



SECURING AGAINST NEW OFFENSIVE TECHNIQUES ABUSING ACTIVE DIRECTORY CERTIFICATE SERVICE

Our address

Unit 3E-3F, 33-34 Westpoint,
Warple Way, Acton, W3 0RG

Give us a call

0333 939 8080

Send us a message

hello@jumpsec.com

Find out more

www.jumpsec.com

Applying recent offensive security research into adversarial techniques targeting insecure functionality in Active Directory Certificate Service in a defensive context.

Table of contents

1.	Background	3
1.1	Why is this development so significant?	3
1.2	What can organisations do about it?	4
2.	What is Active Directory Certificate Service?	5
2.1	How can Active Directory Certificate Service be abused?	5
3.	What is the risk?	8
3.1	What is the impact of a successful attack?	9
4.	Identifying exposure	10
4.1	Baseline Certificate Authority configuration overview	10
4.2	Exposure assessment: "Am I vulnerable?"	13
4.3	Susceptibility audit: "Where am I vulnerable?"	15
5.	How can organisations protect, detect, and respond to the techniques?	17
5.1	Enable Certificate Authority logging	18
5.2	Implementing defence-in-depth secure architecture	19
5.3	Securely configure Certificate Authority settings	20
5.4	Identify and monitor relevant Event IDs	32
5.5	Plan for a potential compromise	33
6.	Next steps	34

Summary

SpecterOps recently released an offensive security research paper that details techniques enabling an adversary to abuse insecure functionality in Active Directory Certificate Service.

SpecterOps reports that abusing the legitimate functionality of Active Directory Certificate Service will allow an adversary to forge the elements of a certificate to authenticate as any user or administrator in Active Directory. JUMPSEC has highlighted numerous changes that can be made to Active Directory Certificate Service configuration to protect the domain through a defence-in-depth approach.

We at JUMPSEC wanted to understand the defensive application of this offensive research to pre-emptively defend our clients from these techniques before exploitation is observed in the wild. To do this, we utilised our Active Directory lab and attempted to harden the service to reduce the risk of compromise and limit the ability for an attacker to cause harm.

In this article, JUMPSEC has documented the most effective and efficient methods we took to implement the broad defensive guidance in SpecterOps research. In our attempts to harden Active Directory Certificate Service, we have identified ways to harden the environment against compromise, and leverage auditing toolkits to make it easier to identify and remediate areas of exposure.

1. Background

If you are short on time, please go directly to the remediation guidance found [here](#).

Recent research by SpecterOps details a new method of attack in Windows Active Directory targeting the Certificate Service.

Active Directory Certificate Service is Microsoft's PKI implementation that integrates with existing Active Directory forests and provides everything from encrypting file systems to digital signatures and user authentication. SpecterOps' research enables total compromise and robust persistence in susceptible environments.

There have previously been fragments of theoretical research and small-scale practical exploitation of Microsoft PKI implementations. However, SpecterOps are the first to publicly have amassed together a set of novel techniques that target the Certificate Service directly.

SpecterOps have said that they will release offensive security tooling capable of replicating the techniques at [BlackHat USA](#), running July 31st - August 5th. However, JUMPSEC predicts we will see exploitation in the wild before the SpecterOps deadline, as adversaries research and develop their own methods using the guidance.

The discovery of these new methods presents an opportunity for organisations to review and harden their Active Directory environment. Robust Active Directory implementation and configuration are critical components of an effective security strategy for the vast majority of organisations due to widespread reliance on Active Directory - the primary user directory service trusted by 90% of businesses worldwide.

1.1 Why is this development so significant?

Active Directory is the primary repository responsible for authentication and authorisation services for users and devices. This is achieved by creating a series of user roles and associated permissions that govern the actions that a given user account can perform on a specific system. It is often integrated with other single sign-on solutions to extend its reach to services running on non-Microsoft platforms.

Active Directory is frequently abused by attackers due to its highly pivotal nature. Its functionality has grown significantly over time, managing a widening array of services and corresponding user roles, permissions, and accounts, while its implementation with Windows has grown in complexity. Therefore, the risk of misconfigurations, insecure practices, or unintended interactions rendering the service vulnerable to compromise is significantly increased due to the broad attack surface.

The techniques researched by SpecterOps are unique in that they are more likely to punish organisations that have looked to implement more security-conscious configurations of Active Directory beyond a default deployment. As a result, many less mature organisations will not have enabled the Certificate Service, which typically facilitates a more robust approach to authentication and trust.

We are extremely grateful to the research published by SpecterOps, and as always, we are a firm supporter of offensive security research and its role in improving the security baseline for organisations.

1.2 What can organisations do about it?

JUMPSEC has examined SpecterOps research and compiled guidance to prepare the defences and harden the configurations of an environment before adversaries have the opportunity for exploitation.

This guidance leverages much of SpecterOps research, and we have explored and contributed to some of the defensive components by:

- Condensing SpecterOps' research and guidance into concise, actionable next steps for organisations looking to safeguard against the risk
- Providing PowerShell scripts and command-line alternatives to SpecterOps' GUI-based guidance to streamline remediation activities
- Providing step-by-step visuals for some of the unavoidable GUI-based reconfigurations.

The guidance is simple to implement but it is highly particular to Active Directory. It is vital that organisations make use of this opportunity to review the overall effectiveness of their Active Directory deployment to ensure that the inevitable increase in Active Directory-based scrutiny by threat actors does not increase their cyber risk exposure.

This document:

- Establishes what Active Directory Certificate Service is and what it does legitimately.
- Discusses the offensive security research that SpecterOps have undertaken to abuse Active Directory Certificate Service.
- Explores how the threat landscape has changed thanks to new research in Active Directory Certificate Service exploitation, and the new challenges to managing this risk.
- Outlines the actions we have taken and will continue to build upon in the face of this new research.
- Provides a range of recommendations to assess if you are vulnerable and mitigate the risk.

JUMPSEC recommends that organisations action all the recommendations provided to reduce their Active Directory attack surface and mitigate the impact posed by these emerging techniques.

This is a live article and as such will be continually updated as the recommendations are refined and improved. The nature of this security exposure is that no single patch or fix will suffice to eliminate the issue. The only viable strategy at this stage is to implement layered controls to reduce susceptibility and improve resilience by increasing the cost and complexity for an attacker, and enabling effective detection and response in the case of a compromise.

2. What is Active Directory Certificate Service?

Active Directory Certificate Service is Microsoft's implementation of public-key infrastructure (PKI) that integrates with existing Active Directory implementations, and can underpin the cryptography in encrypting file systems, digital signatures, user authentication, and more. Since Windows 2000, the Certificate Service has existed as an optional, configurable element of an Active Directory deployment. While the service is not shipped as default, it is usually enabled by administrators.

A certificate is a signed statement that binds an identity to a public-private key pair. Much of the same philosophy behind SSL certificates can be applied to Active Directory Certificate Service certificates, and the most critical to highlight for a Windows context is that a certificate is all about authentication, permission, and privilege. In other words, a certificate is something an account can carry around and show whenever it wants to do something, like interact with an application or system process.

2.1 How can Active Directory Certificate Service be abused?

Conceptually secure, Active Directory Certificate Service becomes a liability when it is deployed. Research has demonstrated that most Certificate Services are set up with insecure configurations. This is not a consequence of technical inability but a knowledge gap: many administrators of Active Directory Certificate Service did not and do not realise that adjusting a configuration can create a security risk that an adversary can take advantage of in a live client environment.

Certificate Authority servers are a crucial element to this attack. Certificate Authorities are the ultimate arbiter and signer of a certificate request; unfortunately, they are also susceptible to abuse.

When used legitimately, the Certificate Authority receives a certificate signing request from an endpoint, and upon passing certain checks the server will sign this request with the private key and send it along the conveyor belt of other Active Directory Certificate Service processes.

It seems incompatible with cryptographic philosophy that the ultimate authority can be naive when it uses encryption to verify everything. However, SpecterOps have evidenced that Certificate Authorities are poor arbiters because they can be forced to consider additional evidence that may be tampered with: **Subject Alternate Names (SAN)**.

A SAN is an optional extension for a certificate. It allows additional identities to be associated with the certificate. In the Windows environment, a SAN is an extension to the certificate. It can convey a lot of information and of particular interest is the ability for the SAN to convey a **user principal name**. This is a legitimate function of SAN and certificates. SpecterOps have highlighted that the SAN is **incapable** of security sanitisation and that an adversary can arbitrarily 'add' an identity here.

An adversary can poison the SAN extension by supplying a Domain Administrator's details in that SAN field which gets bundled in the certificate request. In return, the Certificate Authority server receives this request and judges it as authentic. The adversary is then gifted a signed certificate, allowing them to authenticate across the Active Directory as a Domain Administrator.

