MANDIANT
NOW PART OF Google Cloud

MANDIANT CONSULTING
WHITE PAPER

Active Directory Certificate Services: Modern Attack Paths, Mitigations, and Hardening

Table of Contents

Background 3
Permissions 4

Common Abuse Scenarios 5
Subject Alternative Names (SANs) 5
Golden Certificate Attack 6
Permissions 6
Web Enrollment and NTLM Relay Attacks 6
CVE-2022-26923 7

Changes in Certificate-Based Authentication after Windows Update (KB5014754) 9
Multiple AD CS CVEs 9
Changes to Account Mapping 9
Weak vs. Strong Mappings 9
Compatibility Mode 11
Full Enforcement Mode 12
Disabled Mode 12

Continued Attack Vectors Post KB5014754 13
Custom SAN Abuse 13

Hardening and Visibility Recommendations 14

Hunting for CVE-2022-26923 Exploitation Attempts 20

Query AD CS Servers to Identify Misconfigured Templates 22

Background

Microsoft's Active Directory Certificate Services (AD CS) platform provides public key infrastructure (PKI) functionality to facilitate various capabilities including Encrypting File System (EFS), domain authentication, digital signatures, and email security. Certificates are issued by AD CS Certification Authorities (CAs) after receiving a certificate signing request (CSR) that is generated by a user or machine, based on published certificate templates. Certificate templates define parameters such as certificate validity, certificate usage, subject name and issuance requirements to validate a certificate requestor's identity.

The SpecterOps team first highlighted many AD CS abuse scenarios in 2021. Since that publication, Mandiant has continued to observe both threat actors and red teamers enhance targeting of AD CS in support of post-compromise objectives. This document provides an overview of common AD CS abuse scenarios (Table 1), newly introduced measures to mitigate potential abuse methods (primarily KB5014754), and general best practices for hardening and enhancing visibility of infrastructure that leverages AD CS. This document provides an overview of common AD CS abuse scenarios (Table 1), newly introduced measures to mitigate potential abuse methods (primarily KB5014754), and general best practices for hardening and enhancing visibility of infrastructure that leverages AD CS.

TABLE 1. Common AD CS abuse scenarios and associated mitigations / hardening

AD CS Abuse Scenarios	Associated Mitigations / Hardening
Custom SAN Abuse (Pre KB5014754)	Installation of KB5014754, with full enforcement of Strong Mappings for issued certificates Hardening and Visibility Recommendations • Remove the EDITF_ATTRIBUTESUBJECTALTNAME2 flag from any CA's where this is enabled • Disable the use of custom SANs on templates where this is allowed. • Enforce "CA Certificate Manager approval" for any published templates that allow for a SAN as an issuance requirement.
Custom SAN Abuse (Post KB5014754)	Hardening and Visibility Recommendations • Enforce "CA Certificate Manager approval" for any published templates that allow for a SAN as an issuance requirement. • Disable the use of custom SANs on templates where this is allowed. • Do not disable strong mapping compatibility/enforcement mode.
Certificate Template and CA Permissions	Hardening and Visibility Recommendations • Review published templates configured with EKUs that support domain authentication. • For templates with sensitive EKUs, limit enrollment permissions to predefined users or groups. • Access control lists for templates should be audited to ensure that they align with the principle of least privilege. • Enforce "CA Certificate Manager approval" for any published templates that allow for domain authentication. • Treat both root and subordinate certificate authorities as Tier 0 assets and enforce logon restrictions

TABLE 1. Common AD CS abuse scenarios and associated mitigations / hardening

AD CS Abuse Scenarios	Associated Mitigations / Hardening
Permissions **Template Permissions** Access controls on certificate templates are maintained using security descriptors (the same as any AD object), these define security principals and associated permissions assigned through access control entries (ACEs). Throughout investigations, Mandiant has observed published certificate templates that were misconfigured with overly permissive ACEs, allowing for non-privileged users to edit the properties of a certificate template, thus providing a potential path for privilege escalation. One example would be enabling the flag (CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT_ALT_NAME) that allows for custom SANs to be leveraged. As these abuse vectors are categorized as misconfigurations, patches have not been released by Microsoft. **Note:** These abuse vectors do not exist in a default installation of Active Directory Certificate Services, and require specific changes that result in vulnerable configurations. **CA Server Permissions** In addition to overly permissive certificate templates, misconfigured permissions for the CA server(s) can provide an escalation path. It is important to ensure that the CA computer object in AD is within a well-controlled OU, where the ability to edit attributes related to the object are restricted. The CA server(s) should also be treated as a Tier 0 asset, with a limited scope of accounts that are assigned administrative permissions for the Operating System and the certificate services. Web Enrollment (NTLM Relay Attack)	Hardening and Visibility Recommendations • Enable Extended Protection for Authentication (EPA) for Certificate Authority Web Enrollment and the Certificate Enrollment Web Service. • Enforce AD CS to support only HTTPS connections.
Golden Certificate Attack	Hardening and Visibility Recommendations • Treat both root and subordinate certificate authorities as Tier 0 assets and enforce logon restrictions. • To avoid the theft of a CA's private keys (e.g., via the DPAPI backup protocol), protect the private keys by leveraging a Hardware Security Module (HSM). • Enforce multi-factor authentication (MFA) for CA and AD management and operations.
CVE-2022-26923	Installation of KB5014754, with Compatibility Mode or full enforcement of Strong Mappings for issued certificates.

Common Abuse Scenarios

Subject Alternative Names (SANs)

Active Directory (AD) can perform certificate-based authentication using the Kerberos and Secure Channel protocols. When authentication is attempted against AD with a certificate, the Domain Controller will check to ensure the certificate is valid (e.g., was issued/signed by a known Certificate Authority and is within the defined validity period) and will then look for a mapping that ties the certificate to a given account (the method used can vary depending on the account type).

By default, certificate templates are configured to populate the subject from information already present in AD based on which account requests the certificate but this information can be customized in the CSR if a template allows for the "`Supply in the request`" option. This setting enables the requester to specify a custom subject alternative name (SAN) which is used during the authentication process.

Certificate mapping is used to correlate a given identity with a presented certificate. There are two ways in which AD maps a certificate to an identity, using either implicit or explicit mappings. An implicit mapping uses the UPN or DNS host name configured in the SAN extension, whereas an explicit mapping would leverage the `altSecurityIdentities` attribute of an AD identity and allow an administrator to explicitly tie a given certificate to a user (removing the need for any automated mapping).

When a template allows for authentication and for the CSR to specify the SAN, any account with enrollment rights can provide an arbitrary UPN, including that of an account with higher privileges, which will then be used by AD when authentication occurs. When certificate templates are configured for auto enrollment (e.g., no approval is required) this allows for privilege escalation from the enrolling (non-privileged) user to the role of Domain Admin or any other administrative account.

