored using the Data Protection API (DPAPI) which encrypts the key using the local computer account credentials. Additionally, by default the CA private key is marked exportable. Therefore, any person authenticating with local administrator privileges can access, back up, or export these keys using a number of various methods including the Certificates MMC Snap-in, the backup function in the Certificate Services MMC Snap-In, and using certutil.exe and use them in another location. Use hardware protected keys and avoid using software-based keys for CAs unless the PKI is considered low assurance or the PKI is not used to protect high value data or transactions (e.g. lab environment, proof of concept, etc.) or if using an HSM is not feasible due to operational or fiscal constraints.

### Migrating Software Keys to HSMs

While it is technically possible in many cases to migrate an existing software-based key to an HSM, in general it is not the preferred approach. One of the benefits in using an HSM is the knowledge that the key has never been stored or used outside the secure HSM. Even if no compromise has occurred or is suspected, with a software-based key there is no real assurance that other copies of the key do not exist. In the event that you have an existing PKI and want to begin leveraging HSMs, consider a migration to a new infrastructure with new keys that are generated within the HSM.

### Network Based HSMs

Some HSMs are network based. Since offline CAs should not be connected to a live network, it is recommended that if network HSMs are used for offline CAs, that they are connected via a private network that only contains the CA and the HSM device. For example, a dedicated switch can be used to connect the two devices. Avoid connecting the private network to a live network.

### Multi Person Control

HSMs help enforce multi person control for sensitive processes, such as configuring a new HSM module or activating a key for use. This is commonly known as “k of n”, or having a “quorum.” The basic premise of k of n is to divide the interactions needed to access information among multiple entities. In the case of an HSM connected to a CA, multiple objects, typically cards or tokens, need to be connected to the HSM to generate or activate use of the CA private key. The cards or token can then be separated, distributed, and securely stored to help enforce these processes.

When determining how many tokens to create and how many to require for different tasks, it is important to take into account the following:

* The level of oversight required – For a sensitive root CA, it may be desirable to have multiple organizations hold portions of the access tokens, so there is oversight from multiple teams prior to any work being technically performed
* Operational overhead – Generally speaking, the greater the level of protection given to a key the greater the operational overhead will be to perform tasks on the CA
* Backup and disaster recovery – Always create enough objects such that a quorum of objects is available in the event of a disaster

Consider the following example and how it illustrates the recommendations above. A private key could be stored on an HSM where, to access the key, three of six objects would be needed to administer the HSM, and three of six objects would be required to activate the private key for use. In this example, where both the operational set and administrative set would require three objects for access, the six objects could be divided into two sets; a primary set and a backup set. After this division, the objects would be divided into three distinct parts. These three parts could be distributed to representatives from different organizations where their representatives would ensure the objects were individually inventoried, placed in tamper-evident containers, and escrowed to separate locations, or handled in a way to ensure separation.

The figure below shows this example relationship between HSM objects in creation and in distribution.



**Example HSM Object Relationships**

The figure above assumes there are three disjointed entities or organizations involved in this process. Example organizations are:

* **PKI Administration** – A team or organization ultimately responsible for maintenance and upkeep of the PKI servers
* **Information Security** – A team or organization responsible for securing data and assets without having administration ownership
* **Operations Security** – A team or organization responsible for architecting and/or enforcing physical and/or process security
* **Internal Audit** – A team responsible for ensuring security policies, procedures, guidelines and standards are being enforced
* **Other Stakeholders** – Teams or Organizations dependent on PKI services

By ensuring the involvement of organizations beyond PKI Administration, a level of checks and balances is achieved. The cooperation and continued involvement of these organizations is crucial. Ideally, no one person or organization controls these critical assets.

#### Enforcing Multi Person Control

If the policy in the fictitious example above is to require all three organizations to participate whenever the CA key is used, additional controls need to be put into place. In this example it would be possible for only two organizations to be required if one of those organizations obtained both their primary and backup objects. When designing how to separate these critical objects, ensure that adequate technical or procedural controls exist to ensure that no single person can access a quorum of objects. For example, further enforcement could be performed by leveraging the storage facility access procedures. Staff of the storage facility could help by verifying identification and enforcing logical access by requiring multiple person’s access challenges or by utilizing separate secure containers with multiple locks. The process can also be simplified by placing secure storage capacity onsite in control of the data center operations security.

One way to regulate access to secured assets is to ensure two representatives are present and positively identified prior to being granted access to their secured assets. For example, a combination of photo identification, biometrics, or other means of identification can be required prior to allowing access to the secure facility, or the location housing the offline CAs. An example of object mapping is shown in the figure below, where there are three fictional organizations involved; PKI Administration (PKIAdmin), Information Security (InfoSec), and Operations Security (OpsSec).



**Example Object Storage Map**

#### Multi Person Control without HSMs

In situations where an HSM cannot be used, it is critical to protect and monitor persons and accounts with local administrator privileges. Multi-person control can be achieved in several ways including multiple door locks, incorporating multi-factor authentication and separating factors, or even having multiple people set a single password for an elevated account where one person setting a password part does not disclose their part to the others setting a password part. In the example below, a pseudo-randomly generated 60 character strong password is separated into distinct parts.



**Example Password Part Separation**

The password parts could be generated and set separately, and stored similarly to the HSM object storage example above. The figure below shows a simple example solution for separating password parts between the three fictitious organizations.



**Password Part Storage Map**

If a method in which passwords are written or printed is used, special care should be taken to ensure the password parts are distinctly generated and not stored electronically.

## Artifact Protection and Chain of Custody

Protecting the private key ensures that the trust granted to the CA is protected. If the private key is protected by an HSM, handle the HSM cards or tokens as critical assets. These objects, along with any other important data such as backup drives, USB-form factor HSMs, standalone safe keys, written combinations, or written password parts need to be tracked, inventoried, and verified end to end. If this data is not properly secured, it may be possible for an attacker to compromise the PKI without ever having to compromise a single computer.

### Use Tamper-Evident Containers

Some assets in a PKI should have their usage tracked, and any unauthorized use or attempted access should be detectable. For these situations, it is recommended to use tamper-evident bags or containers to store assets. For example, using tamper-evident bags for objects used to activate a CA private key provides assurance that the objects have not been used or compromised.

Choosing a good tamper-evident container is not a trivial task. Ideally, the container has a strong seal, has a good resistance to humidity, has a unique number stamped on the outside, and has an area to write details for chain of custody. Since these objects may contain electronics, anti-static properties may be desirable. Some manufacturers will offer samples to evaluate before making a purchase. Stores specializing in assisting financial businesses or law enforcement have many options for tamper-evident containers as well.

While it may be a good idea to use a transparent container for asset-tagged artifacts for identification and verification, ensure any artifacts with secrets such as password parts are safely concealed, either by using a non-transparent tamper-evident container, placing the material in a non-transparent envelope, or placing the envelope in the transparent tamper-evident container.

### Artifact Storage

PKI artifacts are often required to be stored for long lengths of time. When storing artifacts, ensure that they are in a climate-controlled environment such as an onsite datacenter vault or a safe-deposit box. Silica gel packets can be used to eliminate any residual moisture if they are inserted with artifacts in anti-static bags or storage containers.

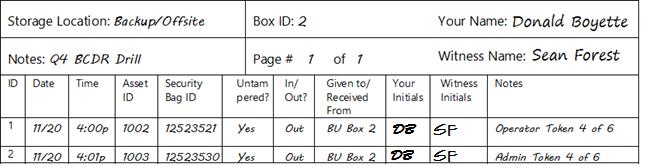
### Maintain a Chain of Custody and Asset Inventory

As you operate a PKI, the secure assets may need to be moved, or possession may change to another person or organization. For all secure assets, create a chain of custody that tracks who is responsible for the asset. Keeping a chain of custody will ensure that if there is ever an investigation into misuse of the asset, a strong audit trail will exist to show who had possession at all times.

When storing the artifacts in a vault, safe, or other type of storage cell, perform a hand-written check-in log and inventory and retain the logs. When extracting the artifacts, the previous inventory needs to be verified, and a check-out log written. These check-in and check-out logs should have fields for the following:

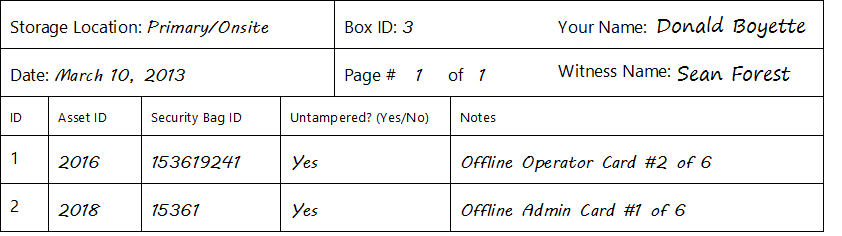
* Date and time
* The tamper-evident container unique code or number
* Identification or asset label of the artifact contained within the container
* Verification that the container is not tampered with
* The person performing the inventory
* Check-out or check-in
* A location where the artifacts were extracted or secured

Below is an example of a check-out log with fictitious entries.



**Sample Check-out log**

At regular intervals, even if artifacts are not used, inventories should be performed and a copy of the inventory stored with the artifacts. Below is a sample inventory log with fictitious entries.



**Sample Inventory Log**

## Conclusion

Using an HSM to provide strong protection of CA keys or other high value keys is one of the strongest controls you can implement to protect your PKI. HSMs can provide assurance that keys are only available for use from authorized systems and help to mitigate the risk of an attacker or insider threat obtaining unrestricted access to the keys. Protecting other PKI assets and auditing their use helps to ensure that keys are not used in an unauthorized manner after they have been created in the HSM.

For a complete list of the recommendations for protecting CA keys and critical artifacts, along with the level of [business impact](#_Determining_the_Level) at which you should consider implementing them, refer to [Appendix F: List of Recommendations by Impact Level](#_Protecting_CA_Keys_4).

# Monitoring Public Key Infrastructure

A key component of any PKI security plan is ongoing monitoring of the infrastructure and supporting processes. With critical systems such as aCA, it is vital to collect the right information and have appropriate alerting configured and review processes defined so that if there is an issue, it can be properly investigated. CAs must be treated as a high value systems and be monitored closely for suspicious activity.

A security logging and monitoring plan for a traditional line of business application will typically include watching specific logs for indicators of suspicious activity. While this is a vital part of PKI logging and monitoring, there are more aspects to consider. If operating offline CAs, also monitor the physical access to the offline CAs, and create a process for reviewing logical access to those systems. You must also monitor and review physical access for all online systems, including RAs and CAs.

The events discussed in this section will assist you in creating a security monitoring plan for PKI. Monitoring PKI for operational issues is not considered in this paper. The recommendations are strictly for determining malicious or suspicious activity related to the PKI.

## Monitoring Events

An event is defined as any activity recorded to provide a detailed history of actions that have taken place. Some events may be tracked electronically within the system, while others may be tracked via manual paper-based processes. Any event collected, electronic or otherwise, should contain the following information:

* Type of event
* Date and time of the occurrence of the event
* Identity that logged the event
* Success or failure where appropriate
* Description of the event

Regardless of the technology being used to generate or collect alerts, it is vital to have a process in place that engages the correct support teams if events indicate a security incident has occurred or will occur. Events that generate a security alert should contain the following:

* High likelihood that occurrence of the event indicates unauthorized activity
* Low number of false positives
* Result in an investigative/forensics response

Two types of events should trigger security alerts:

1. Events in which even a single occurrence indicates unauthorized activity
2. An accumulation of events above an expected and accepted baseline

An example of the first type of event occurs if a user not authorized for interactive logon on a CA registers a logon event. An example of the second type of event is multiple failed access attempts from a user, which could be a sign of a brute force password guessing attack.