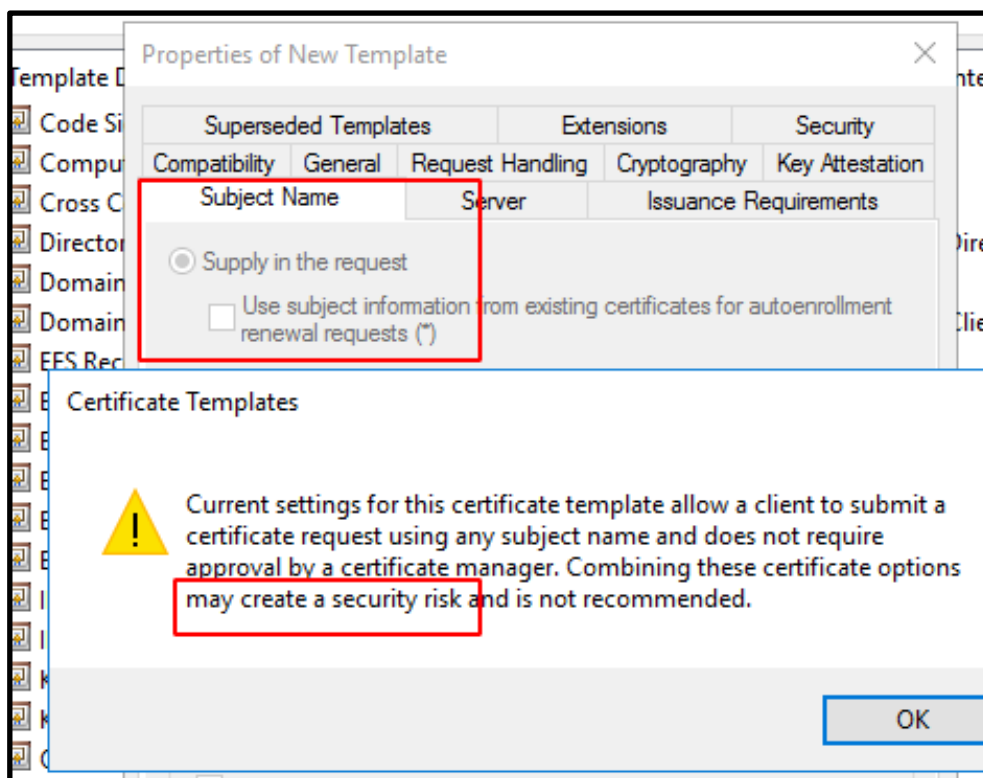


Figure 1. Vulnerable configurations of SAN extension prompt security warning

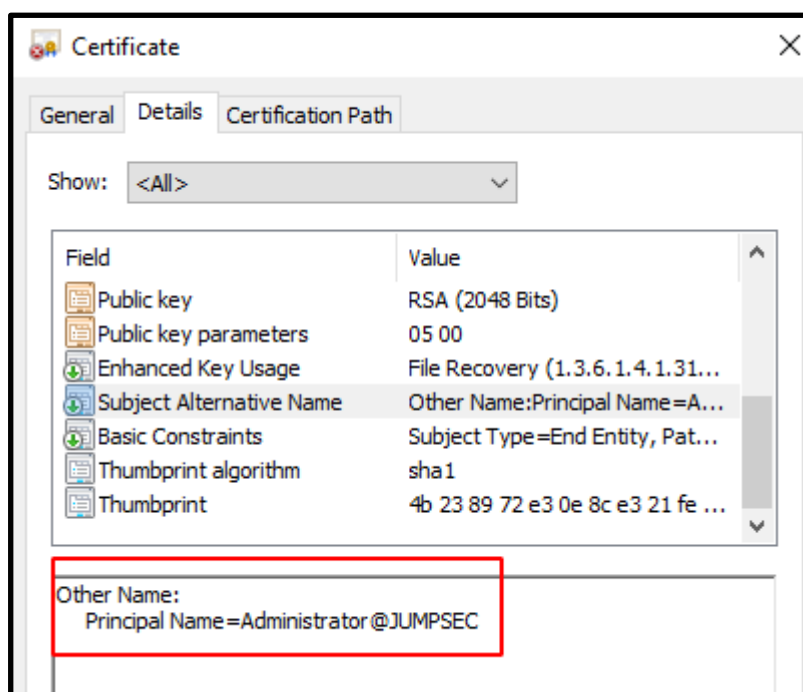


Figure 2. An example of how a user identity can be attached in the SAN

Certificates - an implementation of key-based cryptography - have a relationship with Kerberos and Secure Channel (Schannel), as these are themselves also methods of authentication and communication through cryptography. Certificates have unique integrations with the normal authentication methods that we already know, meaning that insecure, poisoned certificates can interact and integrate with authentication methods across Active Directory.

Through Kerberos, the endpoint will issue a Ticket-Granting-Ticket (TGT) request with their certificate's private key. The endpoint then sends this request to the Domain Controller who then completes some checks and returns a TGT. The "NTAuthCertificates" object is then written to, as it is the root of all trust for any certificate authentication in Active Directory. The NTAuthority Certificate object contains all entries for all Certificate Authorities that can issue certificates for particular forms of authentication. As part of the verification process of a certificate, a Domain Controller checks the NTAuthCertificates object.

Schannel will then instruct the server to request a certificate from the client for authentication. When an endpoint provides a certificate, the Schannel attempts several ways to map this to a user account. One of the methods maps the certificate to an account using the SAN.

3. What is the risk?

The techniques grant an attacker with prior access to the internal network a trivial means of bypassing domain controls and otherwise secure configurations to achieve administrative control of the environment, enabling further malicious activities to be launched against the network from this privileged position.

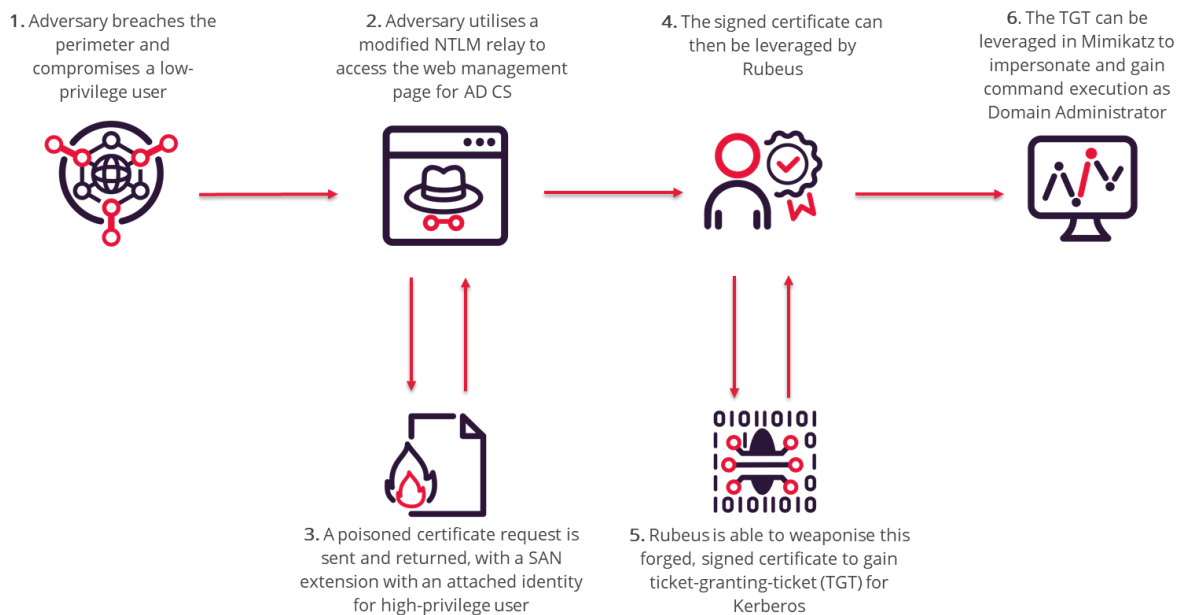


Figure 3. Path an adversary can take to exploit the Certificate Service

The new techniques introduce a reliable method of compromising robust Active Directory configurations which have been hardened against typical methods of compromise.

The visual below details a potential adversarial approach to domain compromise, highlighting two common attack paths alongside the newer explored Active Directory Certificate Service attack. The paths have been defined at a high level and replicate offensive tactics, techniques and procedures observed in the wild.

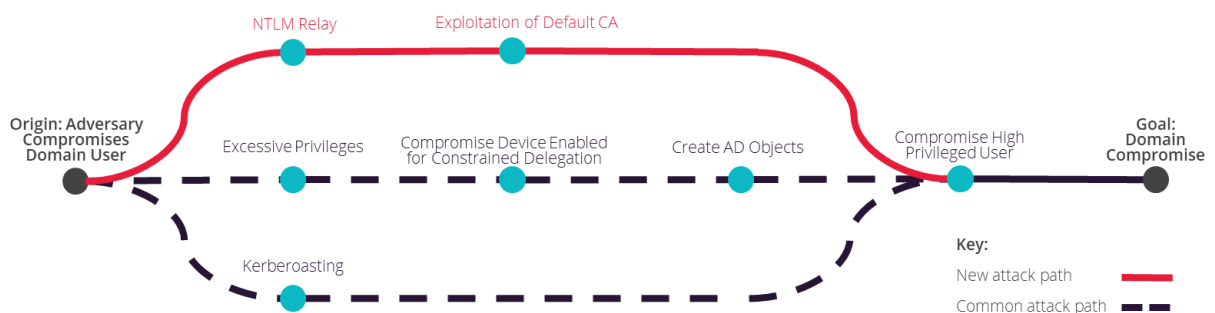


Figure 4. Example paths an adversary can take to a full domain compromise

Whilst Kerberos-based attacks are the more prevalent type of attack encountered in this context, these typically require a number of pre-requisite criteria to be met. They can be effectively countered through the best practice configuration of accounts to restrict service and user access to the minimum level of access to carry out their task, and the application of strong encryption types.

In contrast, the new technique presents a trivial method for an attacker to compromise previously hardened environments that are thought to be secure, **posing a particular risk to organisations who have previously invested in hardening their Active Directory environment with additional security controls**. Adversaries leveraging other misconfigurations or vulnerabilities who can gain access to the internal network can trivially achieve domain compromise against Active Directory environments with the Certificate Service enabled.

3.1 What is the impact of a successful attack?

The impact of domain compromise via abuse of Active Directory Certificate Service is extremely high; posing a colossal challenge to respond to and recover from due to the level of control that an adversary can achieve over all the devices and systems administered under the domain. Advantages for the attacker include:

- **Resilient domain control** - stealing certificates offers an adversary persistence beyond password reset or Kerberos ticket changes. **This is a total compromise from which it is impossible to recover without a complete rebuild of the Active Directory environment.**
- **Indefinite persistence** - an adversary who steals the Certificate Authority's certificate gains a unique capability: they will have a valid persistence method for as long as the certificate is valid. For administrative purposes, certificates are usually signed to have expiration dates long into the future. **This means that, for example, a certificate that doesn't expire until 2031 will allow an adversary certificate-based persistence to your Active Directory until 2031.**

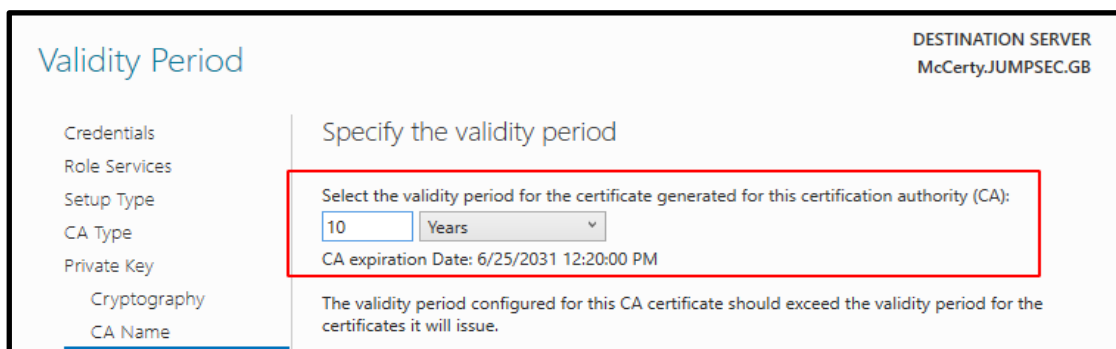


Figure 5. JUMPSEC's lab was configured with a decade-long certificate.

- **Trivial lateral movement and privilege escalation** - an attacker can achieve full domain compromise in just a few steps, with greater reliability and fewer dependencies, compared to pre-existing techniques.
- **Total Forest compromise** - the attack can quickly and easily scale to complete compromise of Active Directory, across all domains that are not segregated. In some instances, an adversary does not need to do anything 'extra' to achieve inter-domain compromise.