A high-level overview of this attack path is depicted in Figure 1.

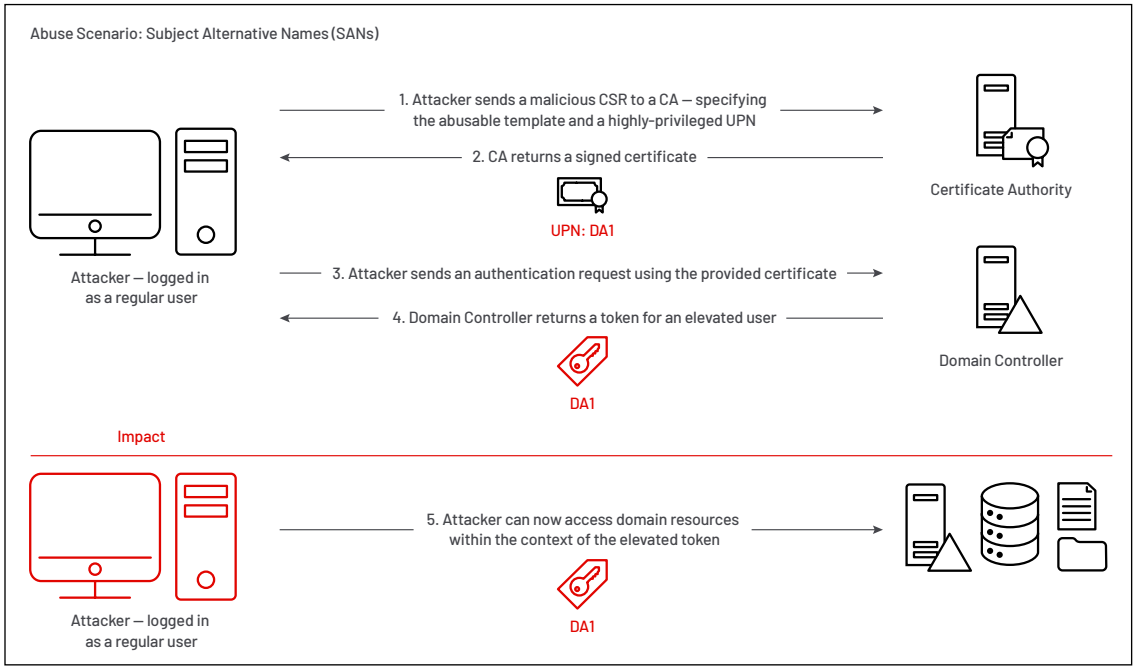

FIGURE 1. Attack path example of SAN abuse for privilege escalation

Golden Certificate Attack

As part of the standard certificate request and issuance process, a certificate authority (CA) will issue a certificate based upon receiving a valid certificate signing request (CSR) from a user or machine. The issuance process includes the CA signing the certificate using its own private key (generated when the CA is created), allowing AD to then validate the integrity of issued certificate using the CA's published public certificate.

By default, the CA's private key is protected by the Data Protection API (DPAPI) which encrypts the CA private key using a symmetric key derived from the local computer account credentials. The CA's private key is marked as exportable, so any users that have local administrator privileges on the CA server can access the private key using the Certificate Services MMC (Figure 2) or by using tools such as Mimikatz or SharpDPAPI.

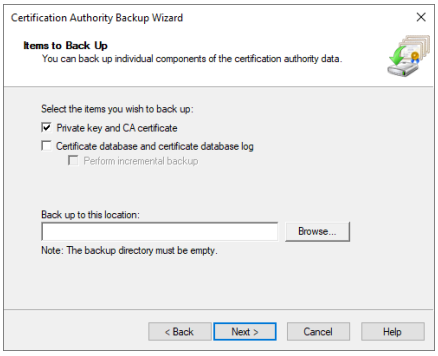

FIGURE 2. MMC method to backup a CA's private key and certificate

If an attacker gains access to a CA server and can extract the CA private key, the attacker could then maliciously forge and sign certificate requests, providing the ability to impersonate any active account within the AD domain / forest for as long as the CA certificate is valid. Once the private key is stolen, the malicious issuance process can be completed offline, therefore leaving no trace of abuse in the AD infrastructure. This technique would allow the attacker to persist, even if passwords for accounts were rotated.

Permissions

Template Permissions

Access controls on certificate templates are maintained using security descriptors (the same as any AD object), these define security principals and associated permissions assigned through access control entries (ACEs). Throughout investigations, Mandiant has observed published certificate templates that were misconfigured with overly permissive ACEs, allowing for non-privileged users to edit the properties of a certificate template, thus providing a potential path for privilege escalation. One example would be enabling the flag (`CT_FLAG_ENROLLEE_SUPPLIES_SUBJECT_ALT_NAME`) that allows for custom SANs to be leveraged.

As these abuse vectors are categorized as misconfigurations, patches have not been released by Microsoft.

Note: These abuse vectors do not exist in a default installation of Active Directory Certificate Services, and require specific changes that result in vulnerable configurations.

CA Server Permissions

In addition to overly permissive certificate templates, misconfigured permissions for the CA server(s) can provide an escalation path. It is important to ensure that the CA computer object in AD is within a well-controlled OU, where the ability to edit attributes related to the object are restricted. The CA server(s) should also be treated as a Tier 0 asset, with a limited scope of accounts that are assigned administrative permissions for the Operating System and the certificate services.

Web Enrollment and NTLM relay attacks

Any web enrollment services that are configured (CA Web Enrollment, Certificate Web Services, Device Enrollment Services), can potentially be used for NTLM relay attacks. An attacker can leverage attack vectors like PetitPotam to coerce authentication from systems such as Domain Controllers, relay them through AD CS web services, and request a legitimate client authentication certificate. This can result in a privilege escalation scenario.

CVE-2022-26923

At the end of 2021, a new privilege escalation vulnerability for AD CS and Active Directory (AD) was reported to Microsoft and eventually assigned a CVE number (CVE-2022-26923). Exploitation of the vulnerability relies on the following AD attributes:

- `dNSHostName`: An attribute of Computer objects that specifies the corresponding entry registered in DNS. Accounts with write permissions for computer objects in AD can modify this attribute to be any computer account name, as the value did not need to be unique in the domain.
- `SubjectAltRequireDNS`: For certificate templates enabled with the `SubjectAltRequireDNS` flag (specified in the `msPKI-Certificate-Name-Flag` attribute), certificates can be requested for authentication based on `dNSHostName` instead of User Principal Name (UPN). A common example of this are default machine certificate templates used by computer accounts to authenticate with AD.
- `MS-dS-MachineAccountQuota`: This AD attribute defines the number of computer objects that can be created by authenticated domain accounts. In a default AD configuration, any domain account can create up to ten (10) computer objects in the domain.