For electronically collected events, develop a plan for continued storage of collected event data and retention of the events. Consider using tools such as Windows Event Forwarding® (WEF) to aggregate events on a separate system for analysis and long term storage. For events recorded through manual processes, establish processes for archiving and reviewing event logs for abnormalities at regular intervals.

## Monitoring Active Directory Objects and Attributes

The following are important Active Directory® items to monitor in order to detect compromise of AD CS based PKI:

* Changes to critical groups that control access to the CA. This would include any custom groups containing users with elevated rights in the PKI to manage CAs, RAs, or enroll for important certificate types
* Changes in membership to the “Cert Publishers” domain local group(s)
* Changes to accounts that have privileged access to the PKI. Monitor for changes to attributes on the Account tab (cn, name, sAMAccountName, userPrincipalName, or userAccountControl). When using additional software packages that act as a RA to a CA, include the service accounts used by these systems in this list.
* Unauthorized changes to certificate templates (Refer to [Monitoring Change to Certificate Templates](#_Monitoring_Changes_to))

## Monitoring Certification Authority Activity

Because a CA is a high-value system, monitor it closely for abnormal activity. The events to monitor closely can be broken down into two major categories:

1. Events to watch for on any high value system
2. Events that are specific to the AD CS CA role

Below are some of the activities that should be monitored to help detect compromise of an AD CS based PKI:

**General Events**

* Successful and failed logons
* Addition, removal, or deletion of user accounts
* Changes to membership in the local administrators group
* Usage of the built-in administrator account
* Changes to system time outside a defined threshold (changes greater than ten minutes)
* Abnormal startup or shutdown events
* Clearing of event logs
* Disabling or modification of antivirus and antimalware software
* Antivirus or antimalware action taken (quarantine, etc.)
* Installation of new services
* Unknown processes starting or stopping

**CA-specific activity**

* Unauthorized changes to CA security settings
* Revocation of a significant number of certificates during a short time period
* Changes to the audit filter settings for the CA
* Issuance of certificates that contain restricted usages (Enrollment Agent, Key Recovery Agent)
* Changes to the active Policy Module on the CA
* Changes to the configured Key Recovery Agents
* Changes to role separation settings if role separation is enabled
* Addition of certificate templates that are not normally issued by the CA
* Addition or deletion of certificates from the CA database
* Usage of the CA private key outside of certsrv.exe (certutil.exe, custom executables or scripts)
* Suspicious use of accounts belonging to registration authorities. For example, if a smart card management system uses a specific service account to request certificates from the CA and that account makes certificate requests from systems that are not part of the smart card management system.

## Recording and Reviewing Additional Events

In addition to monitoring CAs that are online and issuing certificates, it is also important to record and review other events that may impact PKI security. These events may not be captured electronically, but may rely on paper-based logs and require periodic review for anomalies. Below are some recommendations for additional activities to record and review:

* Entry and exit to the secure area where PKI hardware is stored or operated. This could include access to the secure CA cage, access to the server room where the CAs are located, review of camera footage, etc.
* Access to Hardware Security Modules (HSMs) and any tokens used to activate the HSMs. This includes any transportation of HSMs or tokens when they are physically moved.
* Firewall/router activities that may indicate compromise
* Access to any secure storage locations containing PKI backups or sensitive data. Examples include access logs for a safe, access records to a document archive facility, etc.

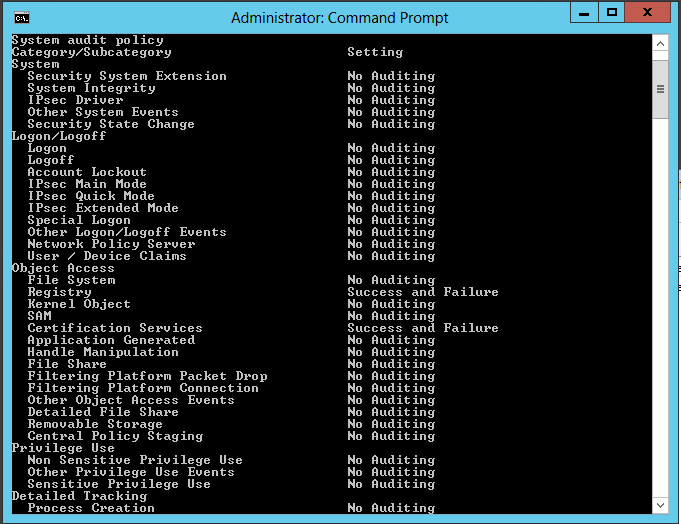
## Configuring Microsoft Windows Audit Policy

In order to take full advantage of the auditing capabilities available in AD CS, you should configure an appropriate Microsoft Windows® audit policy for all of the events logged. Beginning with Microsoft Windows Vista® and Microsoft Windows Server 2008®, Microsoft improved the way event log category selections are made by creating subcategories under each main audit category. Subcategories allow auditing to be far more granular than it could be otherwise by using the main categories. By using subcategories, portions of a particular main category can be enabled, and events for which there is no need can be eliminated. Each audit policy subcategory can be enabled for Success, Failure, or Success and Failure events.

For a complete description of Microsoft Windows® audit policies and subcategories, refer to [Best Practices for Securing Active Directory](http://www.microsoft.com/en-us/download/confirmation.aspx?id=38785). To see the current audit policy for a system, type the following at the Command Prompt:

**auditpol /get /category:\***

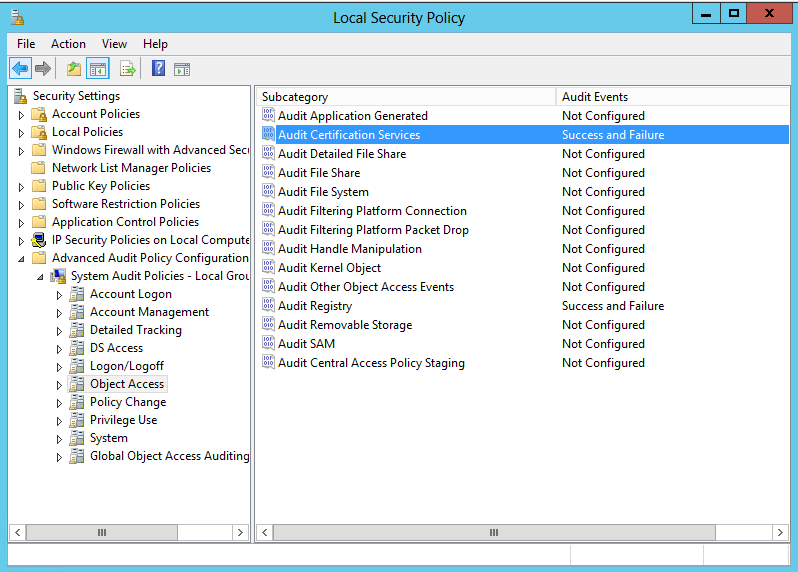
The following screenshot represents an audit policy status.

****

In order for AD CS to log all configured events, the audit policy must have the “Audit Certification Services” subcategory configured for success and failure. To configure this using auditpol.exe, type the following at the Command Prompt:

**auditpol /set /subcategory:”Certification Services” /success:enable /failure:enable**

While audit policy can be configured per computer using auditpol.exe, a preferable method for domain-joined CAs is to apply a common audit policy as appropriate and configure it using group policy. To set audit policy using group policies, configure the “Certification Services” subcategory under **Computer Configuration\Windows Settings\Security Settings\Advanced Audit Policy**. Set the subcategory to be enabled for **Success and Failure**. See the screenshot below.



### Command Line Process Auditing

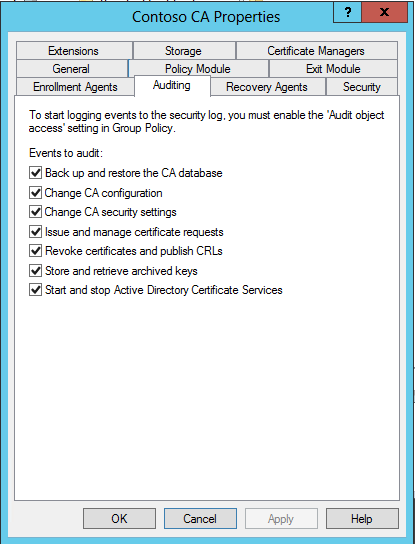
Beginning with Windows Server 2012 R2®, additional data containing command line information can be collected when you are auditing for process creation events. This data includes the exact command used to launch a process, including command line parameters. More information on configuring command line process auditing can be found [here](http://technet.microsoft.com/en-us/library/dn535776.aspx).

## Configuring Certification Authority Auditing

The CA role in Active Directory® Certificate Services includes two main sources of events:

* **Microsoft-Windows-CertificationAuthority** – Contains operational and installation events related to the CA. Object access auditing is not required for these events to be written to the Application log.
* **Microsoft-Windows-Security-Auditing** – Contains numerous events related to the security and configuration of the CA. Object access auditing must be configured for Certification Services and an appropriate CA audit filter must be configured.

The CA audit filter is a bitmask value stored in the registry of the CA. Refer to [Appendix B: Certification Authority Audit Filter](#_Appendix_B:_Certification) for details on each individual value and which flags are required to trigger specific audit events. As shown in the following screenshot taken from the **Properties** dialog box of the CA, there are seven categories available for auditing:



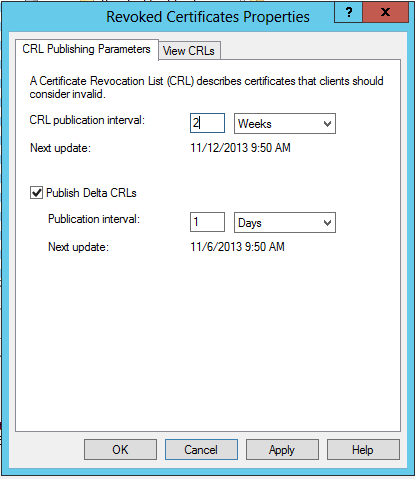
* **Back up and restore the CA database** –Controls logging of events triggered when the CA database is issued backup or restore commands
* **Change CA configuration** –Controls logging of events related to the changing of properties and configuration of the CA through the CA snap-in. Example events logged are changing CRL validity periods, changing policy or exit module configuration, or updating configured CDP/AIA extensions.
* **Change CA security settings** – Controls logging of events triggered by modification of the CA security settings done through the CA snap-in. Example events include enabling/disabling role separation, changing the audit filter, or changing the access control list for the CA.
* **Issue and manage certificate requests** – Controls logging of events related to the issuance of certificates. This includes logging when a request is received, or set to pending, denied, and issued. In high volume issuance environments this can generate a large number of alerts, but it is a recommendation to enable it where possible because it provides a strong audit trail of all issuance events.
* **Revoke certificates and publish CRLs** –Controls auditing of events related to revocation and publishing of CRLs.
* **Store and retrieve archived keys** – Controls auditing of events related to the CA archiving keys or recovering previously archived keys. This includes when a key is imported into the CA database and archived.
* **Start and stop Active Directory Certificate Services** –Controls creation of audit events whenever AD CS is started and stopped. A similar event is also logged to the application log, although enabling of this event writes an event to the security log. If enabled, a cryptographic hash of the CA database is taken on startup and shutdown of the CA service. When the database becomes large, this may begin to impact service availability, as the RPC interface for the CA is not available while the hash is being computed. The start and stop times of the service may be very long depending on the size of the database.