Previous Microsoft documentation guided administrators to set up a centralised certificate distributor. This architectural guidance once served to aid administrators, as maintaining a 'singular' public-key infrastructure across an organisation was easier. However, this certificate architecture now aids adversaries, not administrators. Centralised Certificate Authority machine(s) may have been established as intentional bridges across discrete domains. A compromised certificate and private key could give the adversary total control over all domains in the forest.

4. Identifying exposure

If Active Directory Certificate Service is in your environment, it is important to precisely investigate how these new techniques pose a threat to your domain.

As of July 2021, there are two toolkits that you can use to evaluate if an environment is vulnerable to certificate abuse.

4.1 Baseline Certificate Authority configuration overview

4.1.1 Do I have a Certificate Authority?

If you have Certificate Services enabled in your Active Directory, you likely have a Certificate Authority server. This server is key to Active Directory Certificate Service orchestration.

4.1.2 Gather Certificate Authority information

You can collect useful information through `certutil.exe` on **any domain joined endpoint**, you do not have to run this on the Certificate Authority itself.

```
CMD

certutil.exe

PS C:\Users\Administrator> & certutil.exe
Entry 0: (Local)
  Name: JUMPSEC-MCCERTY-CA'
  Organizational Unit:
  Organization:
  Locality:
  State:
  Country/region:
  Config: McCerty.JUMPSEC.GB\JUMPSEC-MCCERTY-CA'
  Exchange Certificate:
  Signature Certificate: McCerty.JUMPSEC.GB_JUMPSEC-MCCERTY-CA.crt'
  Description:
  Server: McCerty.JUMPSEC.GB'
  Authority: JUMPSEC-MCCERTY-CA'
  Sanitized Name: JUMPSEC-MCCERTY-CA'
  Short Name: JUMPSEC-MCCERTY-CA'
  Sanitized Short Name: JUMPSEC-MCCERTY-CA'
  Flags: 13'
  Web Enrollment Servers:
1
8
0
https://mccerty.jumpsec.gb/JUMPSEC-MCCERTY-CA_CES_Certificate/service.svc/CES0
CertUtil: -dump command completed successfully.
PS C:\Users\Administrator>
```

Figure 6. Basic Certificate Authority information

CMD

#Display verbose CA info. Takes a minute

Certutil.exe -tcainfo

```
PS C:\Users\Administrator> certutil -tcainfo
=====
CA Name: JUMPSEC-MCCERTY-CA
Machine Name: McCerty.JUMPSEC.GB
DS Location: CN=JUMPSEC-MCCERTY-CA,CN=Enrollment Services,CN=Public Key Se
Cert DN: CN=JUMPSEC-MCCERTY-CA, DC=JUMPSEC, DC=GB
CA Registry Validity Period: 2 Years -- 6/25/2023 12:31 PM
NotAfter: 6/25/2031 12:20 PM
Connecting to McCerty.JUMPSEC.GB\JUMPSEC-MCCERTY-CA ...
Server "JUMPSEC-MCCERTY-CA" ICertRequest2 interface is alive (16ms)

Enterprise Root CA

dwFlags = CA_VERIFY_FLAGS_NT_AUTH (0x10)
dwFlags = CA_VERIFY_FLAGS_CONSOLE_TRACE (0x20000000)
dwFlags = CA_VERIFY_FLAGS_DUMP_CHAIN (0x40000000)
ChainFlags = CERT_CHAIN_REVOCATION_CHECK_CHAIN_EXCLUDE_ROOT (0x40000000)
HCCE_LOCAL_MACHINE
CERT_CHAIN_POLICY_NT_AUTH
----- CERT_CHAIN_CONTEXT -----
ChainContext.dwInfoStatus = CERT_TRUST_HAS_PREFERRED_ISSUER (0x100)

SimpleChain.dwInfoStatus = CERT_TRUST_HAS_PREFERRED_ISSUER (0x100)

CertContext[0][0]: dwInfoStatus=10c dwErrorStatus=0
Issuer: CN=JUMPSEC-MCCERTY-CA, DC=JUMPSEC, DC=GB
NotBefore: 6/25/2021 12:10 PM
NotAfter: 6/25/2031 12:20 PM
Subject: CN=JUMPSEC-MCCERTY-CA, DC=JUMPSEC, DC=GB
Serial: 1bh2978f57f6adb54b3ed22d2b628883
```

Figure 7. Verbose Certificate Authority information

On a Certificate Authority machine itself, we can leverage the **registry** to get an **overview** of configuration information about the Certificate Service. There is additional, granular information that can be gathered here that does not appear in the above results.

POWERSHELL

```
# To be run on CA machine
# Get CA name
(Get-ItemProperty -Path
"HKLM:\SYSTEM\CurrentControlSet\Services\CertSvc\Configuration").Active
# Pass it as variable and then get more info
$CAName = (Get-ItemProperty -Path
"HKLM:\SYSTEM\CurrentControlSet\Services\CertSvc\Configuration").Active;
Get-ItemProperty -Path
"HKLM:\SYSTEM\CurrentControlSet\Services\CertSvc\Configuration\$CAName"
```

```
PS C:\Users\Administrator> Get-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Services\CertS
DSConfigDN : CN=Configuration,DC=JUMPSEC,DC=GB
DSDomainDN : DC=JUMPSEC,DC=GB
ViewAgeMinutes : 16
ViewIdleMinutes : 8
CAType : 0
UseDS : 1
ForceTeletex : 18
SignedAttributes : {RequesterName}
EKUOIDsForPublishExpiredCertInCRL : {1.3.6.1.5.5.7.3.3, 1.3.6.1.4.1.311.61.1.1}
CommonName : JUMPSEC-MCCERTY-CA
Enabled : 1
PolicyFlags : 0
CertEnrollCompatible : 0
CRLFlags : 256
CRLFlags : 2
InterfaceFlags : 1601
EnforceX500NameLengths : 1
SubjectTemplate : {Email, CommonName, OrganizationalUnit, Organization...}
ClockSkewMinutes : 10
LogLevel : 3
HighSerial : 104
CAServerName : McCerty.JUMPSEC.GB
ValidityPeriod : Years
ValidityPeriodUnits : 2
CAXchgCertHash : {}
KRACertHash : {}
KRACertCount : 0
KRAFlags : 0
CRLPublicationURLs : {65:C:\windows\system32\CertSrv\CertEnroll\%3%8%9.cr1, 79:ld
Services,CN=Services,%6%10, 0:http://%1/CertEnroll/%3%8%9.cr
CRLPeriod : Weeks
CRLPeriodUnits : 1
CRLOverlapPeriod : Hours
CRLOverlapUnits : 0
```

Figure 8. Granular Certificate Authority configuration detail


4.2 Exposure assessment: “Am I vulnerable?”

Organisations can initiate a cursory audit of susceptibility to certificate abuse in their environment. This script offers the answer: **'no / yes and how bad?'**. This script can be run from **any endpoint that is connected to the domain**. IJUMPSEC advise to run the script as a high privilege user on an endpoint.

POWERSHELL

```
#To be run from domain joined machine
#Confirm you are happy with the domain name, and then pass as variable
(Get-WmiObject -Class win32_computersystem).domain
$DomainName = (Get-WmiObject -Class win32_computersystem).domain
#Transfer the PS1 and check it
Invoke-WebRequest -Uri "https://raw.githubusercontent.com/RemiEscourrou/Invoke-Leghorn/main/Invoke-Leghorn.ps1" -OutFile ".\Invoke-Leghorn.ps1"
Read-Host -Prompt "Script is from a third party source. Please manually review before execution. Press any key to exit this warning"
#Import it
Import-Module .\Invoke-Leghorn.ps1
#Begin transcript mode. This will allow to save results, as Out-File doesn't work with this script
$ErrorActionPreference="SilentlyContinue"
Stop-Transcript | out-null
$ErrorActionPreference = "Continue"
Start-Transcript -path .\certificate_service_audit.txt -append
#Deploy it, using domain name. Run once normally, and then run again verbose.
Invoke-Leghorn -Domain $DomainName
#wait till above finished
Invoke-Leghorn -Domain $DomainName -verbose
#End transcript mode
Stop-Transcript
#Collect the certificate_service_audit.txt of results, ensure it has collected both results.
```

```
PS C:\Users\Administrator\Desktop> (Get-WmiObject -Class win32_computersystem).domainName | Out-File JUMPSEC.GB
PS C:\Users\Administrator\Desktop> $DomainName = (Get-WmiObject -Class win32_computersystem).domainName
PS C:\Users\Administrator\Desktop> Import-Module .\Invoke-Leghorn.ps1
PS C:\Users\Administrator\Desktop> Invoke-Leghorn -Domain $DomainName
>>
```



ASH

Invoke-Leghorn
PKI Analysis
@RemiEscourrou

```
[info] Request certificate templates on JUMPSEC.GB
[info] Found 33 certificate templates
[info] Search for a vulnerable template
[info] Search for a modifiable template (Admin ACEs are removed)
[info] Admin ACEs removed are *-512 *-519 *-516 *-500 *-498 S-1-5-9 and unresolved
[info] Request enrollment service on JUMPSEC.GB
[info] Found 1 enrollment services
[info] Analyzing JUMPSEC-MCCERTY-CA
[info] Admin ACEs removed are *-512 *-519 *-516 *-500 *-498 S-1-5-9 PKI servers and
[info] JUMPSEC-MCCERTY-CA can be requested (Enrollment) by NT AUTHORITY\Authenticated Users
PS C:\Users\Administrator\Desktop> Invoke-Leghorn -Domain $DomainName -verbose
```

Figure 9. Output that details Certificate Authority names and possible vulnerability status

Results will offer vulnerable certificate template names and the short reason it is considered vulnerable.

For example, in the extract of results below we can see that **SuperInsecureTemplate** has been named as a vulnerable template as it can be modified by non-admin users across the domain.

Moreover, the Certificate Authority server named **JUMPSEC-McCerty-CA** is too permissive in the breadth of users it will allow to make requests.

```
[info] Request certificate templates on JUMPSEC.GB
[info] Found 34 certificate templates
[info] Search for a vulnerable template
[info] Search for a modifiable template (Admin ACEs are removed)
[info] Admin ACEs removed are *-512 *-519 *-516 *-500 *-498 S-1-5-9 and unresolved SID
[CT Modifiable] SuperInsecureTemplate can be modified (ACEs) by non Admin User (Generic SID)
[info] Request enrollment service on JUMPSEC.GB
[info] Found 1 enrollment services
[info] Analyzing JUMPSEC-MCCERTY-CA
[info] Admin ACEs removed are *-512 *-519 *-516 *-500 *-498 S-1-5-9 PKI servers and unresolved SID
[info] JUMPSEC-MCCERTY-CA can be requested (Enrollment) by NT AUTHORITY\Authenticated Users
[CT Modifiable | ES OK] SuperInsecureTemplate is modifiable and is published on JUMPSEC-MCCERTY-CA
[06/28/2021 08:14:33] PS >
```

Figure 10. Results detail where the Certificate Service infrastructure is vulnerable

4.3 Susceptibility audit: "Where am I vulnerable?"

This script offers far more granular information than the above. It leverages the Microsoft official PKI PowerShell modules, with amendments by SpecterOps to audit. JUMPSEC have further amended the usage for easier deployment.