After gaining access to domain credentials which provide the ability to communicate to a Domain Controller and Certificate Authority server, an attacker can, by default, create a computer object in the domain and, prior to patching, modify the `dNSHostName` attribute of the newly created object to reference a privileged computer account name (e.g., a Domain Controller). An attacker could then request a certificate for authentication using a published certificate template configured with the `SubjectAltRequireDNS` flag. The issued certificate would then allow an attacker to access resources within the context of the privileged computer account name that was referenced within the `dNSHostname` attribute.

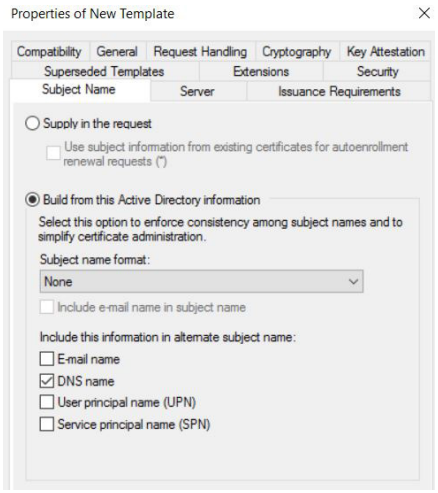

FIGURE 3. Certificate template with the SubjectAltRequireDNS flag

A high-level overview of this attack path is depicted in Figure 4.

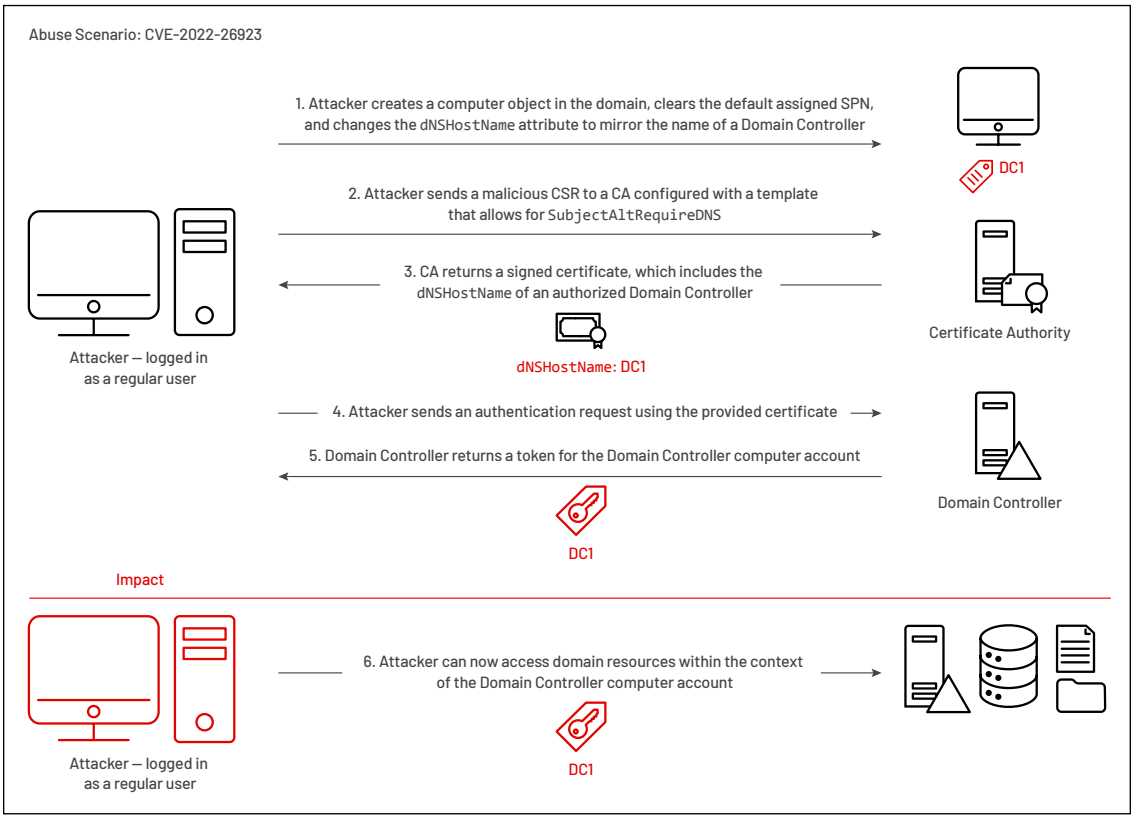

FIGURE 4. Attack path example of CVE-2022-26923 for privilege escalation

This abuse scenario resulted in Microsoft releasing a patch for AD CS and AD in May 2022 (KB5014754), which enforces more stringent checks and validation when certificate-based authentication is leveraged within Active Directory as well as enforcing additional validation against the **dNSHostName** attribute when access permissions are restricted to "Validated Write".

While numerous configuration scenarios can exist within an AD CS environment, the rest of this document primarily focuses on the changes enforced via KB5014754.

Changes in Certificate-Based Authentication after Windows Update (KB5014754)

Multiple AD CS CVEs

Throughout 2022, three (3) specific CVEs related to AD CS were released, prompting Microsoft to release a patch (KB5014754) and roadmap actions for mitigating AD CS vulnerabilities.

- CVE-2022-34691
- CVE-2022-26931
- CVE-2022-26923

Changes to Account Mapping

KB5014754 introduced new concepts related to how certificates are mapped (weak vs strong) to users during AD authentication. The idea behind these changes was to enforce a stronger tie between the requester and the issued certificate, flag discrepancies, and ultimately block access when authentication is attempted from weak, or incorrectly mapped certificates.

Weak vs. Strong Mappings

The primary difference between weak and strong mappings relate to the concept of uniqueness within a given domain. Only attributes that are explicitly unique will be categorized as **strongly mapped** whereas all other mappings are now categorized as **weakly mapped** (due to the potential for duplicates to exist).

For explicit mappings, of the six (6) supported values for `altSecurityIdentities`, only three (3) mapping types are considered strong (Table 2).

TABLE 2. `altSecurityIdentities` supported values and mapping categorization

`altSecurityIdentities` Supported Values	Mapping Categorization
• Issuer Serial Number (X509IssuerSerialNumber) • Security Key Identifier (X509SKI) • SHA1 Hash of the public key (X509SHA1PublicKey)	Strong
• Issuer and Subject Name (X509IssuerSubject) • Subject Name (X509SubjectOnly) • Email Address (X509RFC822)	Weak

For implicit mappings, Microsoft introduced an automatic mechanism for strong mapping with the addition of a new certificate attribute labeled `szOID_NTDS_CA_SECURITY_EXT(1.3.6.1.4.1.311.25.2)`. This attribute is **automatically** populated with the AD Security Identifier (SID) of the requesting account when a certificate is issued by the CA. If the UPN or DNS name specified in the SAN **do not** match the requester SID when authentication occurs, Domain Controllers will flag, and ultimately reject (once enforced) the authentication attempt.