## Advanced CA Monitoring

Although enabling auditing for AD CS provides a solid foundation for capturing events that occur within the scope of the CA service, additional events can occur on the CA that may indicate a compromise or a potential compromise. These additional events are useful to capture in addition to the CA audit events.

### Auditing CA Registry Changes

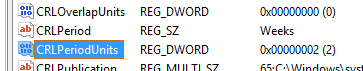
Several of the CA auditing categories capture events triggered when a user changes a configuration setting within the CA snap-in. Several methods exist to change many of the settings on a CA. For example, to change the validity period of CRLs signed by the CA from one week to two weeks, you can utilize the CA snap-in as illustrated in the screenshot below.



Another method to perform the same task involves using certutil.exe with the following commands:

**certutil –setreg CA\CRLPeriod Weeks  
certutil –setreg CA\CRLPeriodUnits 2**

Yet another method involves using regedit.exe:



With an audit filter configured to capture this event, the only method that will trigger an alert is making the change through the snap-in. To capture changes to all AD CS configuration settings stored locally on the CA, configure the registry auditing specifically for the AD CS registry keys. Configuring registry auditing at a broad level is not recommended because it can generate a very large number of alerts.

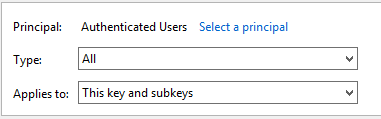
To enable registry auditing, configure the Microsoft Windows® auditing policy. This can be accomplished with auditpol.exe or through local or group policy.

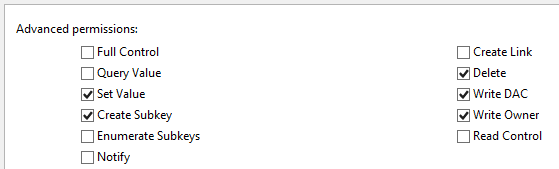
To enable registry auditing with auditpol.exe, use the following settings:

**auditpol /set /subcategory:”Registry” /success:enable /failure:enable**

To set audit policy using group policies, configure the “Audit Registry” subcategory under **Computer Configuration\Windows Settings\Security Settings\Advanced Audit Policy**. Set the subcategory to be enabled for success and failure.

After enabling registry auditing, configure auditing for the Certification Services registry keys. To configure auditing for the AD CS CA registry key:

1. Open **regedit**, and navigate to **HKLM\System\Services\CertSvc\Configuration\**
2. Right-click the **Configuration** registry key and click **Permissions…**
3. Click **Advanced**.
4. Click **Auditing**.
5. Click **Select a principal** and select **Authenticated Users**. From the drop down menus, for **Type** select **All** and for **Applies To,** select **This key and subkeys**.  
   
6. Click **Show advanced permissions** and select the values shown below.



For specific registry values to watch, refer to [Appendix A: Events to Monitor](#_Registry_Values_to). For more information on configuring registry auditing, refer to [How to use Group Policy to audit registry keys in Windows Server 2003](http://support.microsoft.com/kb/324739).

### Monitoring Changes to Certificate Templates

Certificate templates are Active Directory® objects that define key attributes of certificates issued by an Enterprise CA. Standalone CAs do not use certificate templates. Instead they rely on information provided in certificate requests and provided by certificate managers to determine certificate content. An Enterprise CA uses the information in certificate templates to determine who has permissions to enroll and what usages an issued certificate should be valid for. If an attacker has the ability to modify a Certificate Template in Active Directory®, they could potentially add additional usages or change enrollment permissions that could lead to the issuance of certificates that allow additional access.

AD CS includes several audit events that allow monitoring of changes to certificate templates that are actively being used by a CA. The following audit events are available:

* **Certificate Services loaded a template (Event ID 4898)** – This event is triggered whenever a CA loads a template for the first time. For example, if a CA is configured with three templates, at startup this event will trigger for each template as it loads. If a fourth template is added while the CA is running, an event will be triggered on the first attempt to enroll the template on the CA.
* **A Certificate Services template was updated (Event ID 4899)** – This event is triggered when a template loaded by the CA has an attribute updated and an enrollment is attempted for the template. For example, if an additional EKU is added to a template, this event would trigger and provide enough information to determine the change being made.
* **Certificate Services template security was updated (Event ID 4900)** – This event is triggered when security permissions on a Certificate Template loaded on a CA are changed, and an enrollment event for the template occurs.

For the template change events to be recorded, configure the CA for auditing of CA events as described in [Configuring Windows Audit Policy](#_Configuring_Microsoft_Windows) and [Configure Certification Authority Auditing](#_Configuring_Certification_Authority_1). In addition, enable the CA audit setting for “Change CA settings”, and enable a specific policy configuration. To set the policy configuration to enable audit of template events, run the following command:

**certutil –setreg policy\EditFlags +EDITF\_AUDITCERTTEMPLATELOAD**

#### Certificate Template Scenarios to Monitor

When auditing templates, consider the following scenarios for monitoring:

* Changes to templates that add new EKUs (Code Signing, Enrollment Agent, Smart Card Logon, etc.)
* Addition of unexpected new templates on the CA
* Changes to permissions for enrollment
* Changes to permissions for write access to a template
* Assignment of new templates that allow “supply in request” to build the subject

## Conclusion

Implementing monitoring and alerting for PKI systems is a strong detective control, and can help limit the extent of damage caused by an attack or identify an attack before it succeeds. Monitoring will help you identify unauthorized changes to the environment that may allow certificates to be issued to unauthorized users or contain unexpected usages.

For a complete list of the recommendations for monitoring PKI, along with the level of [business impact](#_Determining_the_Level) at which you should consider implementing them, refer to [Appendix F: List of Recommendations by Impact Level](#_Monitoring_Public_Key_1).

# Compromise Response

As with any other critical piece of infrastructure, it is vital to have a plan of action in the event of a PKI compromise. Unfortunately there is not a one-size-fits-all formula for what to do when a PKI is compromised or presumed compromised. Depending on the type and severity of the compromise, you may have several response options, each with their own positive and negative attributes. This section will focus on steps that can be taken prior to compromise and during response to help you make the best decisions to protect the security and availability of your business.

## Inventory PKI Consumers

It is very important to know who the PKI consumers are and to identify systems dependent on PKI. For a Microsoft PKI using Enterprise CAs, use the certificate database to get a general idea of the types of certificates issued using the CA MMC snap-in or using the **certutil –view** command. In general, the certificate templates used by the CA can help to identify the consumers. However, the database may have entries removed and there is a configuration option for certificate templates to allow high-volume certificates to be issued without being entered into the certificate database. Consumers using certificate templates requiring input for the subject name, such as those used for issuing certificates to web servers, or consumers requiring a Certificate Manager approval or an Enrollment Agent need to be handled individually. Consumers using certificate templates with autoenrollment where the request processing is transparent merely need a refresh of their certificate.

## Understanding Compromise Scenarios

There are many [different attack vectors](#_Compromising_PKI_1) for a PKI. In general, attack scenarios can be broken down into four main categories.

### Full Key Compromise

An attacker has a copy of the private key and can sign whatever they desire. An example is a case where an attacker steals a code-signing certificate and can sign arbitrary code from any system of their choosing, or where a PKCS#12 file containing the CA private key is exported from the CA.

### Full Key Access

An attacker has access to the private key, such that the CA does not have sufficient information to determine the set of certificates that needs to be revoked. An example is when an attacker gains access to a CA where the key is stored on an HSM, but allows arbitrary requests to be signed such as with **certutil –sign** in a Windows® environment.

### Limited Key Access

An attacker has access to a certificate issuance system and can get certificates, but the system has full records of the unauthorized certificates issued. In this situation, revocation is possible. An example is if enrollment agent credentials are obtained. The CA will still log all issued certificates.

### Other Attacks

This category includes other attacks such as Denial of Service (DoS) which may block the issuance of new certificates and revocation lists, theft of an HSM without access token, or theft of partial set of HSM access tokens which may make it easier for the attacker to obtain the CA’s private key.

## Full Key Compromise and Access

When a CA is compromised, assume that every certificate issued by the CA is potentially compromised. The response from an End Entity perspective is slightly different, and the handling of the CA or PKI in general depends on the type of compromise.

There are two major compromise possibilities for full key compromise or access for a CA - an operating system compromise or a cryptographic compromise. Both of these compromises are covered below.

### Operating System Compromise

There are a number of attack vectors for the operating system of a CA server and the response depends on many factors. Operating systems can be compromised physically or remotely.

Examples of physical attacks involve an attacker gaining access to a server by using the console due to:

* An unlocked server
* An attack against credentials
* An insider attack using stolen or known good credentials
* The use of storage media to inject an exploit or create a unique bootable operating system partition to make changes to the primary operating system drive

Both offline and online CA server types are susceptible to physical attack, whether it is by an intruder, an insider attack, or an unknowing person via a social engineering attack or infected file transfer device. If an offline CA is kept offline, it is not susceptible to remote attacks. However, online CA servers as well as web servers responsible for enrollment services or enrollment validation are susceptible.

Examples of remote attacks are:

* Brute force attacks against credentials to gain access using Remote Desktop Services or other remote management tools
* Utilizing an operating system vulnerability to gain access to a system
* Malware introduced by an operating system vulnerability or an unknowing person with access to the system being coerced into installing the malware

For CA servers, regardless if the operating system compromise is physical or remote, the severity of the compromise and the corresponding response depends on whether the private key integrity is known to be good or if the key integrity is unknown or compromised.

### Cryptographic Compromise

Another attack vector that can lead to full key access is the cryptography of the PKI. For each public key, there is only one mathematically unique private key and the algorithm to perform encryption and decryption is well known. Researchers or dedicated attackers can dedicate multiple servers and build testing algorithms to try to derive the private key by brute force. If a weakness is found in an algorithm used by the CA, the weakness could be further exploited to identify the private key or issue certificates that appear to come from the CA.

## CA Compromise Response Actions

After identifying there has been compromise, the first actions taken after determining the nature and degree of damage are to restore functionality and assurance of the CA and any End Entity certificate consumers. Some actions are more severe than others and which actions you execute depends on the type of compromise. You should document response actions in an Operations Guide or in Business Continuity plans. The following are some examples of response actions.

* Revoke individual CA certificate and publish parent CRL
  + [Log on to the parent CA and revoke the subordinate CA certificate](http://technet.microsoft.com/en-us/library/cc739815(v=ws.10).aspx)
  + [Publish a new full CRL](http://technet.microsoft.com/en-us/library/cc778151(v=ws.10).aspx)
  + Ensure CRL is copied to CDP locations
  + Force CRL Cache renewal on client workstations
  + Update parent OCSP Responders
* Force new issuance of End Entity certificates
  + Auto enrollment to replace user or computer certificates as possible
  + Certificates requiring an enrollment agent, certificate manager approval, or other intervention handled individually.
* Addition of CA certificate to untrusted store
  + [Publish in Group Policy](http://technet.microsoft.com/en-us/library/cc772491.aspx)
* Retire CA server
  + Copy any data, logs, or databases needed for retention
  + Reformat or destroy the hard drives
  + Recycle server chassis parts
* Complete server replacement and HSM re-initialization
  + Obtain new CA server hardware
  + Re-initialize HSM security world or partition
  + Perform a new key ceremony
* Use existing server or replace server hardware and restore backup
  + Use existing CA server hardware as possible or obtain new hardware
  + Reinstall operating system and dependent parts
  + Restore a known good backup
* Update exploited vulnerability
  + Utilize existing hardware as possible or obtain new hardware
  + Reinstall operating system and dependent parts
  + Restore a known good backup
  + Update system after operating system is restored
* Renew CA Keys [For the entire chain/ use a larger key size / use an uncompromised algorithm]
  + Offline and Online renewal of CA Keys
  + Ensure certificates are published to AIA locations
* Migrate Key Archive
  + Extract key archival data
  + Migrate to new server
* Establish new Enrollment Agents
  + Enroll for certificates
* Establish new Key Recovery Agents
  + Enroll for certificates
  + Configure Key Archive for new agents

## Correlating Compromise Types and Response Actions

Once you understand the types of compromises and have established the potential actions for response, put a response plan in place. The diagrams below shows an example correlation between compromise and actions for a server operating system compromise and an example correlation between compromise and actions for a cryptographic compromise.