This should be run as a highly privileged user that has full permissions in Active Directory Certificate Service - running as **Enterprise Admin** is an acceptable shortcut method. It is best to run this on a Certificate Authority server.

POWERSHELL

```
# Run from CA machine as a high integrity process
# Install necessary features
Get-WindowsCapability -Online -Name "Rsat.*" | where Name -match
"CertificateServices\ActiveDirectory" | Add-WindowsCapability -Online
# Get the zip for PSPKIAudit
# https://github.com/GhostPack/PSPKIAudit/archive/refs/heads/main.zip
Read-Host -Prompt "Script is from a third party source. Please manually review before
execution. Press any key to exit this warning"
# Unzip
Expand-Archive .\PSPKIAudit-main.zip
# Move to directory. Should be in same directory as .psml and .psdl
cd "PSPKIAudit-main\PSPKIAudit-main"
# Prepare the modules
Get-ChildItem -Recurse | Unblock-File -verbose
# Import it. If fails, make sure you're in the right directory
Import-Module .\PSPKIAudit.psml -verbose
# Gather variables
$CAName = (Get-ItemProperty -Path
"HKLM:\SYSTEM\CurrentControlSet\Services\CertSvc\Configuration").Active;
$CAHost = (Get-ItemProperty -Path
"HKLM:\SYSTEM\CurrentControlSet\Services\CertSvc\Configuration\$CAName").CAServerName;
Write-host "$CAHost\$CAName"
# Deploy PSPKIAudit. This option will audit misconfigs and report back vulns
Invoke-PKIAudit -CAComputerName $CAHost -CAName $CAName
#optional final flag -verbose

#To better understand results, see Readme page - https://github.com/GhostPack/PSPKIAudit
```



```
[06/26/2021 07:49:49] PS >Invoke-PKIAudit -CAComputerName $CAHost -CAName $CAName

PSPKIAudit
v0.3.5

[*] Enumerating certificate authorities with Get-AuditCertificateAuthority...

=== Certificate Authority ===

ComputerName      : McCerty.JUMPSEC.GB
CAName            : JUMPSEC-MCCERTY-CA
ConfigString      : McCerty.JUMPSEC.GB\JUMPSEC-MCCERTY-CA
IsRoot            : True
AllowsUsersSuppliedSans : False
VulnerableACL     : False
EnrollmentPrincipals : NT AUTHORITY\Authenticated Users
EnrollmentEndpoints : http://McCerty.JUMPSEC.GB/certsrv/
NTLMEEnrollmentEndpoints : http://McCerty.JUMPSEC.GB/certsrv/
DACL              : NT AUTHORITY\Authenticated Users (Allow) - Enroll
                  BUILTIN\Administrators (Allow) - ManageCA,
                  ManageCertificates
                  JUMPSEC\Domain Admins (Allow) - ManageCA,
                  ManageCertificates
                  JUMPSEC\Enterprise Admins (Allow) - ManageCA,
                  ManageCertificates
Misconfigurations : ESC8

[!] The above CA is misconfigured!
[!] Potentially vulnerable Certificate Templates:

CA : McCerty.JUMPSEC.GB\JUMPSEC-MCCERTY-CA
Name : SuperInsecureTemplate
SchemaVersion : 2
OID : 1.3.6.1.4.1.311.21.8.1007155.10732202.4672896.3299
    683.10492925.156.16064494.5139542
vulnerableTemplateACL : True
```

Figure 11. First half of results

```
[!] Potentially vulnerable Certificate Templates:

CA : McCerty.JUMPSEC.GB\JUMPSEC-MCCERTY-CA
Name : SuperInsecureTemplate
SchemaVersion : 2
OID : 1.3.6.1.4.1.311.21.8.1007155.10732202.4672896.3299
    683.10492925.156.16064494.5139542
VulnerableTemplateACL : True
LowPrivCanEnroll : False
EnrolleeSuppliesSubject : False
EnhancedKeyUsage : Server Authentication (1.3.6.1.5.5.7.3.1)|Client
                  Authentication (1.3.6.1.5.5.7.3.2)
HasAuthenticationEku : True
HasDangerousEku : False
EnrollmentAgentTemplate : False
CAManagerApproval : False
IssuanceRequirements : [Issuance Requirements]
                      Authorized signature count: 0
                      Reenrollment requires: same criteria as for
                      enrollment.
ValidityPeriod : 1 years
RenewalPeriod : 6 weeks
Owner : JUMPSEC\Cert_Enterprise
DACL : NT AUTHORITY\ENTERPRISE DOMAIN CONTROLLERS
      (Allow) - Enroll
      NT AUTHORITY\Authenticated Users (Allow) - Read,
      Write, FullControl
      JUMPSEC\Enterprise Read-only Domain Controllers
      (Allow) - Enroll
      JUMPSEC\Domain Admins (Allow) - Read, Write,
      Enroll
      JUMPSEC\Domain Controllers (Allow) - Enroll
      JUMPSEC\Enterprise Admins (Allow) - Read, Write,
      Enroll
      JUMPSEC\Cert_Enterprise (Allow) - Read, Write
Misconfigurations : ESC4
```

Figure 12. Second half of results, including how and where misconfigurations have caused insecurity

5. How can organisations protect, detect, and respond to the techniques?

The recommended scripts will have signposted specific machines and Active Directory Certificate Service configurations that are vulnerable and require hardening.

JUMPSEC has summarised and refined SpecterOps' guidance on how to proactively defend against the imminent threat that Active Directory Certificate Service exposures will present. The time required to implement these controls will vary. To assist remediation activity, JUMPSEC has prioritised activities according to their contributions to hardening the domain and building broader cyber resilience.

Priority	Description	Actions
P1	Enable Certificate Authority logging	
P2	Implement defence-in-depth secure architecture	<ul style="list-style-type: none"> ➤ Move to tiered system, with a Root and Subordinate Certificate Authority ➤ Evaluate previous Microsoft guidance on centralising a Certificate Authority infrastructure to serve multiple domains/forests ➤ Certificate Authority machines must now be treated as Tier 0 assets (like a Domain Controller) ➤ Disaster recovery plans should be revised to consider Certificate Authority
P2	Securely configure Certificate Authority settings	<p>Essential</p> <ul style="list-style-type: none"> ➤ Disabling Subject Alternative Name (SANs) ➤ Restricting Enrolment Agents ➤ Evaluate permissions on Certificate Authority machines ➤ Review certificate templates and settings ➤ Control Active Directory Certificate Service HTTP endpoints. Turn off, where possible <p>Enhanced protection</p> <ul style="list-style-type: none"> ➤ Control access to templates with over permissive Enhanced Key Usage ➤ Control certificate-based authentication ➤ Isolate and protect certificate's private keys
P3	Identify and monitor relevant Event IDs	
P3	Plan for a potential Active Directory Certificate Service compromise	

5.1 Enable Certificate Authority logging

This is the number one priority, to action immediately.

For various reasons, Active Directory Certificate Service does not enable logs by default, which will prevent your SOC, SIEM, or other monitoring solution from detecting malicious activity. To resolve this, **Certificate Authorities'** must have their logs enabled.

To enable logging, you can do this via `certutil.exe`.

```
CMD

#The 127 value can enable CA logging.
certutil.exe -setreg CA\AuditFilter 127

C:\Users\Administrator\Desktop>certutil.exe -setreg CA\AuditFilter 127
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\CertSvc\Configuration\
JUMPSEC-MCCERTY-CA\AuditFilter:

Old Value:
    AuditFilter REG_DWORD = 0

New Value:
    AuditFilter REG_DWORD = 7f (127)
CertUtil: -setreg command completed successfully.
The CertSvc service may need to be restarted for changes to take effect.
```

Figure 13. Cmdline returns AuditFilter as active and logging everything

If you would prefer to use a GUI, utilise `certsrv.msc` in cmd or search in Start. Right-click on the Certificate Authority, properties and then click under the **Auditing** tab and **enable every single event to audit**.

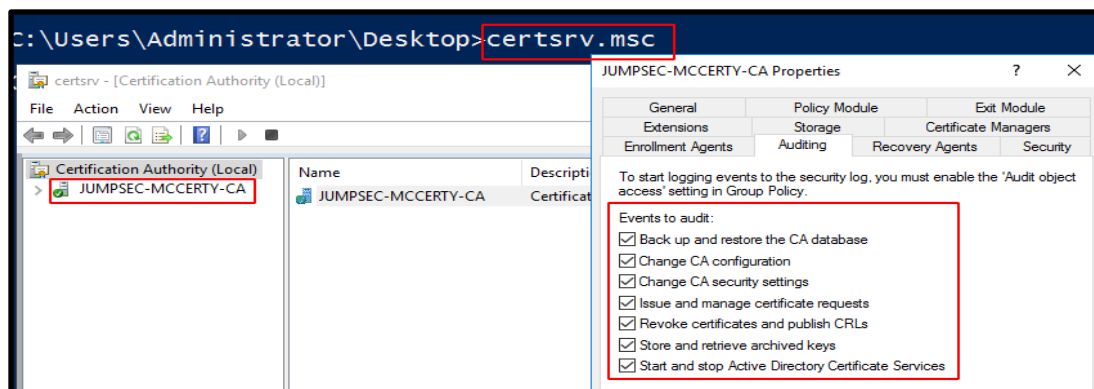


Figure 14. GUI alternative to above cmdline

5.2 Implementing defence-in-depth secure architecture

5.2.1 Implementing tiered architecture for Certificate Authorities

Microsoft's security guidance for Active Directory Certificate Service is to use a tiered architecture of Certificate Authorities. By utilising a **subordinate** Certificate Authority, an endpoint will not receive a certificate directly from the **root** Certificate Authority. Microsoft guidance suggests that root Certificate Authorities are air gapped, and never domain-joined. This is to ensure that an adversary cannot compromise the root certificates or private keys.

5.2.2 Subordinate Certificate Authority defences

By segregating the root Certificate Authority server and tiering an inferior subordinate Authority the risk is significantly reduced. However an adversary can still take advantage of the subordinate Authority. Therefore we must harden the defences surrounding a **Subordinate Certificate Authority**. One method is to set up **Certificate Authority Constraints**. The issuing rules can restrict many things, the most important being restricting certificates issued with **Enhanced Key Usage**.

5.2.3 Cross-forest management

Previous guidance advised having a single certificate infrastructure that could serve multiple domains. The latest guidance advises that administrators set up a centralised Active Directory Certificate Service location (a resource forest). This centralised location will provide enrolment services for the domains (account forests). For cross-forest enrolment, administrators can publish the Root Certificate Authority from the centralised resource forest to the account forests and add the enterprise Certificate Authority certificates from the resource forest to specific objects in the account forest (the specific objects being NTAuthCertificates and AIA containers).