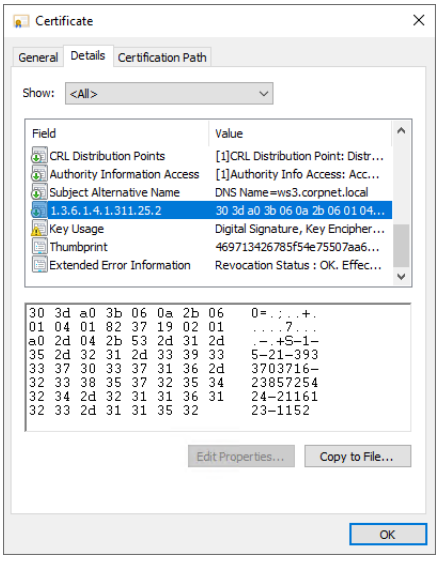

FIGURE 5. KB5014754 - new attribute for implicit mapping

After KB5014754 is applied, CAs will start including the new extension by default in all certificates issued against online certificate templates **except** for templates that allow for custom SANs.

Figure 6 provides a high-level overview of the mitigations enforced via KB5014754.

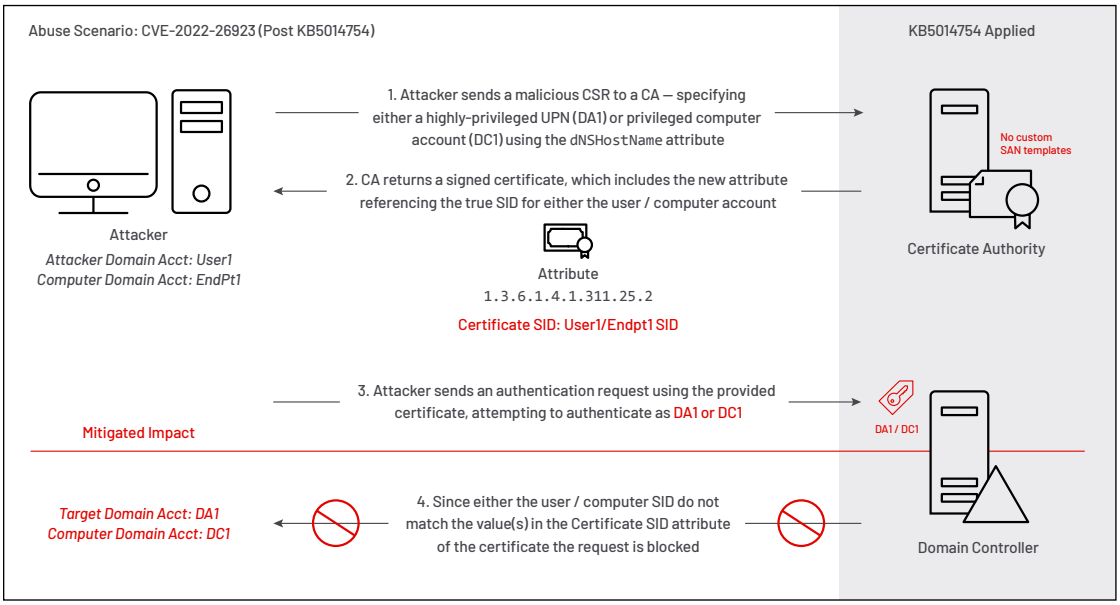

FIGURE 6. KB5014754 mitigated impact

Note: Organizations can prevent the addition of the extension by setting the `0x00080000` bit for the `msPKI-Enrollment-Flag` value in the certificate template, but this will break certificates once strong mapping is automatically enforced (currently scheduled for February 2025). This can be configured via PowerShell using the code snippet shown in Figure 7 (which requires the `ActiveDirectory PowerShell` module).

```
$TemplateName = '<insert-name-of-certificate>'

$TemplateParams = @{
    'Filter'     = { ObjectClass -eq "PKIcertificateTemplate" -and DisplayName -eq $TemplateName}
    'SearchBase' = (Get-ADRootDSE).ConfigurationNamingContext
    'Properties' = 'DisplayName','DistinguishedName','pKIExtendedKeyUsage',
                   'msPKI-Certificate-Name-Flag','nTSecurityDescriptor','msPKI-Enrollment-Flag'
}
$Template = Get-ADObject @TemplateParams -ErrorVariable $ErrVar

$NewValue = $Template.'msPKI-Enrollment-Flag' + 0x80000

Set-ADObject -Identity $Template.DistinguishedName -Replace @{'msPKI-Enrollment-Flag'=$NewValue}
```

FIGURE 7. Removing automatic addition of SID to certificate templates

Compatibility Mode

After installing KB5014754 (part of the May 2022 Windows Cumulative Updates) on Certificate Authorities and Domain Controllers, AD will run in "compatibility mode". Per Microsoft, Domain Controllers will validate that the **`sAMAccountName`** of the initiating account matches the computer name in the DNSName field of the computer certificate that is being leveraged for authentication. If there is not a match, the DC will return **`KDC_ERR_CLIENT_NAME_MISMATCH`**, mitigating CVE-2022-26923 for Domain Controllers configured with either compatibility or full enforcement mode.

In addition, new certificate mapping classifications are introduced for **user authentication** in AD. In compatibility mode, user authentication will occur as expected for historically valid certificates, but when a mapping can only be completed using a weak mechanism, a new event (Event ID 39) will be recorded within the System event log on Domain Controllers (Figure 8).

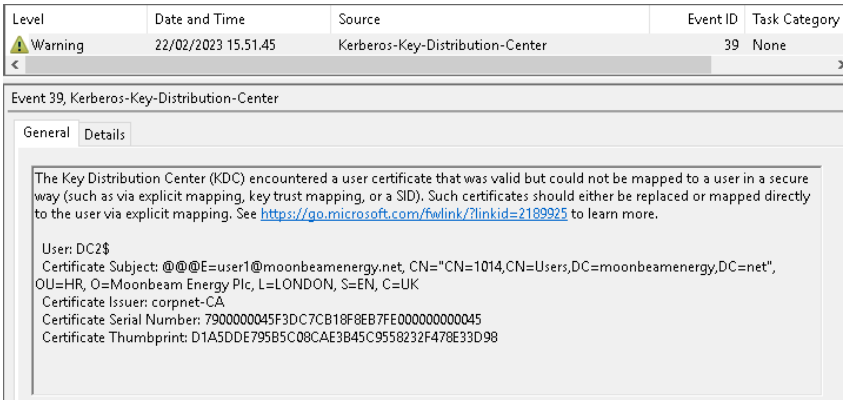

FIGURE 8. Event ID 39 recording a weakly mapped certificate authentication

Additionally, certificates which contain the new OID attribute (and are issued after installing the KB5014754 patch) will be subject to blocking if the SID in this attribute does not match the SID of the authenticating user (i.e., specified in the SAN).This will result in an explicit block, and will be recorded as Event ID 41 within the System event log (Figure 9).