**Example Compromise Action Correlation for Server Operating System Compromise**



**Example Compromise Action Correlation for Cryptographic Compromise**

Once the response plan is complete, perform drills to practice execution of a compromise response in a test environment. Some compromise types can be addressed using the same set of actions. For example, the response for the compromise type in the diagram **Example Compromise Action Correlation for Server OS Compromise** | Online | Remote\Vulnerability | Unknown\Compromised is identical to the response in the diagram **Example Compromise Action Correlation for Cryptographic Compromise** \Online\Logical\OS Vulnerability compromise type.

## Conclusion

While it is not possible to enumerate every potential avenue of compromise, having a generalized response plan in mind prior to an event will accelerate the efforts for recovery and remediation. For a complete list of the recommendations for compromise response, along with the level of [business impact](#_Determining_the_Level) at which you should consider implementing them, refer to [Appendix F: List of Recommendations by Impact Level](#_Compromise_Response).

# Appendices

Appendices are included in this document to augment the information contained in the body of the document. The list of appendices and a brief description of each in included the following table.

|  |  |
| --- | --- |
| **Appendix** | **Description** |
| [Appendix A: Events to Monitor](#_Appendix_A:_Events) | Provides a full list of events generated by Active Directory® Certificate Services CA role, along with recommendations for which events should be monitored. |
| [Appendix B: Certification Authority Audit Filter](#_Appendix_B:_Certification) | The CA audit filter is a bitmask value representing the seven different audit categories that can be enabled. |
| [Appendix C: Delegating Active Directory PKI Permissions](#_Appendix_C:_Delegating) | Provides instructions for delegating permissions for Enterprise CA Installations and permissions for Managing certificate templates. |
| [Appendix D: Glossary of Terms](#_Appendix_D:_Glossary) | A list if PKI terms and their definitions. |
| [Appendix E: PKI Basics](#_Appendix_E:_PKI) | Provides an introduction to some PKI basic concepts. |
| [Appendix F: List of Recommendations by Impact Level](#_Appendix_F:_List) | A complete list of all recommendations made throughout this paper, classified according to the [impact level](#_Determining_the_Level) of the CA |

# Appendix A: Events to Monitor

The following tables provide a full list of events generated by Active Directory® Certificate Services CA role, along with recommendations for which events should be monitored. Unless otherwise specified in the description, the events are available on Microsoft Windows Server 2008® and more recent versions of Windows®.

The “Current Windows Event ID” column lists the current event ID as it is implemented in versions of Microsoft Windows Server® that are currently in mainstream support. The “Potential Criticality” column identifies whether the event should be considered low, medium or high criticality in detecting attacks. The event summary contains a brief description of the event.

A potential criticality of high means that one occurrence of the event should be investigated. Potential criticality of medium or low means that these events should only be investigated if they occur unexpectedly in numbers that significantly exceed the expected baseline in a measured period of time, or the content of the message meets a specific criteria. All organizations should test these recommendations in their environments before creating alerts that require mandatory investigative responses. Every environment is different, and some of the events ranked with a potential criticality of high may occur due to other harmless events.

**Microsoft Windows**® **Security Auditing**

| **Current Windows Event ID** | **Potential Criticality** | **Event Summary** | **Audit Filter Required** | **Description** |
| --- | --- | --- | --- | --- |
| 4868 | Low | The certificate manager denied a pending certificate request. Request ID: %1 | Issue and manage certificate requests |  |
| 4869 | Low | Certificate Services received a resubmitted certificate request. Request ID: %1 | Issue and manage certificate requests |  |
| 4870 | Low | Certificate Services revoked a certificate. Serial Number: %1 Reason: %2 | Revoke certificates and publish CRLs |  |
| 4871 | Low | Certificate Services received a request to publish the certificate revocation list (CRL). Next Update: %1 Publish Base: %2 Publish Delta: %3 | Revoke certificates and publish CRLs |  |
| 4872 | Low | Certificate Services published the certificate revocation list (CRL). Base CRL: %1 CRL Number: %2 Key Container: %3 Next Publish: %4 Publish URLs: %5 | Revoke certificates and publish CRL |  |
| 4873 | Medium | A certificate request extension changed. Request ID: %1 Name: %2 Type: %3 Flags: %4 Data: %5 | Issue and manage certificate requests | If this functionality is not used by the CA, it may indicate tampering with a request |
| 4874 | Medium | One or more certificate request attributes changed. Request ID: %1 Attributes: %2 | Issue and manage certificate requests | If this functionality is not used by the CA, it may indicate tampering with a request |
| 4875 | Low | Certificate Services received a request to shut down. | Start and stop Active Directory® Certificate Services | This event is triggered when the **certutil –shutdown** command is issued to the CA |
| 4876 | Low | Certificate Services backup started. Backup Type: %1 | Back up and restore the CA database |  |
| 4877 | Low | Certificate Services backup completed. | Back up and restore the CA database |  |
| 4878 | Low | Certificate Services restore started. | Back up and restore the CA database |  |
| 4879 | Low | Certificate Services restore completed. | Back up and restore the CA database |  |
| 4880 | Low | Certificate Services started. Certificate Database Hash: %1 Private Key Usage Count: %2 CA Certificate Hash: %3 CA Public Key Hash: %4 | Start and stop Active Directory® Certificate Services |  |
| 4881 | Low | Certificate Services stopped. Certificate Database Hash: %1 Private Key Usage Count: %2 CA Certificate Hash: %3 CA Public Key Hash: %4 | Start and stop Active Directory® Certificate Services |  |
| 4882 | High | The security permissions for Certificate Services changed. %1 | Change CA security settings | May indicate an attacker granting permissions for other accounts to enroll. |
| 4883 | Medium | Certificate Services retrieved an archived key. Request ID: %1 | Store and retrieve archived keys |  |
| 4884 | Low | Certificate Services imported a certificate into its database. Certificate: %1 Request ID: %2 | Issue and manage certificate requests |  |
| 4885 | High | The audit filter for Certificate Services changed. Filter: %1 | Change CA security settings | May indicate an attacker disabling monitoring in an attempt to cover their tracks prior to certificate activities. |
| 4886 | Low | Certificate Services received a certificate request. Request ID: %1 Requester: %2 Attributes: %3 | Issue and manage certificate requests |  |
| 4887 | Medium | Certificate Services approved a certificate request and issued a certificate. Request ID: %1 Requester: %2 Attributes: %3 Disposition: %4 SKI: %5 Subject: %6 | Issue and manage certificate requests | Issuance of certificates that contain usages that allow the owner to perform privileged operations (Enrollment Agent, Code Signing etc.) or certificates issued to VIP users should be monitored. |
| 4888 | Medium | Certificate Services denied a certificate request. Request ID: %1 Requester: %2 Attributes: %3 Disposition: %4 SKI: %5 Subject: %6 | Issue and manage certificate requests |  |
| 4889 | Low | Certificate Services set the status of a certificate request to pending. Request ID: %1 Requester: %2 Attributes: %3 Disposition: %4 SKI: %5 Subject: %6 | Issue and manage certificate requests |  |
| 4890 | High | The certificate manager settings for Certificate Services changed. Enable: %1 %2 | Change CA security settings | May indicate tampering with permissions with what users are able to enroll on behalf of other users, commonly used to issue smart card certificates. |
| 4891 | Medium | A configuration entry changed in Certificate Services. Node: %1 Entry: %2 Value: %3 | Change CA configuration | Can be used to monitor for changes to Policy/Exit modules on the CA or configuration of CDP/AIA extensions. |
| 4892 | Medium | A property of Certificate Services changed. Property: %1 Index: %2 Type: %3 Value: %4 | Change CA configuration | Can be used to track changes to Key Recovery Agent configuration |
| 4893 | Low | Certificate Services archived a key. Request ID: %1 Requester: %2 KRA Hashes: %3 | Store and retrieve archived keys |  |
| 4894 | Low | Certificate Services imported and archived a key. Request ID: %1 | Store and retrieve archived keys |  |
| 4895 | Low | Certificate Services published the CA certificate to Active Directory® Domain Services. Certificate Hash: %1 Valid From: %2 Valid To: %3 |  |  |
| 4896 | High | One or more rows have been deleted from the certificate database. Table ID: %1 Filter: %2 Rows Deleted: %3 | Issue and manage certificate requests | May indicate an attacker covering their tracks after issuing certificates. |
| 4897 | Medium | Role separation enabled: %1 | Change CA security settings | If role separation is used, this can be used to trigger an alert if the expected configuration changes. |
| 4898 | Medium | Certificate Services loaded a template. %1 v%2 (Schema V%3) %4 %5 Template Information: Template Content: %7 Security Descriptor: %8 Additional Information: Domain Controller: %6 | Change CA security settings | Alert if templates that are not expected on a CA are loaded. |
| 4899 | Medium | A Certificate Services template was updated. %1 v%2 (Schema V%3) %4 %5 Template Change Information: Old Template Content: %8 New Template Content: %7 Additional Information: Domain Controller: %6 | Change CA security settings |  |
| 4900 | Medium | Certificate Services template security was updated. %1 v%2 (Schema V%3) %4 %5 Template Change Information: Old Template Content: %9 New Template Content: %7 Old Security Descriptor: %10 New Security Descriptor: %8 Additional Information: Domain Controller: %6 | Change CA security settings |  |