5.2.4 Certificate Authority machines are now a Tier 0 Asset

Certificate Authority servers must now be given the same treatment as a Domain Controller. Administrators likely already appreciate the sensitivity that an Authority system requires, but it is worth explicitly categorising the level of risk management that must be applied, often referred to as 'Tier 0'.

As defined by NCSC, Tier 0 encompasses "the root of trust that other administration relies upon". If a Tier 0 asset is compromised, an adversary will be able to gain access to the interconnected components that other tiers are built on. The NCSC explicitly names "infrastructure used to generate cryptographic material which other components rely on" as an example of a Tier 0 system related to Active Directory Certificate Service.

A Tier 0 Certificate Authority server must have restricted access for authentication, and ideally it should be protected with trust-based obstacles - for example, Privileged Access Management or (if in Azure Active Directory) Privilege Identity Management. Disaster recovery plans must also consider how a Tier 0 system like a Certificate Authority can be managed in an emergency. Contingency plans should be revised, given that compromise of a Certificate Authority instigates a 'cascading failure' of security across an entire organisation. **Both Subordinate and Root Certificate Authorities must be given Tier 0 treatment.**

5.3 Securely configure Certificate Authority settings

There are 'quick-wins' in the defensive strategy for Active Directory Certificate Service. Various settings must be amended on all of the various Certificate Authorities (Root, Subordinate, Enterprise)

5.3.1 Turn off Subjective Alternate Name (SAN)

By default, a Certificate Authority ignores SAN entries in a certificate request. If your environment has enabled SAN extensions it is advised where possible to turn off this optional feature.

To do this, the below PowerShell must be run on **every** Certificate Authority.

First, you can collect the **status** of the SAN via PowerShell registry. These may claim they are disabled here, but it's worth going through the process of disabling SAN nonetheless

```
POWERSHELL

#Check SAN status via pwsh

$CAName = (Get-ItemProperty -Path
"HKLM:\SYSTEM\CurrentControlSet\Services\CertSvc\Configuration").Active;

Get-ItemProperty -Path
"HKLM:\SYSTEM\CurrentControlSet\Services\CertSvc\Configuration\$CAName\PolicyModules\Certif
icateAuthority_MicrosoftDefault.Policy" | select-object SubjectAltName* | Format-list

SubjectAltName : DISABLED: Set to Email to set SubjectAltName extension
                to the email address
SubjectAltName2 : DISABLED: Set to Email to set SubjectAltName2 extension
                 to the email address
```

Figure 15. SAN config status

Second, you can disable the SAN extension via `certutil.exe`

```
POWERSHELL

# Collect CA Hostname and CA Name, and save as variables.
$CAName = (Get-ItemProperty -Path
"HKLM:\SYSTEM\CurrentControlSet\Services\CertSvc\Configuration").Active;
$CAHost = (Get-ItemProperty -Path
"HKLM:\SYSTEM\CurrentControlSet\Services\CertSvc\Configuration\$CAName").CAServerName;
Write-host "$CAHost\$CAName"

# Then,leverage certutil.exe. You can run the below in pwsh to maintain the set variable
```

```
& certutil.exe -config "$CAHost\$CAName" -getreg "policy\EditFlags"

# the flag we are looking is -EDITF_ATTRIBUTESUBJECTALTNAME2
# disable the flag if above returns as present

& certutil.exe -config "$CAHost\$CAName" -setreg policy\EditFlags -
EDITF_ATTRIBUTESUBJECTALTNAME2

PS C:\Users\Administrator> Write-Host "$CAHost\$CAName"
McCerty.JUMPSEC.GB\JUMPSEC-MCCERTY-CA
PS C:\Users\Administrator> & certutil.exe -config "$CAHost\$CA
"policy\EditFlags"
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\CertSvc\C
JUMPSEC-MCCERTY-CA\PolicyModules\CertificateAuthority_Microsof
cy\EditFlags:

EditFlags REG_DWORD = 15014e (1376590)
EDITF_REQUESTEXTENSIONLIST -- 2
EDITF_DISABLEEXTENSIONLIST -- 4
EDITF_ADDOLDKEYUSAGE -- 8
EDITF_BASICCONSTRAINTSCRITICAL -- 40 (64)
EDITF_ENABLEAKIKEYID -- 100 (256)
EDITF_ENABLEDEFAULTSMIME -- 10000 (65536)
EDITF_ATTRIBUTESUBJECTALTNAME2 -- 40000 (262144)
EDITF_ENABLECHASECLIENTDC -- 100000 (1048576)
CertUtil: -getreg command completed successfully.
PS C:\Users\Administrator>
```

Figure 16. EDITF_ATTRIBUTESUBJECTALTNAME2 is the dangerous config setting that we are to disable

SAN can be disabled in other places too. This involves some registry manipulation on every Domain Controller in the environment.

```
POWERSHELL

#Check if UseSubjectAltName exists. May not show up here
Get-ChildItem -recurse -path "HKLM:\SYSTEM\CurrentControlSet\Services\Kdc"

Get-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Services\Kdc" -Name
"UseSubjectAltName"

#disable SAN
Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Services\Kdc" -Name
"UseSubjectAltName" -Value '0' -verbose

#same for Schannel, reg may be empty as it is not switched on by default
# We are looking for CertificateMappingMethods
Get-ChildItem -path "HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\"

#if CertificateMappingMethods is present, change it's value
Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\"
-Name "CertificateMappingMethods" -Value '1' -verbose # or -value 2
```

```
PS C:\Users\Administrator> Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Services\Kdc" -Name "UseSubjectAltName" -Value 0 -verbose
VERBOSE: Performing the operation "Set Property" on target "Item: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Kdc Property: UseSubjectAltName".
PS C:\Users\Administrator> get-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Services\Kdc" -Name "UseSubjectAltName"

UseSubjectAltName : 0
PSPath             : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Kdc
PSParentPath       : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services
PSChildName        : Kdc
PSDrive            : HKLM
PSProvider         : Microsoft.PowerShell.Core\Registry
```

Figure 17. Registry manipulation to neutralise SAN threat to Domain Controllers

5.3.2 Securing SAN if you choose to keep it

JUMPSEC has discussed why a SAN extension is a liability in an Active Directory Certificate Service environment, and current evidence suggests disabling the extension is appropriate. However, this may not be suitable for some organisations that need SAN extensions.

To configure SAN to reduce the risk, run `certtmpl.msc`, go to the **certificate template**, and right-click for **properties** on certificates that allow domain authentication.

Under **Issuance Requirements**, enable **manager approvals**.

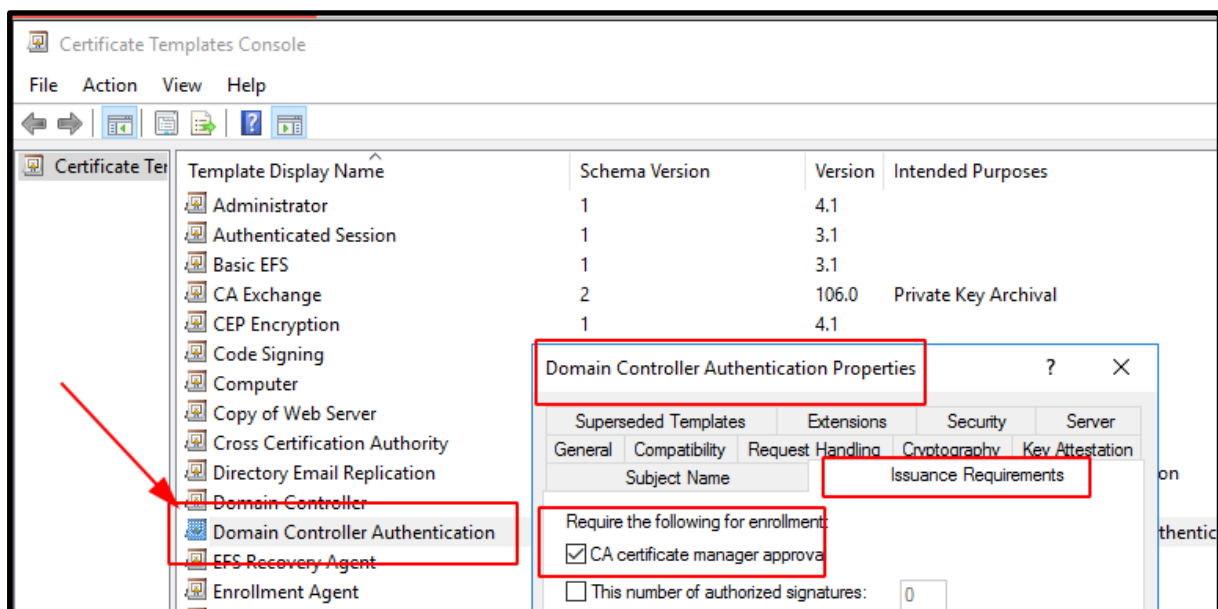


Figure 18. GUI method to restrict certificate template enrolment

5.3.3 Restrict enrolment agents

Some administrators may have leveraged enrolment agents. It is important to restrict which principles can act as enrolment agents, and the certificate templates and users that those agents can enrol.

You can get to this setting via `certsrv.msc` by right-clicking on the **Certificate Authority** and navigating to **> properties, > Enrolment Agents**.

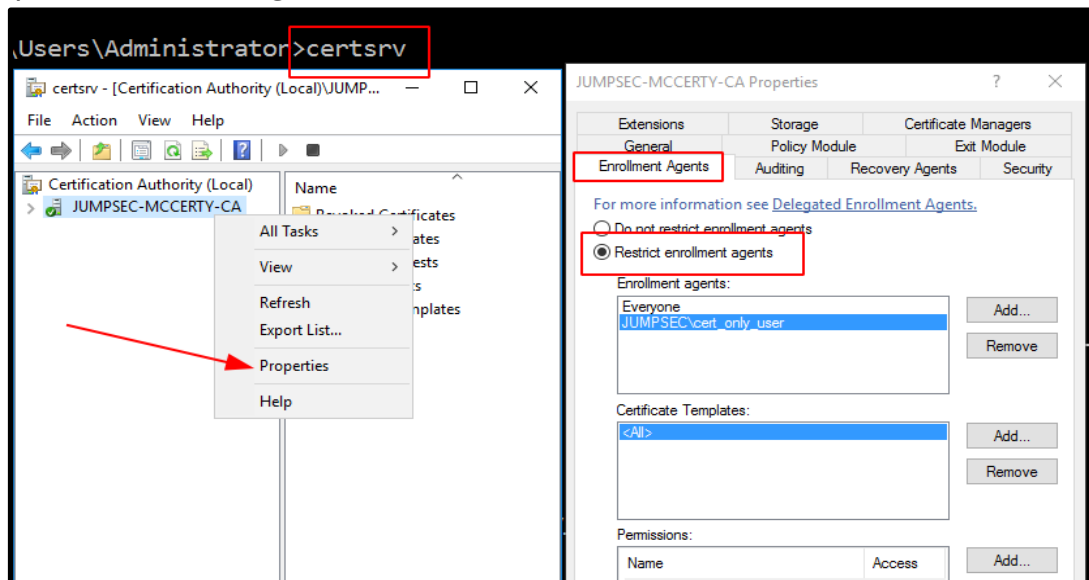


Figure 19. GUI method to further restrict enrolment permissions

5.3.4 Restrict Certificate Authority Permissions

It is important to audit the permissions, privileges, and various forms of access existing on a Certificate Authority machine. `certsrv.msc` can be leveraged again here. On the Certificate Authority right-click on **properties**, and go into the **security** tab. Here you will be able to **restrict permissions** to ensure that only specific administrative groups have full control.