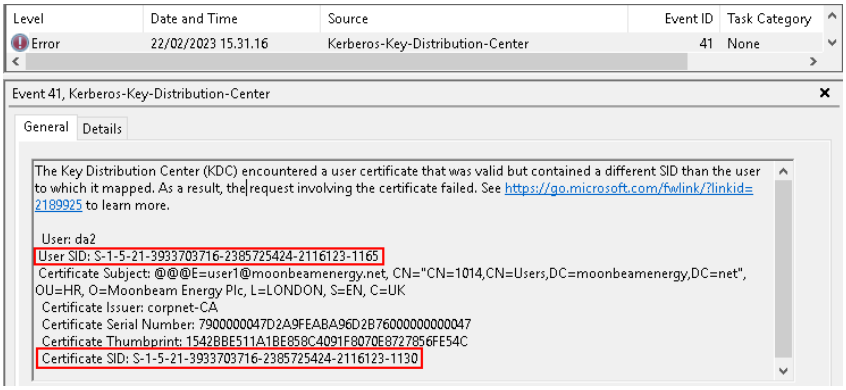

FIGURE 9. Event ID 41 flagging a SID mismatch between requester and authenticating account

After the May 2022 Windows Cumulative Updates are installed across all Domain Controllers and Certificate Authorities, any existing certificates that have been issued for authentication will result in Event ID 39 being recorded, since the certificates were issued before the new attribute (**1.3.6.1.4.1.311.25.2**) existed. These will need to be renewed or manually mapped prior to enforcement to ensure smooth authentication post enforcement.

Full Enforcement Mode

Per Microsoft, full enforcement mode will be enforced for certificate authentication beginning on February 11, 2025. This will require that strong mapping is used for all authentication, and any certificates with weak mappings will be denied.

Note: Organizations should ensure that all previously issued certificates flagging Event ID 39 are rotated prior to February 11, 2025, as Microsoft plans to enforce blocking of weakly mapped certificates at this time. If an organization has applied the patches and Event Id 39 events are not being generated for certificate authentication, full enforcement mode can be proactively enabled (Figure 10 and Figure 11) prior to the February 11, 2025 date.

Disabled Mode

Disabled mode reverts operations back to pre-patch behavior if issues with weak mappings are discovered. This mode is not recommended and will be removed as an option once full enforcement mode is enforced.

Registry Settings to Control Behavior

Registry	HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Kdc
Value	StrongCertificateBindingEnforcement
Data Type	REG_DWORD
Data	0 = Disabled 1 = Compatibility (default) 2 = Enforced

FIGURE 10. Registry key to change Kerberos allowed certificate authentication modes

Registry	HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\SecurityProviders\Schannel
Value	CertificateMappingMethods
Data Type	DWORD
Data	0x1F = Allow Strong and Weak mapping 0x18 = Strong mapping only (default post patching)

FIGURE 11. Registry key to change Schannel allowed certificate authentication modes

Continued Attack Vectors Post KB5014754

Custom SAN Abuse

After the May 2022 Windows Updates, certificate authorities will populate the SID of the certificate requester in the attribute `szOID_NTDS_CA_SECURITY_EXT(1.3.6.1.4.1.311.25.2)` for most newly issued certificates. The pre-described attack method of elevating the privileges by referencing a DNS value in the SAN attribute of a certificate will no longer be successful due to the mismatch of requester SID populated in the `szOID_NTDS_CA_SECURITY_EXT` attribute and the value stored in the SAN (assuming custom SANs are not enabled). Additionally, if a custom UPN is specified when not explicitly allowed within the certificate template (e.g., where `EDITF_ATTRIBUTESUBJECTALTNAME2` is enabled on a CA), authentication attempts will be denied.

While the above helps reduce attack paths, **it is still possible for certificate templates to allow for custom SANs.** If this configuration exists and is accessible to lower privileged accounts, a malicious requester can still specify an account with higher permissions, although the requester must also ensure a SID is specified in the request that matches the UPN being targeted (e.g., provide OID `1.3.6.1.4.1.311.25.2` in the CSR, reference Figure 12). While the structure of this new OID is not completely straightforward, the ability to add this data in the correct format is possible and already integrated into open-source tools.

Based upon the constantly evolving abuse techniques, it is imperative to conduct regular auditing and monitoring to detect any misconfigurations in certificate templates.

```
[ req ]
prompt                 = no
days                   = 365
distinguished_name     = req_distinquished_name
req_extensions         = v3_req

[ req_distinguished_name ]
countryName            = US
stateOrProvinceName    = VA
organizationName       = Moonbean Energy Plc
commonName             = "CN=1014, CN=Users, DC=moonbeamenergy. DC=net"
emailAddress           = user1@moonbeamenergy.net

[ v3_req ]
basicConstraints       = CA:false
extendedKeyUsage       = clientAuth
subjectAltName         = otherName:msUPN;UTF8:da2moonbeamenergy.net
1.3.6.1.4.1.311.25.2   = DER:303DA03B060A2B060104018237190201A02D0...
```

FIGURE 12. Example OpenSSL CSR including the new OID with SID embedded

Hardening and Visibility Recommendations

1. Ensure that all Domain Controllers and Certificate Authority servers are patched with the latest updates and hotfixes.
2. After installing Windows update (KB5014754) and monitoring/remediating for Event IDs 39 and 41, configure Active Directory to support full enforcement mode to reject authentications based on weaker mappings in certificates.
3. Regularly review published certificate templates, specifically for any settings related to SAN specifications configured in existing templates.

   The command referenced in Figure 13 can be leveraged to export existing templates from a Windows certificate authority server.

```
certutil.exe -TCAInfo
```

FIGURE 13. Windows Command line program to display certificate template

4. Review the security permissions assigned to all published certificate templates and validate the scope of enrollment and write permissions are delegated to the correct security principals.

   The command referenced in Figure 14 can be leveraged to export the permissions for existing templates from a Windows certificate authority server. Alternatively, reference the Query AD CS Servers to Identify Misconfigured Templates section of this document, which includes PowerShell code that can be leveraged to identify potentially misconfigured templates.

```
certutil.exe -v -dstemplate
```

FIGURE 14. Windows Command line program to display permissions of the published certificate templates

5. Review published templates configured with the following Enhanced Key Usages (EKUs) that support domain authentication and verify the operational requirement for these configurations.