**Microsoft-Windows-CertificationAuthority**

| **Current Windows Event ID** | **Potential Criticality** | **Message** | **Description** |
| --- | --- | --- | --- |
| 3 | Low | Request failed | Not available in Microsoft Windows Server 2012® |
| 5 | Low | Active Directory Certificate Services could not find required registry information. The Active Directory Certificate Services may need to be reinstalled. |  |
| 6 | Low | Active Directory Certificate Services issued a certificate for request %1 for %2. |  |
| 7 | Low | Active Directory Certificate Services denied request %1 because %2. The request was for %3. |  |
| 8 | Low | Active Directory Certificate Services left request %1 pending in the queue for %2. |  |
| 9 | Low | The Active Directory Certificate Services did not start: Unable to load an external policy module. |  |
| 10 | Low | Active Directory Certificate Services were unable to build a new certificate or certificate chain: %1. |  |
| 15 | High | Active Directory Certificate Services did not start: Version does not match certif.dll. |  |
| 16 | Low | Active Directory Certificate Services did not start: Unable to initialize OLE: %1. |  |
| 17 | Low | Active Directory Certificate Services did not start: Unable to initialize the database connection for %1. %2. |  |
| 19 | Low | Active Directory Certificate Services did not start: The Subject Name Template string in the registry value HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\CertSvc\Configuration\%1\SubjectTemplate is invalid. An example of a valid string is: CommonName OrganizationalUnit Organization Locality State Country |  |
| 20 | Low | Active Directory Certificate Services did not start: The Certificate Date Validity Period string in the registry value HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\CertSvc\Configuration\%1\ValidityPeriod is invalid. Valid strings are "Seconds", "Minutes", "Hours", "Days", "Weeks", "Months" and "Years". |  |
| 21 | Low | Active Directory Certificate Services could not process request %1 due to an error: %2. The request was for %3. |  |
| 22 | Low | Active Directory Certificate Services could not process request %1 due to an error: %2. The request was for %3. Additional information: %4 |  |
| 23 | Low | Active Directory Certificate Services could not process request %1 due to an error: %2. The request was for %3. The certificate would contain an encoded length that is potentially incompatible with older enrollment software. Submit a new request using different length input data for the following field: %4 |  |
| 25 | Low | Active Directory Certificate Services revoked the certificate for request %1 for %2. |  |
| 26 | Low | Active Directory Certificate Services for %1 was started.%2%3 |  |
| 27 | Low | Active Directory Certificate Services did not start: Hierarchical setup is incomplete. Use the request file in %1.req to obtain a certificate for this Certificate Server, and use the CA administration tool to install the new certificate and complete the installation. |  |
| 28 | Low | Active Directory Certificate Services did not start: The Certificate Revocation List Period string is invalid in the registry value HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\CertSvc\Configuration\%1\CRLPeriod. Valid strings are "Seconds", "Minutes", "Hours", "Days", "Weeks", "Months" and "Years". | Not available in Microsoft Windows Server 2008® |
| 29 | Low | Active Directory Certificate Services issued a new Certificate Revocation List for %1. | Not available in Microsoft Windows Server 2008® |
| 33 | Low | Active Directory Certificate Services did not start: Could not create the Certificate Server service thread for %1. %2. |  |
| 34 | Low | Active Directory Certificate Services did not start: Could not initialize RPC for %1. %2. |  |
| 35 | Low | Active Directory Certificate Services did not start: Could not initialize OLE for %1. %2. |  |
| 38 | Low | Active Directory Certificate Services for %1 was stopped. |  |
| 39 | Low | Active Directory Certificate Services did not start: The CA DCOM class for %1 could not be registered. %2. Use the services administration tool to change the CA logon context. |  |
| 40 | Low | Active Directory Certificate Services did not start: Could not initialize DCOM class factories for %1. %2. |  |
| 41 | Low | Active Directory Certificate Services did not start: Could not initialize DCOM Security Context for %1. %2. | Not available in Microsoft Windows Server 2008® |
| 42 | Low | Could not build a certificate chain for CA certificate %3 for %1. %2. |  |
| 43 | Low | The "%1" Policy Module "%2" method caused an exception at address %4. The exception code is %3. |  |
| 44 | Low | The "%1" Policy Module "%2" method returned an error. %5 The returned status code is %3. %4 |  |
| 45 | Low | The "%1" Exit Module "%2" method caused an exception at address %4. The exception code is %3. |  |
| 46 | Low | The "%1" Exit Module "%2" method returned an error. %5 The returned status code is %3. %4 |  |
| 48 | Low | Revocation status for a certificate in the chain for CA certificate %3 for %1 could not be verified because a server is currently unavailable. %2. |  |
| 49 | Low | A certificate in the chain for CA certificate %3 for %1 could not be verified because no information is available describing how to check the revocation status. %2. |  |
| 51 | Low | A certificate in the chain for CA certificate %3 for %1 has been revoked. %2. |  |
| 52 | Low | Active Directory Certificate Services issued a certificate for request %1 for %2. Additional information: %3 |  |
| 53 | Low | Active Directory Certificate Services denied request %1 because %2. The request was for %3. Additional information: %4 |  |
| 54 | Low | Active Directory Certificate Services left request %1 pending in the queue for %2. Additional information: %3 |  |
| 55 | Medium | Active Directory Certificate Services unrevoked the certificate for request %1 for %2. | Not available in Microsoft Windows Server 2008® |
| 56 | Low | Active Directory Certificate Services denied request %1. The request was for %2. |  |
| 57 | Low | Active Directory Certificate Services denied request %1. The request was for %2. Additional information: %3 |  |
| 58 | Low | A certificate in the chain for CA certificate %3 for %1 has expired. %2. |  |
| 59 | Low | Active Directory Certificate Services did not start: Could not connect to the Active Directory for %1. %2. |  |
| 60 | High | Active Directory Certificate Services refused to process an extremely long request from %1. This may indicate a denial-of-service attack. If the request was rejected in error, modify the MaxIncomingMessageSize registry parameter via certutil -setreg CA\MaxIncomingMessageSize <bytes>. Unless verbose logging is enabled, this error will not be logged again for 20 minutes. |  |
| 62 | Low | Active Directory Certificate Services had problems loading valid CRL publication values and has reset the CRL publication to its default settings. |  |
| 63 | Low | Active Directory Certificate Services did not start: %1 %2. |  |
| 64 | Low | Active Directory Certificate Services cannot publish enrollment access changes to Active Directory. |  |
| 65 | Low | Active Directory Certificate Services could not publish a Base CRL for key %1 to the following location: %2. %3.%5%6 |  |
| 66 | Low | Active Directory Certificate Services could not publish a Delta CRL for key %1 to the following location: %2. %3.%5%6 |  |
| 67 | Low | Active Directory Certificate Services made %1 attempts to publish a CRL and will stop publishing attempts until the next CRL is generated. |  |
| 68 | Low | Active Directory Certificate Services successfully published Base CRL(s). |  |
| 69 | Low | Active Directory Certificate Services successfully published Delta CRL(s). |  |
| 70 | Low | Active Directory Certificate Services successfully published Base and Delta CRL(s). |  |
| 71 | Low | Active Directory Certificate Services successfully published Base CRL(s) to server %1. |  |
| 72 | Low | Active Directory Certificate Services successfully published Delta CRL(s) to server %1. |  |
| 73 | Low | Active Directory Certificate Services successfully published Base and Delta CRL(s) to server %1. |  |
| 74 | Low | Active Directory Certificate Services could not publish a Base CRL for key %1 to the following location on server %4: %2. %3.%5%6 |  |
| 75 | Low | Active Directory Certificate Services could not publish a Delta CRL for key %1 to the following location on server %4: %2. %3.%5%6 |  |
| 76 | Low | The "%1" Policy Module logged the following information: %2 |  |
| 77 | Low | The "%1" Policy Module logged the following warning: %2 |  |
| 78 | Low | The "%1" Policy Module logged the following error: %2 |  |
| 79 | Low | Active Directory Certificate Services could not publish a Certificate for request %1 to the following location: %2. %3.%5%6 |  |
| 80 | Low | Active Directory Certificate Services could not publish a Certificate for request %1 to the following location on server %4: %2. %3.%5%6 |  |
| 81 | Low | Active Directory Certificate Services key archival is only supported on Advanced Server. %1 |  |
| 82 | Low | Active Directory Certificate Services could only verify %1 of %2 key recovery certificates required to enable private key archival. Requests to archive private keys will not be accepted. |  |
| 83 | Low | Active Directory Certificate Services encountered an error loading key recovery certificates. Requests to archive private keys will not be accepted. %1 |  |
| 84 | Low | Active Directory Certificate Services will not use key recovery certificate %1 because it could not be verified for use as a Key Recovery Agent. %2 %3 |  |
| 85 | Low | Active Directory Certificate Services ignored key recovery certificate %1 because it could not be loaded. %2 %3 |  |
| 86 | Low | Active Directory Certificate Services could not use the provider specified in the registry for encryption keys. %1 |  |
| 87 | Low | Active Directory Certificate Services could not use the default provider for encryption keys. %1 |  |
| 88 | Low | Active Directory Certificate Services switched to the default provider for encryption keys. %1 |  |
| 90 | Low | %1: Active Directory Certificate Services detected an exception at address %2. Flags = %3. The exception is %4. |  |
| 91 | Low | Could not connect to the Active Directory. Active Directory Certificate Services will retry when processing requires Active Directory access. |  |
| 92 | Low | Active Directory Certificate Services could not update security permissions. %1 |  |
| 93 | Low | The certificate (#%1) of Active Directory Certificate Services %2 does not exist in the certificate store at CN=NTAuthCertificates,CN=Public Key Services,CN=Services in the Active Directory's configuration container. The directory replication may not be completed. |  |
| 94 | Low | Active Directory Certificate Services %1 can not open the certificate store at CN=NTAuthCertificates,CN=Public Key Services,CN=Services in the Active Directory's configuration container. |  |
| 95 | High | Security permissions are corrupted or missing. The Active Directory Certificate Services may need to be reinstalled. |  |
| 96 | Low | Active Directory Certificate Services could not create an encryption certificate. %1. %2. |  |
| 97 | Low | Active Directory Certificate Services %1 will reduce the maximum lifetime of the issued certificate for request %2 because the CA certificate lifetime is shorter than the registry validity period. Consider renewing the CA certificate or reducing the registry validity period. |  |
| 98 | Low | Active Directory Certificate Services encountered errors validating configured key recovery certificates. Requests to archive private keys will no longer be accepted. |  |
| 99 | Low | Active Directory Certificate Services could not create cross certificate %1 to certify its own root certificates. %2. %3. |  |
| 100 | Low | Active Directory Certificate Services did not start: Could not load or verify the current CA certificate. %1 %2. |  |
| 101 | Low | Active Directory Certificate Services created CA cross certificate %2 for %1. |  |
| 102 | Low | Active Directory Certificate Services could not create cross certificate %1 to certify its own root certificates. The %2 extension is inconsistent. %3. %4. |  |
| 103 | Low | Active Directory Certificate Services added the root certificate of certificate chain %1 to the downloaded Trusted Root Certification Authorities Enterprise store on the CA computer. This store will be updated from the Certification Authorities container in Active Directory the next time Group Policy is applied. To verify that the CA certificate is published correctly in Active Directory, run the following command: certutil -viewstore "%2" (you must include the quotation marks when you run this command). If the Root CA certificate is not present, use the Certificates console on the Root CA computer to export the certificate to a file, and then run the following command to publish it to Active Directory: Certutil -dspublish %certificatefilename% Root. |  |
| 104 | Low | Active Directory Certificate Services published certificate %1 to %2. |  |
| 105 | Low | Active Directory Certificate Services deleted invalid certificate %1 from %2. |  |
| 106 | Low | Active Directory Certificate Services cannot add certificate %1 to %2. %3. %4. |  |
| 107 | Low | Active Directory Certificate Services cannot delete invalid certificate %1 from %2. %3. %4. |  |
| 108 | Low | Active Directory Certificate Services could not delete a Certificate for request %1 from the following location: %2. %3.%5%6 |  |
| 109 | Low | Active Directory Certificate Services could not delete a Certificate for request %1 from the following location on server %4: %2. %3.%5%6 |  |
| 110 | Low | Active Directory Certificate Services could not initialize the performance counters. |  |
| 111 | Low | Active Directory Certificate Services upgrade failed because the upgrade path could not be determined. %1 |  |
| 112 | Low | Active Directory Certificate Services upgrade failed because information required for the upgrade was unavailable. %1 |  |
| 113 | Low | A portion of the Active Directory Certificate Services upgrade failed: Could not create CertEnroll folder and/or shared folder with proper permissions. %1 |  |
| 114 | Low | A portion of the Active Directory Certificate Services upgrade failed: Could not create virtual roots. %1 |  |
| 115 | Low | A portion of the Active Directory Certificate Services upgrade failed: Could not update server registry entries. %1 |  |
| 116 | Low | A portion of the Active Directory Certificate Services upgrade failed: Could not create web configuration file. %1 |  |
| 117 | Low | A portion of the Active Directory Certificate Services upgrade failed: Could not create revocation page. %1 |  |
| 118 | Low | A portion of the Active Directory Certificate Services upgrade failed: Could not upgrade key containers. %1 |  |
| 119 | Low | A portion of the Active Directory Certificate Services upgrade failed: Could not register CertSrv request. %1 | Not available in Microsoft Windows Server 2008® |
| 120 | Low | A portion of the Active Directory Certificate Services upgrade failed: Could not register CertSrv admin. %1 | Not available in Microsoft Windows Server 2008® |
| 121 | Low | A portion of the Active Directory Certificate Services upgrade failed: Could not install new templates. %1 |  |
| 122 | Low | A portion of the Active Directory Certificate Services upgrade failed: Could not update service description. %1 |  |
| 123 | Low | A portion of the Active Directory Certificate Services upgrade failed: Could not update security settings. %1 |  |
| 124 | Low | Active Directory Certificate Services upgrade succeeded. Active Directory Certificate Services settings have been upgraded successfully. |  |
| 125 | Low | Active Directory Certificate Services upgrade failed. Active Directory Certificate Services settings have not been upgraded. %1 |  |
| 126 | Low | Current information about advanced features supported by this CA is not available from the domain controller. Stop and restart Certificate Services in order to update this information. %1 |  |
| 127 | Low | Key recovery certificate %1 is about to expire soon and will not be used upon expiration. Contact your administrator to renew this certificate. %2 %3 |  |
| 128 | Low | An Authority Key Identifier was passed as part of the certificate request %1. This feature has not been enabled. To enable specifying a CA key for certificate signing, run: "certutil -setreg ca\UseDefinedCACertInRequest 1" and then restart the service. |  |
| 129 | Low | An invalid OID has been detected in the EnabledEKUForDefinedCACert configuration setting. To resolve, run: "certutil -getreg ca\EnabledEKUForDefinedCACert" to identify the invalid OID and correct it. The default OID ("1.3.6.1.5.5.7.3.9") will be used. |  |
| 130 | Low | Active Directory Certificate Services could not create a certificate revocation list. %1. This may cause applications that need to check the revocation status of certificates issued by this CA to fail. You can recreate the certificate revocation list manually by running the following command: "certutil -CRL". If the problem persists, restart Certificate Services. |  |
| 131 | Low | An invalid OID has been detected in the EKUOIDsForPublishExpiredCertInCRL configuration setting. To resolve, run: "certutil -getreg ca\EKUOIDsForPublishExpiredCertInCRL" to identify the invalid OID and correct it. The default OIDs ("1.3.6.1.5.5.7.3.3" and "1.3.6.1.4.1.311.61.1.1") will be used. |  |
| 132 | Low | The CA was unable to perform a decryption operation. This error can occur when an advanced encryption algorithm such as Advanced Encryption Standard (AES) is used and the CA has not been configured to use a CryptoAPI Next Generation (CNG) key storage provider. If this error occurred during certificate enrollment, check the Certificate Template to ensure that advanced encryption for key archival is not enabled. |  |
| 133 | Low | The CA failed to encode a server extension required to validate a certificate or certification revocation list (CRL). The CA will not issue any certificates or CRLs that do not contain this extension. To correct this problem, use the CA snap-in to remove any Unicode characters in the URLs for the AIA, CDP, and IDP extensions, then restart the CA. | Not available in Microsoft Windows Server 2012® |