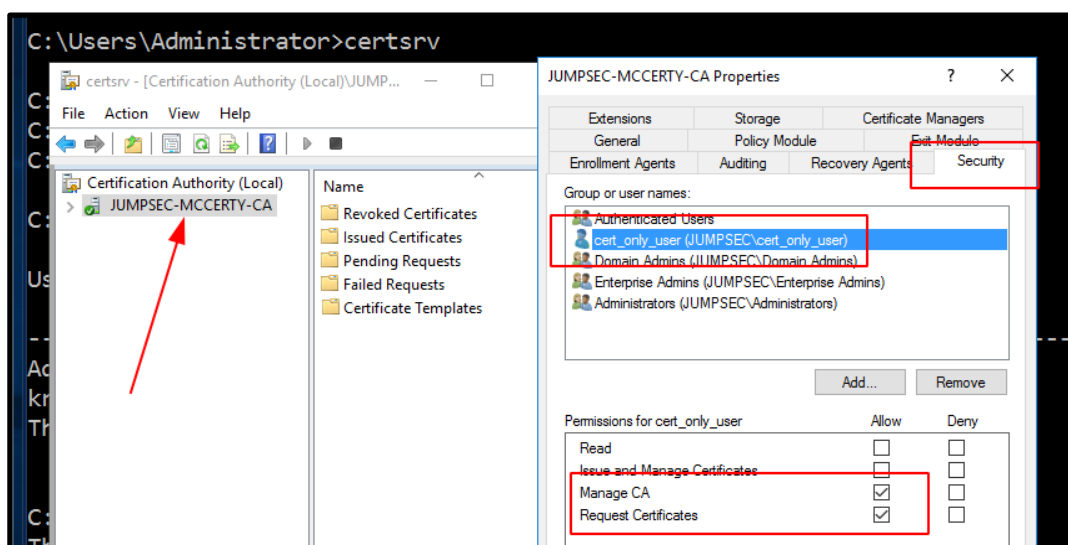


Figure 20. GUI method to harden Certificate Authority permissions and access

5.3.5 Review certificate templates and settings

Existing, published certificate templates should be reviewed and evaluated. If a template isn't in use, please decommission it on every Certificate Authority.

To show published templates, administrators can issue the following command.

CMD

```
#Run this command, and scroll down to show enrolled templates
certutil.exe -TCAInfo

#show template permissions. Very verbose
certutil.exe -v -dsTemplate
```

```
Supported Certificate Templates:
Cert Type[0]: DirectoryEmailReplication (Directory Email Replication)
Cert Type[1]: DomainControllerAuthentication (Domain Controller Authentication)
Cert Type[2]: KerberosAuthentication (Kerberos Authentication)
Cert Type[3]: EFSRecovery (EFS Recovery Agent)
Cert Type[4]: EFS (Basic EFS)
Cert Type[5]: DomainController (Domain Controller)
Cert Type[6]: WebServer (Web Server)
Cert Type[7]: Machine (Computer)
Cert Type[8]: User (User)
Cert Type[9]: SubCA (Subordinate Certification Authority)
Cert Type[10]: Administrator (Administrator)
Validated Cert Types: 11
```

Figure 21. Results of certificate templates that exist in your environment

5.3.6 Restricting over permissive EKUs

Enhanced Key Usage (EKUs) is an extension that dictates how a certificate's public key can be used. Specific OIDs have specific numbers that exist for different purposes (client authentication, server authentication, securing email, for example).

```
2.5.29.14: Flags = 0, Length = 16
Subject Key Identifier
7c 4e b0 7b ca b7 c1 66 a8 b5 c2 15 83 84 f2 7d a1 eb 43 ac

2.5.29.37: Flags = 0, Length = 1
Enhanced Key Usage
Client Authentication (1.3.6.1.5.5.7.3.2)

1.3.6.1.4.1.311.20.2: Flags = 0, Length = 16
Certificate Template Name
ClientAuth
```

Figure 22. An example of what an EKU looks like.

A template may have an EKU that is over permissive. To collect your EKU OID numbers you can run.

CMD

```
#show enrolled templates
certutil.exe -TCAInfo
```

It is easiest to remediate this via **Certification Authority Management Tools** GUI. Start `certsrv.msc`, click the dropdown under your Certificate Authority. Right-click under **Certificate Template**, and then **manage**. Under the **extensions** tab, a template may contain **All Purpose**, **Certificate Request Agent**, or **Null** (subordinate Certificate Authority).

It is important to restrict the enrolment of these certificates to **privileged groups only** which you can administer under the **security** tab (we have shown how to do this in previous sections).

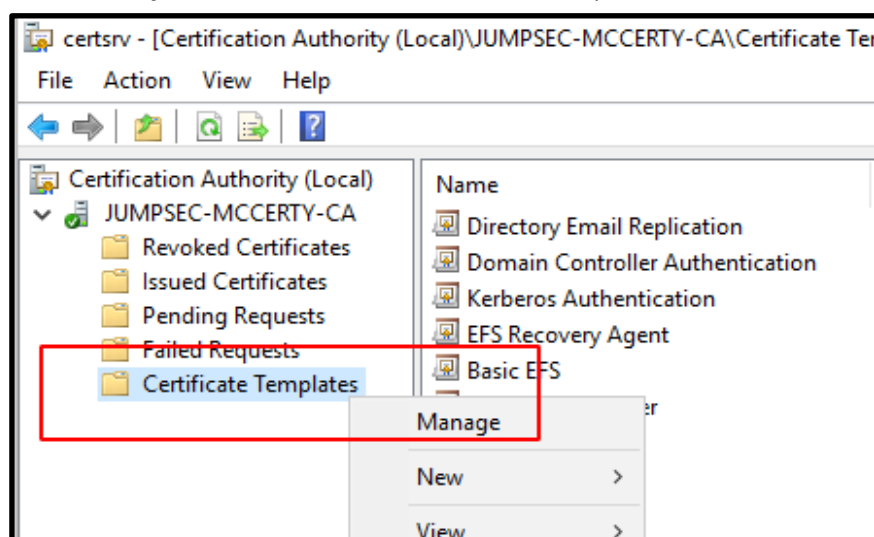


Figure 23. GUI method to manage certificate templates

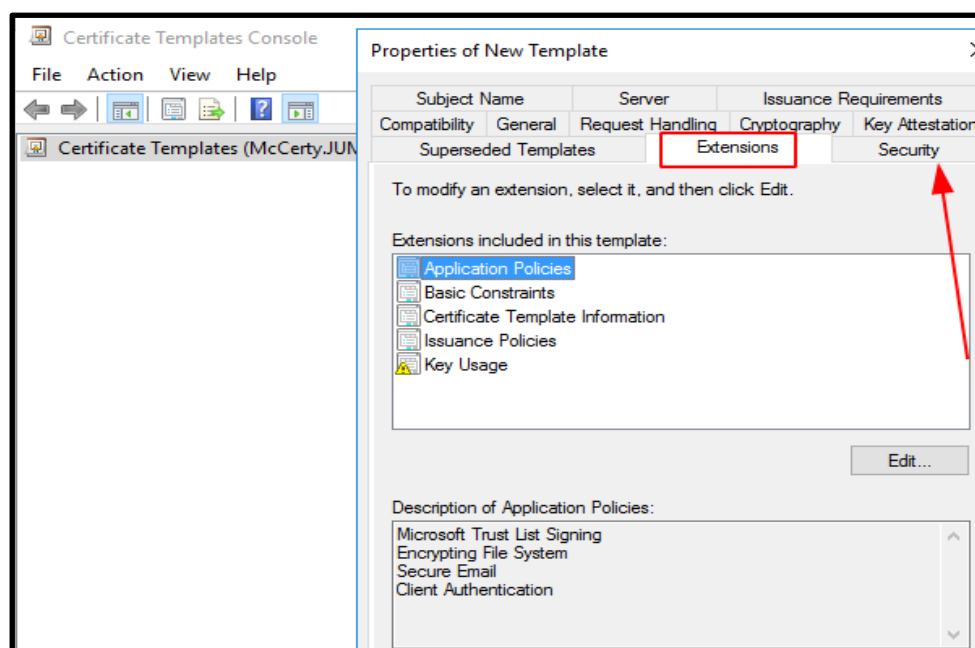


Figure 24. Restrict enrolment to particular groups where EKU's are over permissive.

5.3.7 Control Certificate authentication

Earlier, we discussed the relationship that Active Directory Certificate Service has with Kerberos and authentication. It is prudent to **remove all certificates** from the **NTAuthCertificate** object. This assumes however smart card authentication is not in use and the network does not require certificate authentication.