   - `Any Purpose (2.5.29.37.0)`
   - `Subordinate CA (None)`
   - `Client Authentication (1.3.6.1.5.5.7.3.2)`
   - `PKINIT Client Authentication (1.3.6.1.5.2.3.4)`
   - `Smart Card Logon (1.3.6.1.4.1.311.20.2.2)`

6. For templates with sensitive Enhanced Key Usage (EKU), limit enrollment permissions to predefined users or groups, as certificates with EKUs can be used for multiple purposes. Access control lists for templates should be audited to ensure that they align with the principle of least privilege.

   Templates that allow for domain authentication should be carefully reviewed to verify that built-in groups that contain a large scope of accounts are not assigned enrollment permissions. Example: built-in groups that could increase the risk for abuse include:

   - `Everyone`
   - `NT AUTHORITY\Authenticated Users`
   - `Domain Users`
   - `Domain Computers`

7. Where possible, enforce "`CA Certificate Manager approval`" (Figure 15) for any templates that include a SAN as an issuance requirement. This will require that any certificate issuance requests be manually reviewed and approved by an identity assigned the "`Issue and Manage Certificates`" permission on a certificate authority server.

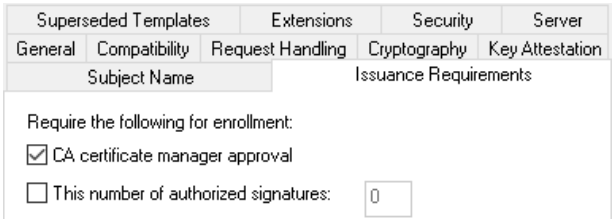

FIGURE 15. CA Certificate Manager approval option

8. Ensure that Certificate Authorities have not been configured to accept any SAN (irrelevant of the template configuration), this is a non-default configuration and should be avoided wherever possible. This abuse vector is mitigated by KB5014754, but until enforcement of strong mappings is enforced, abuse could still occur based upon historical certificates missing the new OID containing the requester's SID. For additional information, reference the following Microsoft article.

   The command in Figure 16 can be used to show the current configuration on a CA, and the command in Figure 17 can be used to disable the ability for SANs to be accepted by a CA as part of a certificate request.

```
certutil.exe -getreg policy
```

FIGURE 16. Windows Command to validate the existence of the flag EDITF_ATTRIBUTESUBJECTALTNAME2.

```
certutil -setreg policy\EditFlags -EDITF_ATTRIBUTESUBJECTALTNAME2
```

FIGURE 17. Windows Command line program to disable SAN feature

Figure 18 contains PowerShell code that can be used to export previously issued certificates from a CA. The output (CSV file) can then be reviewed to identify certificates where the requester and SAN attributes are mismatched.

```
# requires -Module PSPKI (Install-Module -Name PSPKI)
# Get all issued certificates from the specified CA
# May need to filter this for large envs e.g.:Get-IssuedRequest -Filter "NotAfter -ge $(Get-Date)",
"NotAfter -le$((Get-Date).AddMonths(2))"

param (
    [Parameter(Position = 0, Mandatory = $True)]
    [ValidateNotNullOrEmpty()]
    $CA,
    [Parameter(Position = 1, Mandatory = $True)]
    [ValidateNotNullOrEmpty()]
    $OutputPath
)

$IssuedCerts = Get-IssuedRequest -CertificationAuthority $CA -Property rawcertificate

#Iterate through issued certificates and extract pertinent data
$Results = foreach ($IssuedCert in $IssuedCerts){
    $CertBytes = [Convert]::FromBase64String($IssuedCert.RawCertificate)
    $Cert = [System.Security.Cryptography.X509Certificates.X509Certificate2]$CertBytes
    #Where template is an OID, try to map to named templates
    if ($IssuedCert.CertificateTemplate -match '^(?:\d\.){5}'){
        try {
            $Template = (Get-CertificateTemplate -Oid $IssuedCert.CertificateTemplate -ErrorAction
Stop).DisplayName
        } catch {
            $Template = $IssuedCert.CertificateTemplate
        }
    } else {
        $Template = $IssuedCert.CertificateTemplate
    }
    #Export the pertinent fields
    [PSCustomObject] @{
        'Subject' = $Cert.Subject
        'Template' = $Template
        'Requester' = $IssuedCert.'Request.RequesterName'
        'SubjectAlternativeName' = $Cert.GetNameInfo([System.Security.Cryptography.X509Certificates.
X509NameType]::UpnName, $false)
        'Serial' = $IssuedCert.SerialNumber
        'NotBefore' = $IssuedCert.NotBefore
        'NotAfter' = $IssuedCert.NotAfter
    }
}

#List results and look for mismatches between requestor and SAN
$Results | Export-Csv -NoTypeInformation -Encoding utf8 -Path $OutputPath
```

FIGURE 18. PowerShell script to export previously issued certificates from a CA

9. Treat both root and subordinate certificate authorities as Tier 0 assets and enforce logon restrictions or authentication policy silos to limit the scope of accounts that have elevated access to the servers where certificate services are installed and configured.
10. Audit and review the `NTAuthCertificates` container in AD to validate the referenced CA certificates, as this container references CA certificates that enable authentication within AD. Before authenticating a principal, AD checks the `NTAuthCertificates` container for the CA specified in the authenticating certificate's `Issuer` field to validate the authenticity of the CA. If rogue or unauthorized CA certificates are present, this could be indicative of a security event that requires further triage and investigation. The code snippet in Figure 19 will list the current CAs that can be used for authentication.

```
$params = @{
  'Identity' = "CN=NTAuthCertificates,CN=Public Key Services,CN=Services,$((Get-ADRootDSE).
ConfigurationNamingContext)"
  'Properties' = 'caCertificate'
}
[array]$NTAuth = Get-ADObject @params

$NTAuth | ForEach-Object {
  [Security.Cryptography.X509Certificates.X509Certificate2]::new($($_.caCertificate))
}
```

FIGURE 19. Code to extract a list of certificate authorities approved for AD authentication

11. To avoid the theft of a CA's private keys (e.g., via the DPAPI backup protocol), protect the private keys by leveraging a Hardware Security Module (HSM) on servers where certificate authority services are installed and configured.
12. Enforce multi-factor authentication (MFA) for CA and AD management and operations.
13. Keep the root CA offline and use subordinate CAs to issue certificates.
14. Leverage the PowerShell script referenced in the Query AD CS Servers to Identify Misconfigured Templates section of this document to validate and identify potential misconfigurations within existing certificate templates. Alternatively, public tools (e.g., PSPKIAudit or Certify) can be used to validate and identify misconfigurations in certificate templates although be aware they may be flagged by EDR products as they are frequently used by red teams and threat actors.
15. To mitigate NTLM Relay attacks in AD CS, enable Extended Protection For Authentication for Certificate Authority Web Enrollment and Certificate Enrollment Web Service. Additionally, require that AD CS accept only HTTPS connections. For additional details, reference the following Microsoft Article.