## Registry Values to Monitor

The following events are recommendations for advanced monitoring of registry changes that affect the security of a CA. While many of these same alerts are generated when enabling auditing on the CA, there are cases where values can be changed and no alert is generated. In those cases, registry auditing can be enabled and the following events can be monitored for.

In the table below, “Event ID” is the current Microsoft Windows® event ID for versions of Microsoft Windows® currently in mainstream support. “Text to Alert On” is the text to search for within the event body when an alert is generated. “Potential Criticality” identifies whether the event should be considered of low, medium or high criticality in detecting attacks. The event summary contains a brief description of the event.

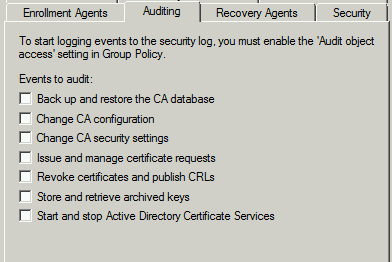
| **Event ID** | **Text to Alert On** | **Potential Criticality** | **Event Summary** |
| --- | --- | --- | --- |
| 4657 | "AuditFilter" | High | The audit filter controls which Microsoft Windows® Security Auditing events are logged. Changing the audit filter may indicate an attacker attempting to disable logging prior to performing a certificate operation. Normally the audit filter is configured when the CA is created and not changed after. |
| 4657 | "EKUOIDsForPublishExpiredCertInCRL" | High | This value controls what types of certificates remain on a CRL even after the certificate expires. An attacker could remove specific certificate types (such as Code Signing) that would allow a previously revoked certificate that malware was signed with to validate successfully again after the next CRL publication.  This value is not changed during normal CA operation. |
| 4657 | “EditFlags” | Medium | Alert if the new value enables EDITF\_ATTRIBUTESUBJECTALTNAME2. This can be identified by taking the value found in the “New Value” field and performing a bitwise “AND” operation with 262144 (the decimal value for the bitmask for the EDITF\_ATTRIBUTESUBJECTALTNAME2 value). Adding this value will allow any certificate request to contain arbitrary alternative names. |
| 4657 | "KRACertHash" | Medium | This will happen rarely in normal operations and an attacker who has control of a valid KRA certificate could assign it to a CA to get access to any private keys that are subsequently archived on the CA. |
| 4657 | "RoleSeparationEnabled" | Medium | Role separation allows for a CA to tightly control the rights of a specific user and enforce that all users can only have one role on the system (CA Admin, Cert Issuer, administrator, Auditor). A local administrator can always disable role separation, which may allow an account who should not have rights to perform an operation to be eligible for those rights. |
| 4657 | "Security" | High | This alert is raised when the permissions on the CA are modified. These permissions control which users and groups are allowed to Read CA information, issue certificates, manage the CA settings and enroll for certificates from a given CA. Modification could allow an attacker to give privileges to an unwanted account for enrollment.  This is a similar alert to 4882. |
| 4657 | "ExitModules","Active" | High | Indicates a change to the default exit module(s) being used by the CA. Exit modules allow additional actions to be performed by the CA after an event occurs (issuance, revocation, CRL publishing, etc.). Additions/deletions of exit modules occur very infrequently in normal operations and may indicate tampering with the CA. |
| 4657 | "PolicyModules","Active" | High | Indicates a change to the active policy module being used by the CA. The policy module control certificate issuance and is changed very infrequently in normal operations. |

# Appendix B: Certification Authority Audit Filter

The CA audit filter is a bitmask value representing the seven different audit categories that can be enabled. If all values are enabled, the audit filter will have a value of 127.

|  |  |
| --- | --- |
| **Value (Decimal)** | **Setting** |
| 1 | Start and stop Active Directory® Certificate Services |
| 2 | Back up and restore the CA database |
| 4 | Issue and manage certificate requests |
| 8 | Revoke certificates and publish CRLs |
| 16 | Change CA security settings |
| 32 | Store and retrieve archived keys |
| 64 | Change CA configuration |

The CA audit filter can be set through the CA snap-in GUI or via the command line. To set the audit filter via the GUI:

1. Open the CA snap-in (certsrv.msc).
2. Right-click the CA and click **Properties.**
3. Click the **Auditing** tab and select the desired values.  
     
   

To set the audit filter via the command line, run the following command from an elevated command prompt:

**certutil –setreg CA\AuditFilter <value>**

**net stop certsvc && net start certsvc**

For example, **certutil –setreg CA\AuditFilter 18** will set the audit filter to include events covered under “Change CA security settings” and “Backup and restore the CA database.”

# Appendix C: Delegating Active Directory PKI Permissions

In order to install an enterprise root or subordinate CA in Active Directory®, typically the account performing the installation should be a member of the Enterprise Admins (EA) group for the forest, as well as a member of the local administrators group on the CA computer. In some cases it may be desirable to delegate the permissions required to perform installation, or other PKI related activities to accounts that are not members of privileged Active Directory® groups such as Enterprise Admins or Domain Admins. Regardless of which groups are used to perform PKI activities, group membership should be tightly controlled and monitored.

The instructions below do not address the permissions required to install other AD CS roles such as NDES. It might not be possible to perform other AD CS-related tasks based on the permissions delegated through the steps below.

## Permissions for Enterprise CA Installation

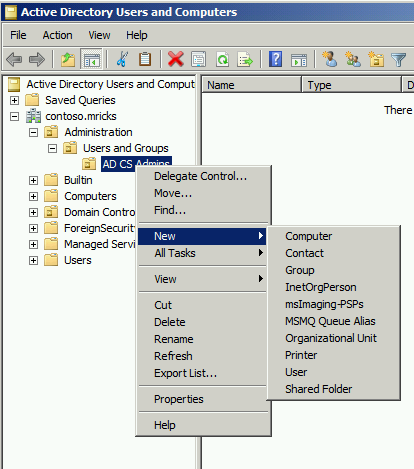
Delegating rights to a non-Enterprise Admin user to perform complete CA installations requires delegating the following permissions:

* Rights to create new objects underneath the **Public Key Infrastructure** container underneath the **Configuration** partition. This includes creating new Certificate Template objects, adding objects to the **Certification Authorities** node, the **Enrollment Services** node, the **AIA** node, and the **CDP** node.
* Rights to add members to the **Cert Publishers** groups in each domain in the forest. The computer account of the CA is added to the **Cert Publishers** group of its domain during the CA installation.
* Rights to add members to the **Pre-Windows 2000 Compatible Access** group
* Membership in the local administrators group of the soon-to-be CA

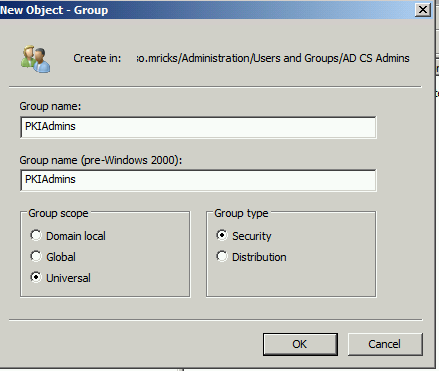
**Note:** If installation of the first CA in a forest is done using delegated permissions, it will be necessary to install the default templates separately as an Enterprise Admin. This can be done prior to installing the CA by running the command **certutil –installdefaulttemplates**.

### Delegating Rights to the Public Key Infrastructure Container

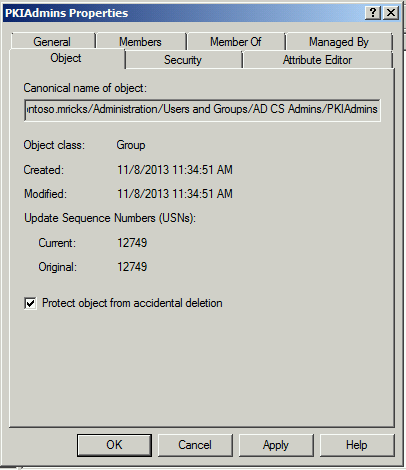
1. In the Organizational Unit (OU) where you will be storing the groups and/or accounts that will be managing the PKI, right-click the OU where you want to create the group, click **New** and click **Group**.