To **view** the certs in the NTAuthCert container, run the following from a domain-elevated prompt

```
CMD

#Check if you have any NTAuthCerts. It is entirely possible this will all be empty
certutil.exe -store -? | findstr "CN=NTAuth"

#or this more verbose option
certutil -ds -v NtAuthCertificates

PS C:\Users\Administrator> certutil.exe -store -? | findstr "CN=NTAuth"
"ldap:///CN=NTAuthCertificates,CN=Public Key Services,CN=Services,CN=Configurati
on,DC=JUMPSEC,DC=GB?cACertificate?base?objectClass=certificationAuthority" (Enterprise CA Ce
rtificates)
```

Figure 25. Returns NTAuthCertificate object in LDAP

```
PS C:\Users\Administrator> certutil -ds -v NtAuthCertificates
CN=Public Key Services,CN=Services,CN=Configuration,DC=JUMPSEC,DC=GB:
NTAuthCertificates
  objectClass
    Element 0: "top"
    Element 1: "certificationAuthority"
  cn = "NTAuthCertificates"
  cACertificate
===== Certificate 0 =====
Serial Number: 1bb2978f57f6adb54b3ed22d2b628883
Issuer: CN=JUMPSEC-MCCERTY-CA, DC=JUMPSEC, DC=GB
NotBefore: 6/25/2021 12:10 PM
NotAfter: 6/25/2031 12:20 PM
Subject: CN=JUMPSEC-MCCERTY-CA, DC=JUMPSEC, DC=GB
CA Version: V0.0
Signature matches Public Key
Root Certificate: Subject matches Issuer
Cert Hash(sha1): 59 3a c6 26 f4 7c 96 1e 2c 01 e2 10 1a 79 ec 07 6e 0f

  authorityRevocationList = EMPTY
  certificateRevocationList = EMPTY
  distinguishedName = "CN=NTAuthCertificates,CN=Public Key Services,
C=JUMPSEC,DC=GB"
  instanceType = "4"
  whenCreated = "20210625192043.0Z" 6/25/2021 12:20 PM
  whenChanged = "20210625192043.0Z" 6/25/2021 12:20 PM
  uSNCreated = "16408" 0x4018
  uSNChanged = "16409" 0x4019
  showInAdvancedViewOnly = "TRUE"
  ntSecurityDescriptor = "D:AI(A;OICI;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;[
RCWDWO;;;S-1-5-21-105986035-36484523-135057450-519)(A;OICI;CCDCLCSWRP
CRPLORC;;;WD)(A;CIID;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;S-1-5-21-105986035-
CCLCSWRPWPLOCRSDRCWDWO;;;DA)"

  Allow Full Control JUMPSEC\Domain Admins
  Allow Full Control JUMPSEC\Enterprise Admins
  Allow Full Control BUILTIN\Administrators
  Allow Read Everyone
  Allow Full Control JUMPSEC\Enterprise Admins
  Allow Full Control JUMPSEC\Domain Admins

  name = "NTAuthCertificates"
  objectGUID = 4e88eec9-3caf-43dd-a3ca-59e9718233a9
  objectCategory = "CN=Certification-Authority,CN=Schema,CN=Configu
dsCorePropagationData = "16010101000000.0Z" EMPTY
```

Figure 26. Verbose details of NTAuthCertificate's contents

To simplify the output, here is a powershell query that simply returns the cert names that exist in the NTAAuthCert store.

```
POWERSHELL

(Get-ItemProperty -path
"HKLM:\Software\Microsoft\EnterpriseCertificates\NTAuth\certificates\*").PsChildname

PS C:\Users\Administrator> (Get-ItemProperty -path
NTAuth\certificates\*").PsChildname
593AC626F47C961E2C01E2101A79EC076E01DBC2
```

Figure 27. Simplified results to just collect names of NTAAuthCertificate certificates.

Identified certificates can be deleted if you run the following commands from an Enterprise Admin prompt.

```
CMD

#delete certificates from NTAAuth store

certutil.exe -viewdelstore "ldap:///CN=NtAuthCertificates,CN=Public Key
Services,CN=Services,CN=Configuration,DC=,DC=?cACertificate?base?objectclass=certificationA
uthority"
```

Deleting the certificates in the NTAAuth container can also be achieved through the GUI.

Start PKIView.msc and ignore any errors. Right-click on **Enterprise PKI**, and select **Manage Active Directory Containers**.

Select a particular certificate to be **deleted** in this container.

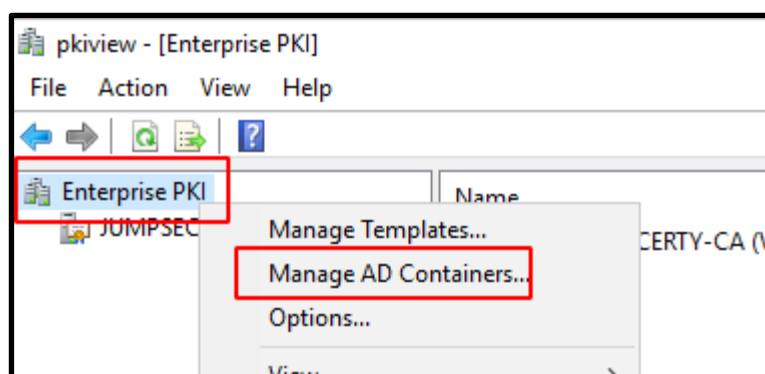


Figure 28. GUI method to manage NTAAuthCertificate object

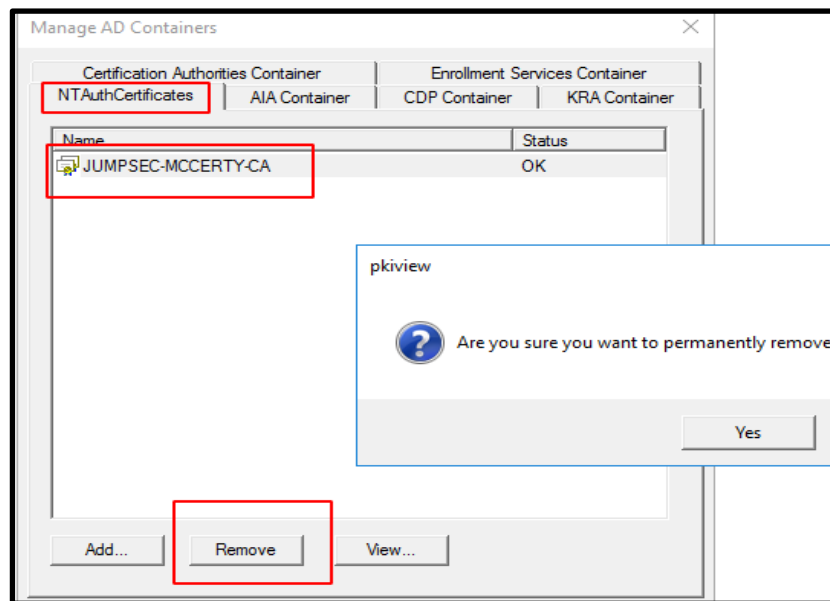


Figure 29. Removing the contents of the container

5.3.8 Isolate and protect Certificate Authority private keys

Protecting private keys is a crucial part of public-key infrastructure control. **Data Protection API (DPAPI)** is a control that encrypts the key on the endpoint, using the local computer account credentials. DPAPI can protect Certificate Authority private keys at a 'hardware' level.

It is possible for an adversary to abuse DPAPI's backup functionality to retrieve protected objects. It is advised that organisations enable **Credential Guard** and **Trusted Platform Module (TPM)**, the latter of which cryptographically affirms (to the inquiring Certificate Authority) that the private key is 'trusted'. TPM must be initialised at a **BIOS** level.

Credential Guard can then be issued at a registry level with PowerShell:

POWERSHELL

```
#check your specific use cases. Our guidance may not be compatible with your organisation:
https://docs.microsoft.com/en-us/windows/security/identity-protection/credential-guard/credential-guard-manage
```

#Enable virtualisation portion of security

```
new-itemproperty -Path "HKLM:\System\CurrentControlSet\Control\DeviceGuard" -Name
"EnableVirtualizationBasedSecurity" -value '1' -PropertyType 'DWORD' -Force
new-itemproperty -Path "HKLM:\System\CurrentControlSet\Control\DeviceGuard" -Name
"RequirePlatformSecurityFeatures" -value '3' -PropertyType 'DWORD' -Force
```

#Enable Cred Guard portion

```
new-itemproperty -Path "HKLM:\System\CurrentControlSet\Control\LSA" -Name "LsaCfgFlags" -
value '1' -PropertyType 'DWORD' -Force
```

#Confirm Credential Guard is enabled

```
Get-ComputerInfo | select DeviceGuardSecurityServicesConfigured
```

```
PS C:\Users\Administrator> Get-ComputerInfo |  
>> select DeviceGuardSecurityServicesConfigured  
  
DeviceGuardSecurityServicesConfigured  
-----  
{CredentialGuard}
```

Figure 30. Hardware-level hardening confirmation

5.3.9 Control Active Directory Certificate Service HTTP

Certificate Authority systems can be interacted with through (internal) web pages, hosted by IIS. This web service allows common tasks like requesting, submitting, and retrieving certificates. Adversaries can leverage the web service of Active Directory Certificate Service and relay credentials (NTLM hashes) to achieve exploitation as part of the attack path.

Where possible, the web service of the Certificate Authority system should be turned off.

If the web service is mission critical and will not be disabled, there are steps to secure it.

One option is to enforce HTTPS access.

Another is to disable NTLM authentication in the certificate web service.

- Search and start **IIS Manager**, and pick your Certificate Authority server.
- Follow the dropdown from the Certificate Authority server, down to **sites** and then **default web site**.
- Follow the drop-down again to **CertSrv**, and click it
- On the right hand side, click into **authentication**
- Right-click on **Windows Authentication**, and select **Providers**
- Remove the default **NTLM** and **Negotiate Providers**.
- Add **Negotiate:Kerberos** as a provider