16. Enable audit logging for certificate services on CA servers by using group policy (Figure 20).

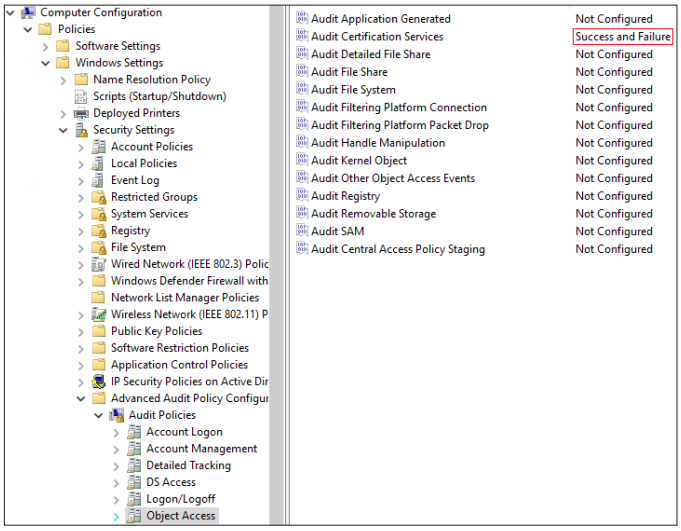

FIGURE 20. Audit both successful and failed events by using Group Policies

17. Enable the audit filter on each CA (Figure 21 or Figure 22). This is a bitmask value that represents the seven (7) different audit categories that can be enabled, if all values are enabled, the audit filter will have a value of 127.

    Note: The certificate service must be restarted after modifying the audit filters.

```
certutil –setreg CA\AuditFilter 127
```

FIGURE 21. Windows command line to enable all the audit categories

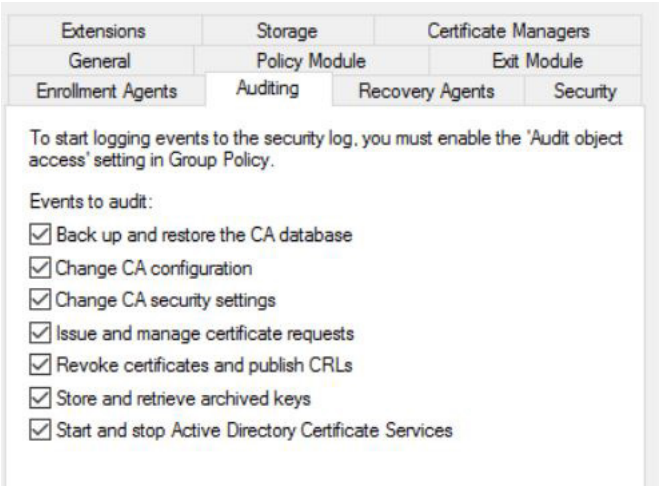

FIGURE 22. Event categories to audit

18. Log and monitor the events referenced in Table 3 to enhance detections related to AD CS activities (steps 16 and 17 are needed to ensure the appropriate logs are generated).

TABLE 3. Relevant Windows Event IDs for AD CS activities

System Type	Log Source	Event ID	Description
CA	Security	4868	The certificate manager denied a pending certificate request
CA	Security	4870	Certificate Services revoked a certificate.
CA	Security	4876	Certificate Services backup started.
CA	Security	4877	Certificate Services backup completed.
CA	Security	4880	Certificate Services started.
CA	Security	4881	Certificate Services stopped.
CA	Security	4882	The security permissions for Certificate Services changed.
CA	Security	4885	The audit filter for Certificate Services changed.
CA	Security	4886	Certificate Services received a certificate request
CA	Security	4887	Certificate Services approved a certificate request and issued a certificate
CA	Security	4889	Certificate Services set the status of a certificate request to pending
CA	Security	4890	The certificate manager settings for Certificate Services changed.
CA	Security	4891	A configuration entry changed in Certificate Services.
CA	Security	4892	A property of Certificate Services changed.
CA	Security	4895	Certificate Services published the CA certificate to Active Directory Domain Services.
CA	Security	4899	A Certificate Services template was updated.
CA	Security	4900	Certificate Services template security was updated.
Domain Controller	Security	4768	Kerberos Authentication Ticket request and this includes details of the "Certificate Information" fields with the authenticating certificate's Issuer, Serial Number, and Thumbprint
Domain Controller	Security	4674	An operation is attempted on a privileged object. This event will get triggered for any changes in security descriptors of the certificate templates. The object name in the event will list the certificate template name
Domain Controller	Security	39	No strong certificate mappings could be found, and the certificate did not have the new security identifier (SID) extension that the KDC could validate. *Warning* if the KDC is in Compatibility mode *Error* if the KDC is in Enforcement mode
Domain Controller	Security	40	The certificate was issued to the user before the user existed in Active Directory and no strong mapping could be found. This event is only logged when the KDC is in Compatibility mode.
Domain Controller	Security	41	The SID contained in the new extension of the users certificate does not match the users SID, implying that the certificate was issued to another user

Hunting for CVE-2022-26923 Exploitation Attempts

Once auditing is configured, specific event IDs can be used to identify potential evidence of an attacker exploiting CVE-2022-26923.

As part of the exploitation path, an attacker can clear an associated service principal name (SPN) for their managed computer object in the AD domain. Event ID 4742 (recorded in the Security log of a Domain Controller) will capture any changes to a domain computer object. These recorded events can be used to identify computer objects with changes to the `Service Principal Name` attribute from a populated value to `<value not set>` (Figure 23). It is also possible that an attacker will just remove the FQDN references, therefore log monitoring should be focused on reviewing computer accounts where the `HOST/computer.domain` entries are missing from the `Service Principal Name` value.

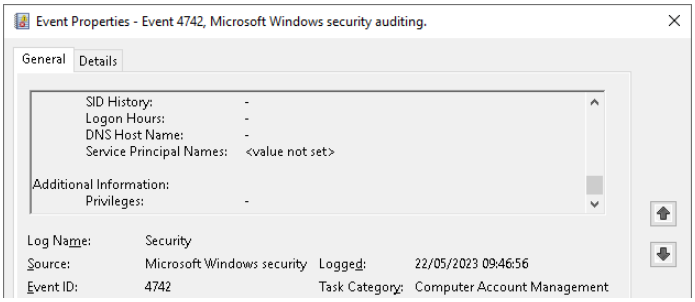

FIGURE 23. Example EID 4742 event for a computer object where the Service Principal Name value was modified

An attacker may also create computer accounts without an associated Service Principal Name (SPN), as default settings in AD will allow authenticated users to create up to ten (10) computer objects. Event ID 4741 (recorded in the `Security` log of a Domain Controller) will record the creation of computer objects. These recorded events can be used to identify newly created computer objects where a Service Principal Name is not set (Figure 24).