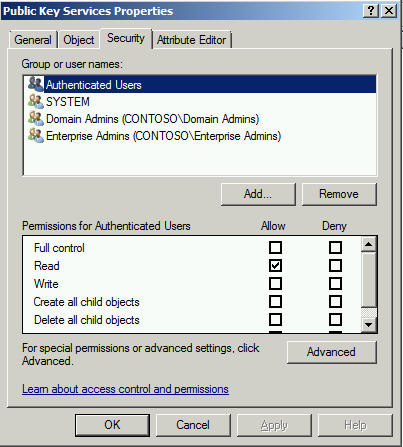
1. In the **New Object – Group** dialog box, enter a name for the group. If you plan to have certification authorities in multiple domains in your forest, make it a universal group. Otherwise, create a global group. Click **OK** to create the group.



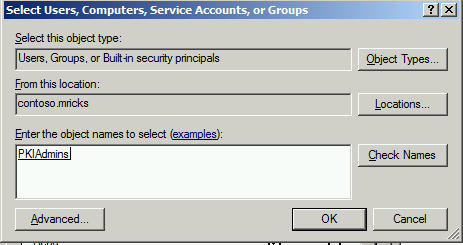
1. Right-click the group you just created, click **Properties**, and click the **Object** tab. In the group’s **Object property** dialog box, select **Protect object from accidental deletion**, which will not only prevent otherwise-authorized users from deleting the group, but also from moving it to another OU unless the attribute is first deselected.



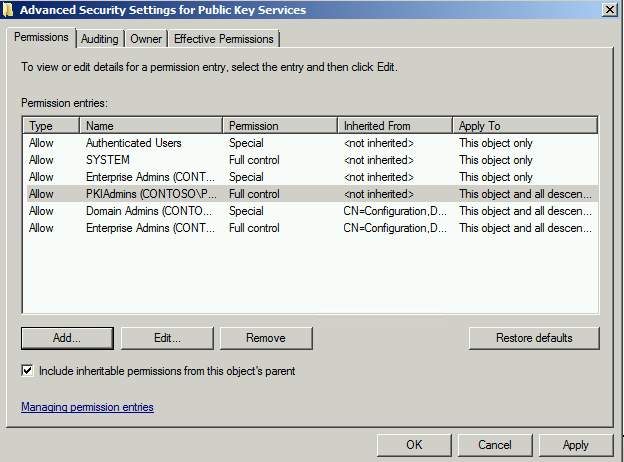
1. Open **Active Directory Sites and Services** with an account in the Enterprise Admins group.
2. Click the **View** menu option and select **Show Services Node.**
3. Under the **Services** node, right-click **Public Key Services,** click **Properties** and click the **Security** tab.



1. Click **Advanced**.
2. Click **Add...** and search for the newly created management group and click **OK.**



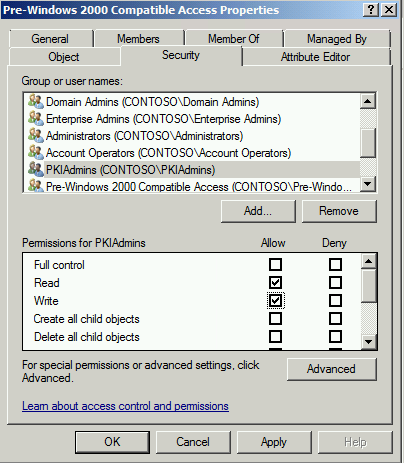
1. Grant the new management group full control and click **OK**.



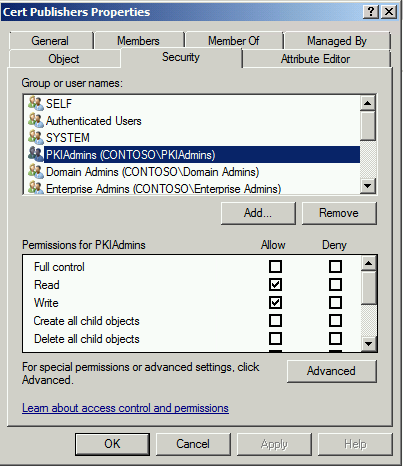
1. Click **OK** to close the **Public Key Services Properties** dialog box.

### Delegating Group Permissions

1. Open **Active Directory Users and Computers** with an account that has rights to modify security permissions for **Pre-Windows 2000 Compatible Access** and **Cert Publishers** groups.
2. Under **Builtin**, right-click **Pre-Windows 2000 Compatible Access** and click **Properties.**
3. Click the **Security** tab. Click **Add…** and select the newly created management group. Grant the group **Read** and **Write** access and click **OK**.



1. Under **Users,** right click **Cert Publishers** and click **Properties**.
2. Click the **Security** tab. Click **Add…** and select the newly created management group. Grant the group **Read** and **Write** access and click **OK**.



1. If you have a multi-domain forest, you will need to repeat the step of granting **Write** access to the management group to **Cert Publishers** for each domain in the forest. Each domain has its own **Cert Publishers** group. During installation, the CA computer account will be added to the **Cert Publishers** group in the domain in which the CA resides.

## Permissions for Managing Certificate Templates

For guidance on implementing delegation for management of certificate templates, refer to [Administering Certificate Templates](http://technet.microsoft.com/en-us/library/cc725621(v=WS.10).aspx).

# Appendix D: Glossary of Terms

For the NIST glossary of terms, refer to [Glossary of Security Information Terms](http://csrc.nist.gov/publications/PubsNISTIRs.html#NIST-IR-7298) in NISTIR 7298 Rev. 2.

The Microsoft Security Glossary is located at <http://msdn.microsoft.com/en-us/library/ms721607(v=vs.85).aspx>.

The Windows Server 2008 R2® glossary is located at <http://technet.microsoft.com/en-us/library/dd919232(v=ws.10).aspx>.

**Policy Terms**

| **Term** | **Term Definition** |
| --- | --- |
| **Asset Owner** | The creator, generator, originator, or primary possessor of an Asset, or agent(s) to which the Asset Owner has given consent to act as a fiduciary with regard to specific assets, according to a documented agreement. |
| **Certificate Policy (CP)** | A specialized form of administrative policy tuned to electronic transactions performed during certificate management. A Certificate Policy addresses all aspects associated with the generation, production, distribution, accounting, compromise recovery, and administration of digital certificates. Indirectly, a certificate policy can also govern the transactions conducted using a communications system protected by a certificate-based security system. By controlling critical certificate extensions, such policies and associated enforcement technology can support provision of the security services required by particular applications. |
| **Certificate-Related Information** | Information, such as a subscriber's postal address, that is not included in a certificate. May be used by a CA managing certificates. |
| **Certification Practice Statement (CPS)** | A statement of the practices that a CA employs in issuing, suspending, revoking, and renewing certificates and providing access to them, in accordance with specific requirements (i.e., requirements specified in this Certificate Policy, or requirements specified in a contract for services). |
| **Key Recovery** | Mechanisms and processes that allow authorized parties to retrieve the cryptographic key used for data confidentiality. |
| **Online Certificate Status Protocol (OCSP)** | An online protocol used to determine the status of a public key certificate. |
| **Rekey (a certificate)** | To change the value of a cryptographic key that is being used in a cryptographic system application; this normally entails issuing a new certificate on the new public key. |
| **Renew (a certificate)** | The act or process of extending the validity of the data binding asserted by a public key certificate by issuing a new certificate. |
| **Revoke a certificate** | To prematurely end the operational period of a certificate effective at a specific date and time. |
| **Standard Operating Procedure (SOP)** | A document that describes how to implement a configuration or execute a process that is considered mandatory for a specific PKI. SOPs serve as the documented record of a given team's compliance with relevant Policy and/or requirement statements. |
| **Standards** | Mandatory prerequisites for all PKIs. Standards are subordinate to Policy statements, and are designed to provide more explicit definition of Policy intent. |
| **Update (a Certificate)** | The act or process by which data items bound in an existing public key certificate, especially authorizations granted to the subject, are changed by issuing a new certificate. |

**PKI Objects**

| **Term** | **Term Definition** |
| --- | --- |
| **Active Directory® Certificate Services** | Active Directory® Certificate Services (AD CS) is an Identity and Access Control security technology that provides customizable services for creating and managing public key certificates used in software security systems that employ public key technologies. |
| **Authority Information Access (AIA)** | Specifies where to find up-to-date certificates for the CA. |
| **(Certificate) Repository** | A database containing information and data relating to certificates as specified in a CP; may also be referred to as a directory. |
| **(Certificate) Trust List** | The collection of trusted certificates used by Relying Parties to authenticate other certificates. |
| **Certificate Chain (Certification Path)** | A chain of certificates consisting of the subscriber certificate, issuing CA certificate, intermediate CA certificate(s) and the root CA certificate. |
| **Certificate Profile** | Detailed description of the structure, components, and the origin of the data in the certificate |
| **Certificate Revocation Lists (CRLs)** | A list of certificates (or more specifically, a list of serial numbers for certificates) that have been revoked, and therefore, entities presenting those (revoked) certificates should no longer be trusted |
| **Certificate Status Service** | A trusted entity that provides online verification to a Relying Party of a subject certificate's trustworthiness, and may also provide additional attribute information for the subject certificate. |
| **CRL Distribution Points (CDPs)** | The location where you can download the latest CRL. |
| **Encryption Certificate** | A certificate containing a public key that is used to encrypt electronic messages, files, documents, or data transmissions, or to establish or exchange a session key for these same purposes. |
| **Extended Key Usage (EKU)** | Defines what a certificate will be used for. |
| **Hardware Security Module (HSM)** | A hardware security module is a physical computing device that safeguards and manages digital keys for strong authentication and provides cryptographic operations processing. |
| **Intermediate Certification Authority** | A CA that is subordinate to another CA, and has a CA subordinate to itself. |
| **Issuing Certification Authority** | A subordinate CA that issues certificate to end user and computers (certificate subjects). |
| **Object Identifier (OID)** | A globally unique value associated with an object to unambiguously identify it used in Abstract Syntax Notation (ASN.1) |
| **Registration Authority (RA)** | A trusted entity that establishes and vouches for the identity of a Subscriber to a CA. The RA may be an integral part of a CA, or it may be independent of a CA, but it has a relationship to the CA. |
| **Root Certification Authority** | The CA at the top of a PKI hierarchy that is explicitly trusted by all subscribers and relying parties whose public key serves as the most trusted datum (i.e., the beginning of trust paths) for a security domain. |
| **Self-signed Certificate** | A certificate that 1: uses its public key to verify its own signature; 2: the subject name is identical to the issuer name. |
| **Signature Certificate** | A public key certificate that contains a public key intended for verifying digital signatures rather than encrypting data or performing any other cryptographic functions. |
| **Subordinate Certification Authority** | A CA whose certificate signature key is certified by another CA, and whose activities are constrained by that other CA. |
| **Superior Certification Authority** | A CA that has certified the certificate signature key of another CA, and who constrains the activities of that CA. |

# Appendix E: PKI Basics

A PKI is a collection of hardware, software, personnel, and operating procedures that issue and manage certificates. The certificate binds a public key to a named subject, which allows relying parties to trust signatures or assertions made by the subject and expressed in the certificate.