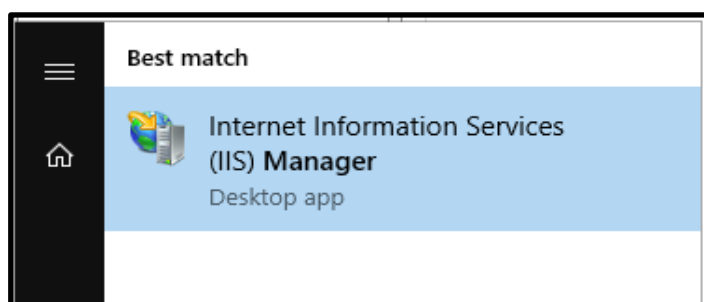


Figure 31. Finding IIS manager

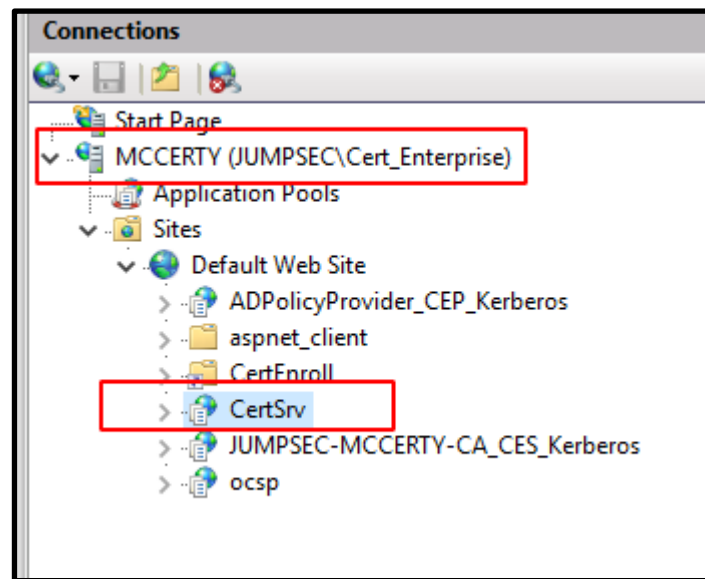


Figure 32. Choosing your Certificate Authority's CertSrv

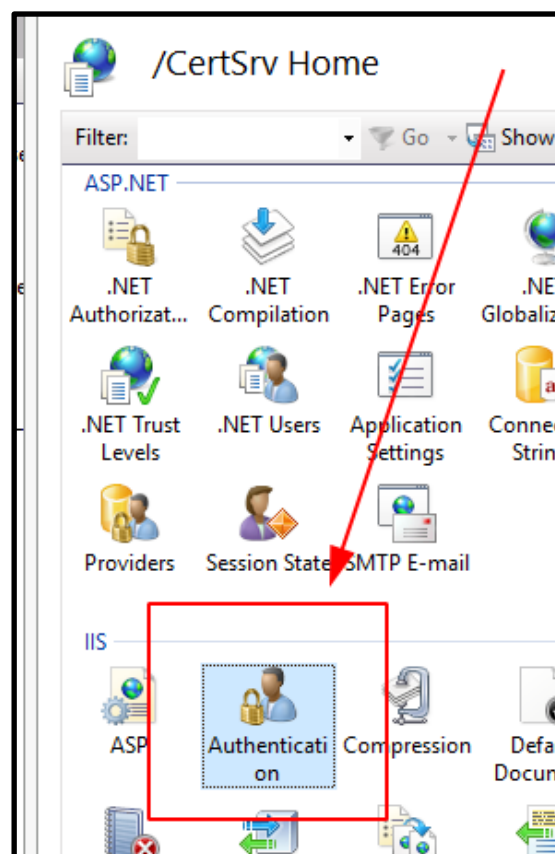


Figure 33. Click on the Authentication icon

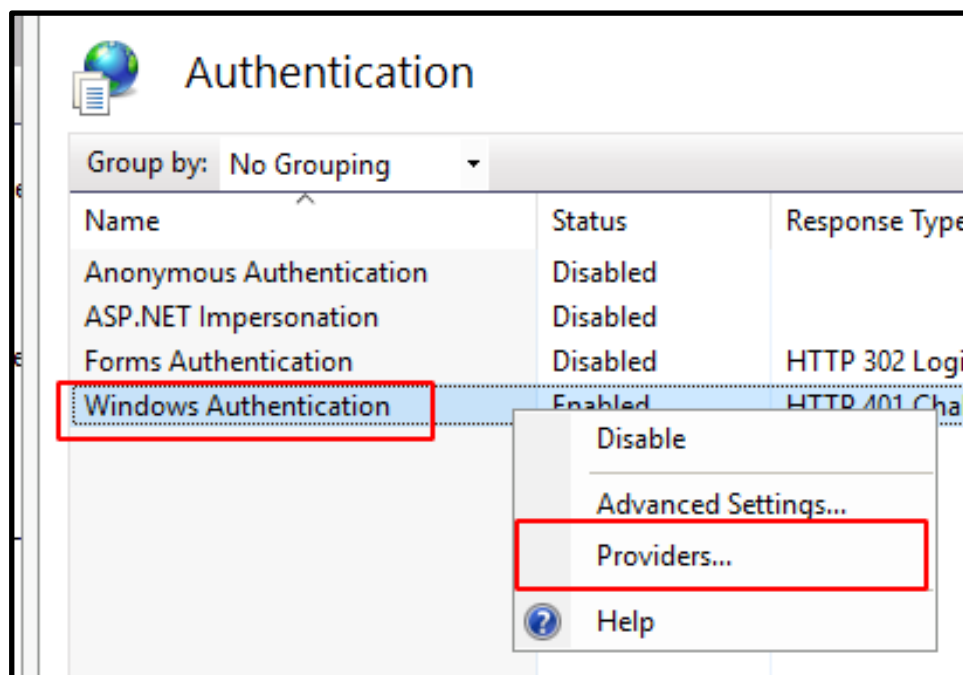


Figure 34. Under Windows Auth, click on Providers

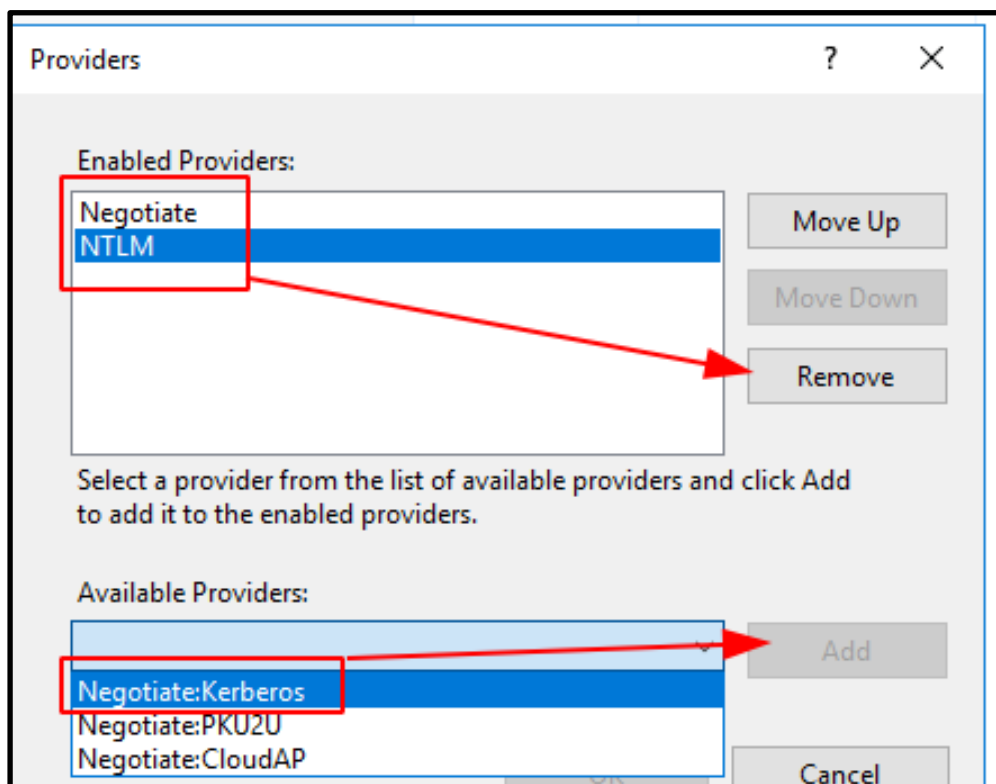


Figure 35. Harden authentication mechanisms

5.4 Identify and monitor relevant Event IDs

Once Active Directory Certificate Service is exploited in the wild, the security community will have better guidance for which events to monitor and how, to facilitate precise and reliable detection of malicious activity. For now, these event IDs will enable noise to be filtered out and normal activity to be baselined.

Complexity	Description	Actions
Low	Certificate Template – any activity should be treated as suspicious	<ul style="list-style-type: none"> ➤ 4989: Brand new certificate template loaded. ➤ 4899: Existing certificate template has been modified. This can catch an adversaries' exploitation attempts as templates will rarely be legitimately modified. This event can fail to log and so cannot be exclusively relied upon. ➤ 4900: Security permissions of a template have been changed AND template has been enrolled.
Low	Certificate Authority Events - any unscheduled activity should be investigated	<ul style="list-style-type: none"> ➤ 4882: Spawned when security permissions modified for the Certificate Service. ➤ 4890: Spawned when certificate manager settings modified for the Certificate Service. ➤ 4892: Spawned when any property of the Certificate Service is modified. ➤ 4876 & 4877: Spawned from Certificate Authority when backup has started and finished, respectively. ➤ 5601, 5061, & 5059: Child-events of above backup. Indicates interaction with key storage provider.
Low	Certificate Requests - requests generate a lot of noise, but baselining will allow abnormal to stand out.	<ul style="list-style-type: none"> ➤ 4886: Spawned from the Certificate Authority when it receives a certificate request. ➤ 4887: A certificate request was approved, and a certificate is issued by the Certificate Authority. ➤ 4889: A certificate request has not yet been approved. Decision pending on administrator. ➤ 4888: A certificate request has been denied. This may note adversaries' failed attempts.

High	<p>Certificate Authentications - to get beyond the noise, these Kerberos and Schannel must be filtered by the contents of the message fields.</p>	<ul style="list-style-type: none"> ➤ 4768: Spawned from a Domain Controller when a TGT is requested. If certificate-based authentication, the event's sub-fields will contain "certificate Information". ➤ 4769: For Schannel, spawned from a Domain Controller when service ticket requested. ➤ 4648: Explicit credentials log. Will contain user identity attached with certificate.
------	--	--

5.5 Plan for a potential compromise

Incident response and disaster recovery plans must assume that an adversary could weaponise all certificates across the Active Directory, and therefore would have unending, unlimited persistence to the network.

Should a Certificate Authority machine ever be compromised, the entire Active Directory Certificate Service infrastructure can no longer be trusted. **The entire infrastructure must be taken offline; every existing certificate object (certificate, certificate template, private keys) must be scrubbed from existence.**

An under-considered factor is the importance of account privileges during an incident investigation. **Ensuring that an account is appropriately permissioned to remove certificates is vital to ensure readiness to respond in an incident scenario.**

As mentioned previously, disaster recovery plans must also consider how a Tier 0 system like a Certificate Authority can be managed in an emergency. Contingency plans should be revised, given that compromise of a Certificate Authority instigates a 'cascading failure' of security across an entire organisation. Both Subordinate and Root Certificate Authorities must be given Tier 0 treatment.

It is tempting to utilise Enterprise Administrator in an emergency to give permissions quickly. However an adversary may be able to steal credentials of the Enterprise Administrator if used during an incident. Therefore, an inferior, burnable account should be used - preferably with full permissions over Active Directory Certificate Service but nothing else across the Active Directory. This account should be decommissioned afterwards.

Microsoft detail greater granularity of what to do, step-by-step, to restore trust in Active Directory Certificate Service after a compromise [here](#).

6. Next steps

JUMPSEC encourages organisations to ensure that their Active Directory configuration is suitably robust ahead of the likely increase in adversarial attention to Active Directory in the face of SpecterOps' research.

Broader configuration issues outside of Active Directory Certificate Service are also likely to surface with this additional scrutiny. **Therefore, organisations that use this research as the catalyst for a broader review of the resilience of their Active Directory environment will be best prepared for what comes next.**

JUMPSEC advises that all organisations using Active Directory Certificate Service look to implement the remediations advised in this document.

The guidance provided is designed to be implemented without third-party assistance; however, should you require additional advice or support, **please reach out to us using the contact form on the webpage.**

This is a live article and as such will be continually updated as the recommendations are refined and improved as new information surfaces. Follow JUMPSEC's live guidance [@jumpsecLabs](#).



Unit 3E-3F, 33-34 Westpoint,
Warple Way, Acton, W3 0RG

0333 939 8080

hello@jumpsec.com

www.jumpsec.com