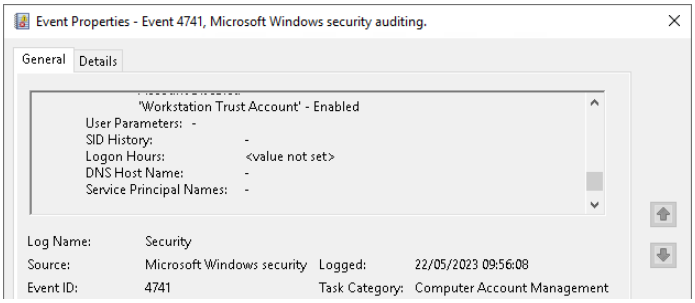

FIGURE 24. Example Event ID 4741 event for a newly created computer object where the Service Principal Name value was not set

An attacker may change the **dNSHostName** attribute value to that of a privileged computer account. Event ID 4742 will also record any changes to a computer object's **dNSHostName** value, which can be used to identify computer accounts with a modified value (Figure 25).

FIGURE 25. Example Event ID 4742 event for a computer object where the dNSHostName attribute was modified

After changing the value in **dNSHostName** to mirror a privileged computer account's dNSHostName, an attacker could potentially request a computer certificate based on the modified **dNSHostName** if there exists a published certificate template configured with the **SubjectALTRequireDNS** flag.

To identify this activity, Event ID 4887 (in the **Security** log of a Certificate Authority server) will record both the requester and subject value within the event. On successful exploitation of CVE-2022-26923, a mismatch will exist between the requester's managed computer object and the subject value generated from the **dNSHostName** attribute (Figure 26).

FIGURE 26. Example EID 4887 event for a certificate request and issuance

Query AD CS Servers to Identify Misconfigured Templates

Figure 27 contains PowerShell code that can be used to identify abusable certificate templates, print them to the console, then export the full list of templates as a CSV file to the location specified on line 2.

For organizations that have multiple certificate authorities, it is recommended to prioritize addressing misconfigurations for the active templates deployed to these authorities (e.g., `DeployedToCAs` is not blank).

```
# Define output location and file
$OutputFile = 'C:\path\to\file.csv'

# Extract certificates
$NamingContext = (Get-ADRootDSE).ConfigurationNamingContext
$TemplateParams = @{
  'Filter'     = { ObjectClass -eq 'PKIcertificateTemplate' }
  'SearchBase' = $NamingContext
  'Properties' = 'DisplayName','DistinguishedName','pKIExtendedKeyUsage',
                 'msPKI-Certificate-Name-Flag','nTSecurityDescriptor',
                 'msPKI-Enrollment-Flag','msPKI-RA-Signature'
}
$Templates = Get-ADObject @TemplateParams -ErrorVariable $ErrVar

$CAObjs = Get-ChildItem "AD:/CN=Enrollment Services,CN=Public Key
Services,CN=Services,$NamingContext"
[array]$CAs = foreach ($CA in $CAObjs){
  Get-ADObject $CA.distinguishedName -Properties dnshostname,certificateTemplates |
      Select-Object Name,dnshostname,ObjectClass,certificateTemplates,ObjectGuid
}

$KeyUsage = @('2.5.29.37.0','1.3.6.1.5.5.7.3.2','1.3.6.1.5.2.3.4','1.3.6.1.4.1.311.20.2.2')

$TemplateStatus = foreach ($Template in $Templates){
  foreach ($Use in $Template.pKIExtendedKeyUsage){
    if ($KeyUsage -contains $Use){
      $AbusableUse = $True
      Break
    } else {
      $AbusableUse = $False
    }
  }

  # Check for user assignable SAN's
  if ($Template.'msPKI-Certificate-Name-Flag' -band 0x1){
    $SupplySAN = $True
  } else {
    $SupplySAN = $False
  }

  # Check for Manager Approval
  if ($Template.'msPKI-Enrollment-Flag' -band 0x2){
    $NoApprovalReqd = $False
  } else {
```

```
    $NoApprovalReqd = $True
  }

  # Check if authorisation signatures required
  if (($Template.'msPKI-RA-Signature' -eq 0) -or ($null -eq $Template.'msPKI-RA-Signature')){
    $NoSigsReqd = $True
  } else {
    $NoSigsReqd = $False
  }

  # Check excessive permissions (non-default Enrolment or GenericAll/Write access)
  $ExcessPermissions = $Template.nTSecurityDescriptor.Access |
    Where-Object {
      ($_.ObjectType -eq '0e10c968-78fb-11d2-90d4-00c04f79dc55' -or
      $_.ActiveDirectoryRights -match 'GenericAll|WriteDacl|WriteOwner|WriteProperty') -and
      $_.IdentityReference -notmatch 'Domain Admins|Enterprise Admins|Domain Controllers'
    }
  if ($ExcessPermissions){
    $NonAdminPerms = $True
  } else {
    $NonAdminPerms = $False
  }

  # Check if template deployed to any CA's
  if ($CAs){
    $DeployedToCAs = New-Object -TypeName 'System.Collections.ArrayList'
    foreach ($CA in $CAs){
      if ($CA.certificateTemplates -contains $Template.Name){
          [void]$DeployedToCAs.Add("$($CA.Name) [$($CA.dnshostname)]")
      }
    }
  } else {
      $DeployedToCAs = $null
  }

  # Collate results
  [PSCustomObject] @{
      TemplateName      = $Template.DisplayName
      AbusableUseType   = $AbusableUse
      UserSuppliedSAN   = $SupplySAN
      NonAdminPerms     = $NonAdminPerms
      NoManagerApproval = $NoApprovalReqd
      NoAuthSignatures  = $NoSigsReqd
      DeployedToCAs     = $DeployedToCAs -join '; '
  }
}

# Print easily abusable templates to console
$TemplateStatus | ForEach-Object {
  if (
    $_.AbusableUseType -eq $True -and
    $_.UserSuppliedSAN -eq $True -and
    $_.NonAdminPerms -eq $True -and
```

```
    $_.NoManagerApproval -eq $True -and
    $_.NoAuthSignatures -eq $True
  ){
    $_
  }
}

# Export all results to CSV
$TemplateStatus | Export-Csv -NoTypeInformation -Encoding utf8 -Path $OutputFile
```

FIGURE 27. PowerShell code to identify potentially abusable CA templates