In their most basic form, digital certificates bind identities to cryptographic key pairs that are mathematically related and can be used to provide confidentiality, integrity, and nonrepudiation for other systems and processes. These key pairs are typically referred to as public and private keys. Public keys are often widely distributed, while private keys are available only to the individual or system that owns them. A certificate provides a means to tie an identity of a user or system to a specific key pair. By using these keys and certificates, the following security functions can be performed:

* **Digital Signatures** –A digital signature can provide assurance that a piece of data such as a document, executable, or script came from a specific source and has not been tampered with since it came from that source.
* **Authentication** –PKI provides a strong method of authenticating users or systems. By asking a user or system to perform a digital signature operation with their private key, you get assurance that the entity presenting you with the certificate has possession of the matching private key, proving they are who they are asserting to be.
* **Encryption** –PKI provides the capability to encrypt data that can only be decrypted by someone in possession of the private key. If you want to send someone encrypted data, you can obtain their public key and encrypt the data with their public key. Since the public and private key are mathematically related, they can use the related private key to decrypt the data.

# Appendix F: List of Recommendations by Impact Level

Below is a complete list of all recommendations made throughout this paper, classified according to the [impact level](#_Determining_the_Level) of the CA. Recommendations are broken out according to the chapter in which they were found. Some of the recommendations are strategic in nature, and require planning and potentially redesign to implement, while some are tactical and focused on specific components and infrastructure.

## Planning a CA Hierarchy

|  |  |  |
| --- | --- | --- |
| Recommendation | Tactical or Strategic | Impact Level |
| Do not use a one-tier hierarchy | Strategic | High Internal |
| Plan for upcoming PKI uses cases as part of the initial design | Strategic | Medium |

## Physical Security

| Recommendation | Tactical or Strategic | Rating |
| --- | --- | --- |
| Leverage existing data center controls for physical security where possible | Strategic | Medium |
| Track and audit requests for physical access to PKI assets | Tactical | High Internal |
| Use biometrics as an authentication mechanism to access PKI assets | Tactical | High Internal |
| Prevent tailgating to sensitive areas where PKI assets are stored | Tactical | High Internal |
| Use alarm systems to detect access to PKI assets | Tactical | High Internal |
| Use cameras to monitor physical access to PKI assets | Tactical | High Internal |
| Geographically separate primary and backup sites | Strategic | High Internal |
| Use obscurity carefully to not disclose unnecessary information about PKI assets | Tactical | High Internal |

## PKI Process Security

| Recommendation | Tactical or Strategic | Rating |
| --- | --- | --- |
| Develop a Certificate Policy to govern the use of the PKI | Strategic | High External |
| Develop a formal Certification Practice Statement | Strategic | High External |
| Document issuance controls and certificate usage (informal CP/CPS) | Tactical | High Internal |
| Document CA standard operating procedures | Tactical | High Internal |
| Utilize any existing policy structure to store and maintain PKI policy | Tactical | Medium |
| Involve your policy team in the creation of PKI policy | Strategic | High Internal |
| Involve your legal department in policy creation if your PKI may affect external customers or partners | Strategic | High Internal |
| Form a Policy Authority to provide governance for the PKI | Strategic | High Internal |
| Formalize the work performed by the Policy Authority with auditable change control and meeting minutes | Tactical | High Internal |
| Meet regularly as a Policy Authority to review and update the PKI policy | Tactical | High Internal |
| Establish formal PKI roles and responsibilities and assign specific individuals to each roles | Tactical | High Internal |
| Provide role specific training for all individuals responsible for the PKI | Strategic | Medium |
| Vet individuals who fill trusted roles with a comprehensive background check (in accordance with local privacy law) or other mechanism | Strategic | High Internal |
| Perform formal key ceremonies that follow a script and include a witness | Tactical | High Internal |

## Technical Controls for Securing PKI

| Recommendation | Tactical or Strategic | Rating |
| --- | --- | --- |
| Create baseline system configurations for CA and RA systems | Tactical | Medium |
| Disable CD-ROM Autoplay | Tactical | Medium |
| Rename local administrator and guest accounts | Tactical | Medium |
| Disable local administrator and guest accounts | Tactical | Medium |
| Use a distinct password for the local administrator account that is not used on other systems | Tactical | Medium |
| Enable the Windows Firewall with Advanced Security and block all traffic that is not required | Tactical | Medium |
| Disable services that are not required for the CA to function | Tactical | Medium |
| Disable LM and NTLMv1 authentication protocols | Tactical | Medium |
| Only install software that is necessary for the CA to perform its function | Tactical | Medium |
| Disable Direct Memory Access (DMA) devices | Tactical | Medium |
| Disable Remote Desktop Services | Tactical | High Internal |
| Do not install additional server roles on Certification Authorities, such as running a CA on a domain controller | Tactical | Medium |
| Use alternate accounts separate from the standard accounts used on productivity workstations to manage the PKI | Tactical | Medium |
| Update CA regularly using update infrastructure separate from what is used to manage the general Windows server®/workstation population | Strategic | High Internal |
| Prevent access to the internet from CAs | Tactical | Medium |
| Limit local administrator group membership to only users in trusted roles who manage the PKI | Tactical | Medium |
| Remove Enterprise Admins and Domain Admins from local administrators group on CAs | Tactical | Medium |
| Eliminate or limit the number of service accounts with administrative rights on CAs and RAs | Tactical | Medium |
| Enable application whitelisting using AppLocker or another third party application | Tactical | High Internal |
| Use secure administrative hosts or jump hosts to perform remote management tasks | Strategic | High Internal |
| Disable Remote Management Boards on physical servers | Tactical | High Internal |
| Require PKI administrators to use smart cards for all accounts that manage the PKI | Strategic | High Internal |
| Use a Hardware Security Module in offline CAs | Strategic | High Internal |
| Keep offline CAs truly offline, allow only physical access to all components | Tactical | Medium |
| Use only authorized, dedicated devices to transfer files to/from offline CAs | Tactical | Medium |
| Update offline CAs with service packs, security updates specific to CA software, and updates related to system time (time zone changes) | Tactical | Medium |
| Update HSM software and firmware when released | Tactical | Medium |
| Ensure that any activity performed on an offline CA can be traced to an individual, either through individual accounts or additional auditing and surveillance | Strategic | Medium |
| When virtualizing offline CAs, decouple the guest files from the physical hardware so the hardware can be easily replaced | Tactical | Medium |
| When virtualizing offline CAs, use a dedicated host machine that is secured in a locked rack or safe. If dedicated hardware cannot be used, build a clean host OS each time the CA VMs need to be brought online | Tactical | Medium |
| When virtualizing offline CAs, securely build the VM on the dedicated hardware, do not build it on an online host and migrate it to the dedicated hardware | Tactical | Medium |
| Prior to performing any operations on an offline CA, verify the system time is correct. | Tactical | Medium |
| When virtualizing offline CAs, perform regular backups of hard disk files. Securely store the backups along with any required software at a backup site | Tactical | Medium |
| When virtualizing online CAs, limit access to the host to only those who should have access to the PKI | Tactical | High Internal |
| When virtualizing online CAs, use network attached HSMs for key protection | Tactical | High Internal |
| When virtualizing online CAs, continue to take regular CA backups with all data needed to restore the CA | Tactical | Medium |
| If using software keys, protect all key backups (PKCS#12, PFX files) with the same level of protection provided to the CA | Tactical | Medium |
| Do not include backups of the private key as part of the standard backup process. Backup the key(s) as needed and physically protect them by storing in a safe, within a tamper-evident bag and audit all access to the backup | Tactical | Medium |
| Do not connect backup systems directly to the CA. Backup the CA to another location which is backed up regularly to eliminate the need for backup software on the CA | Tactical | High Internal |
| Isolate certificate systems from other systems on the network | Strategic | High External |
| Implement “security zones” to isolate certificate systems based on their criticality and relationship to each other | Strategic | High External |
| Only allow inbound and outbound connections that are necessary for the CA and supporting systems to function | Tactical | Medium |
| Restrict access to network HSM devices to only the systems that utilize them | Tactical | High Internal |
| Restrict management access to originate from a limited set of administrative hosts | Strategic | High Internal |
| Control “enroll” access to certificate templates and only provide the access to accounts that require the certificate | Tactical | Medium |
| Remove unused certificate templates from CAs | Tactical | Medium |
| Use additional enrollment controls for templates that allow you to specify the subject in the request | Tactical | Medium |
| Do not use the EDITF\_ATTRIBUTESUBJECTALTNAME2 flag on any CA without additional issuance controls | Tactical | Medium |

## Planning Certificate Algorithms and Usages

| Recommendation | Tactical or Strategic | Rating |
| --- | --- | --- |
| Use 2048 bit and above key length for RSA keys | Strategic | Medium |
| If using ECC for CA keys, use P-256, P-384 or P-521 curves | Strategic | Medium |
| Use RSA 4096 for CA certificates that expire more than 15 years in the future | Strategic | Medium |
| Use the SHA-2 family of hash algorithms | Strategic | Medium |
| Root CA certificate should not be valid for more than 25 years | Strategic | Medium |
| Issuing CA certificates should not be valid for more than 5 years | Strategic | Medium |
| Renew an issuing CA certificate once before replacing the key pair | Strategic | Medium |
| Use certificate expiration events in Windows 8® and Windows Server 2012® and above to assist in expiration notification | Strategic | Medium |
| Match the strength of asymmetric key algorithms with the strength of the hash algorithm | Strategic | Medium |
| Use the correct key usage for each certificate use case | Strategic | Medium |
| Determine the extended key usages for each PKI use case | Strategic | Medium |
| Constrain issuing CAs (use the path length constraint to ensure that CA can only issue end-entity certificates and limit application policies) | Strategic | Medium |

## Protecting CA Keys and Critical Artifacts

|  |  |  |
| --- | --- | --- |
| Recommendation | Tactical or Strategic | Impact Level |
| If using network HSMs for offline CAs, do not connect the HSM to a routable network | Tactical | High Internal |
| Create enough HSM tokens to account for disaster recovery | Strategic | Medium |
| Use tamper-evident containers/packaging to store PKI artifacts such as HSM tokens or backup data | Tactical | High Internal |
| Store PKI artifacts in a climate controlled location | Tactical | Medium |
| Maintain an auditable chain of custody of PKI artifacts | Strategic | High Internal |
| Maintain an inventory of PKI artifacts | Strategic | High Internal |

## Monitoring Public Key Infrastructure

| Recommendation | Tactical or Strategic | Impact Level |
| --- | --- | --- |
| Monitor Active Directory® for changes groups that control access to CAs, membership in the “Cert Publishers” group, changes to privileged and VIP accounts, and unauthorized changes to certificate templates | Tactical | High Internal |
| Record and review physical access events | Tactical | High Internal |
| Record and review all physical access to HSMs | Tactical | High Internal |
| Record and review logs from network equipment that supports PKI | Tactical | High Internal |
| Record and review physical access to PKI artifacts, such as access to safes | Tactical | High Internal |
| Configure Windows® audit policy to enable auditing for Certification Services | Tactical | Medium |
| Monitor changes to the CA registry | Tactical | High Internal |
| Monitor for changes to certificate templates | Tactical | High Internal |

## Compromise Response

|  |  |  |
| --- | --- | --- |
| Recommendation | Tactical or Strategic | Impact Level |
| Identify critical systems and processes that are dependent on PKI | Strategic | Medium |
| Develop a basic plan of action for compromise before a compromise occurs | Strategic | Medium |

Copyright Information

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This white paper is for informational purposes only. Microsoft makes no warranties, express or implied, in this document.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in, or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Microsoft, Active Directory, BitLocker, Hyper-V, Internet Explorer, Windows Vista, Windows, and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks are property of their respective owners.

The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

© 2014 Microsoft Corporation. All rights reserved.