

Implementation Note:

Jamf Active Directory Certificate Services Connector

Jamf Implementation Engineering
17 August 2020/ol



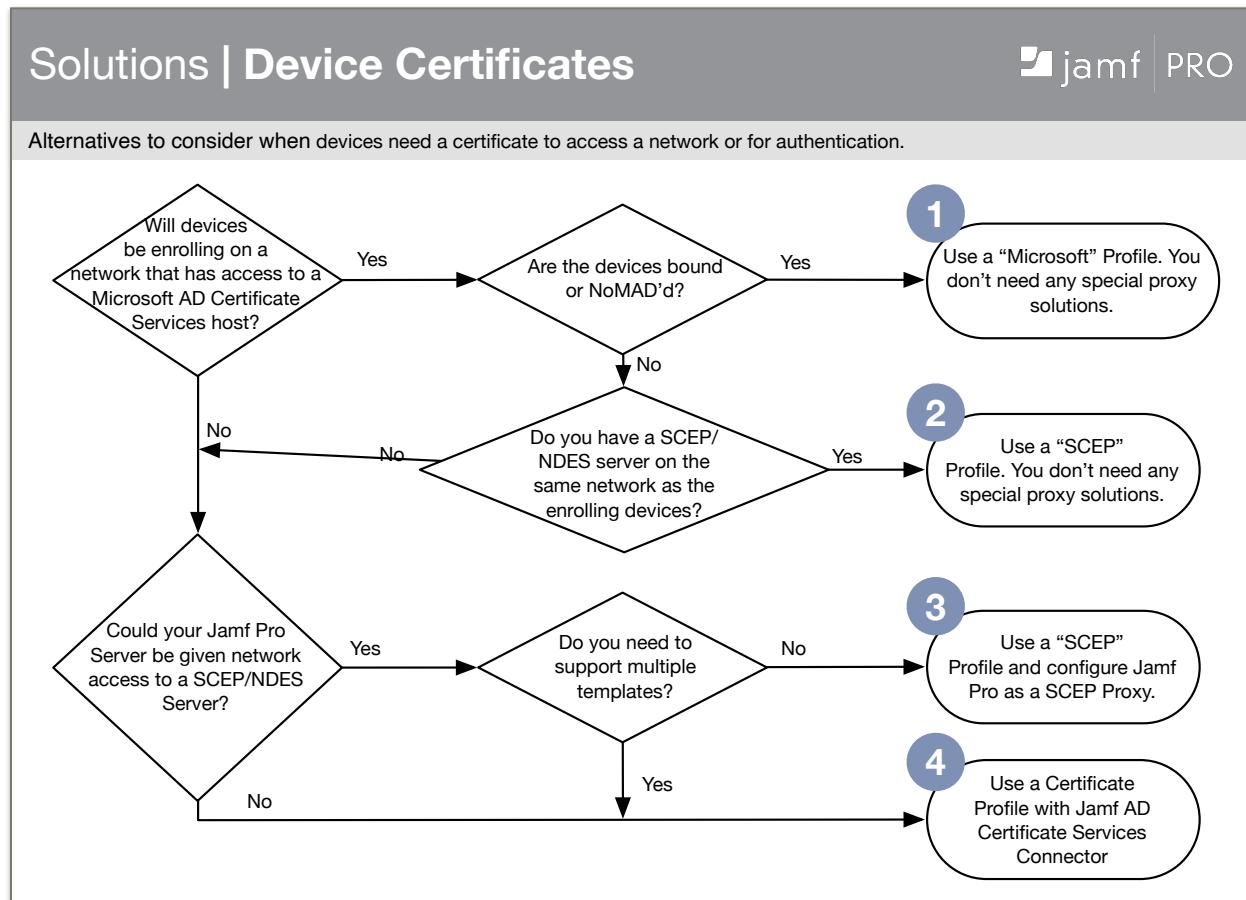
CONTENTS

Options for Deploying Device Certificates.....	1
ADCSC – Background	2
Network Connections.....	3
Jamf Cloud with Jamf ADCS Connector in the DMZ	3
Jamf Cloud with a DMZ Reverse Proxy Layer	3
Self-hosted Jamf Pro Server in the DMZ	3
Network Zones and Firewall Configuration	4
Security Mindset.....	5
Introduction	5
ADCSC Communication Authentication and Trust Basis	5
ADCSC and IT Service Security	5
QA Your Firewall Rules!	6
Private Key Chain of Custody / Data at Rest and In Transit	7
Introduction	7
Installation of ADCS Connector – Requirements Summary	8
Software Installation and Network Requirements	8
Microsoft DCOM Binding Requirement	8
Installing the Jamf ADCS Connector	9
1. Download a copy of the installer	9
2. Run the installer	9
3. Give ADCS Connector permission to talk to the CA	11
Creating a Certificate Template in ADCS and Granting Template Permissions	13
Artifacts	17
Next Steps	17
Resulting Configurations	18
Jamf Pro Configuration	24
Example Certificate Profile and Issued Certificate	26
Introduction to ADCS Connector Customizations	27
Installation Script Customization	27
Use a Domain Service Account when Authenticating to ADCS.....	28
Scenario	28
Procedure: Using a Domain Service Account instead of the default Computer Account	28
Running the ADCS Connector without Domain Binding.	30
Configuring IIS to use an alternate Server Certificate	32
Obtaining a Certificate Signing Request	32
Configuring IIS to Use the Alternate Server Identity	33
Replacing a server certificate in IIS prior to expiration	35
Configuring IIS to use an alternate Client Certificate	36

Requirements for Reverse Proxy, Load-Balanced, and Web Application Firewall Network Configuration	39
Configuring ADCS to use an Alternate Port.....	41
Scenario	41
Procedure	41
Troubleshooting Strategies.....	42
Viewing the IIS Access Logs	42
Example Log Entries — 403.16	43
Viewing the Jamf Software Server Logs	44
Troubleshooting with the Certificates MMC and Certreq	44
Verifying Connectivity between Jamf ADCS Connector and Microsoft ADCS with Certutil	45
Check the DCOM Access Group	46
Check the DCOM Access Restriction GPOs	47
Frequently Asked Questions	48
I have Prod, Test, and QA Jamf Pros. Can I use one ADCS Connector to service all of them?	48
Why does Jamf Pro say the password for my client key file is wrong when I try to upload it?	48
Can I use Azure Web Application Proxy?	48

Options for Deploying Device Certificates

When an organization uses Microsoft PKI, Jamf Pro supports four profile-based methods to deploy certificates to devices. The one you use will depend on your circumstances.



Options 1 and 2 are traditional methods that require devices to enroll on an internal network. Options 3 and 4 work where enrolling devices are not initially on an internal network.

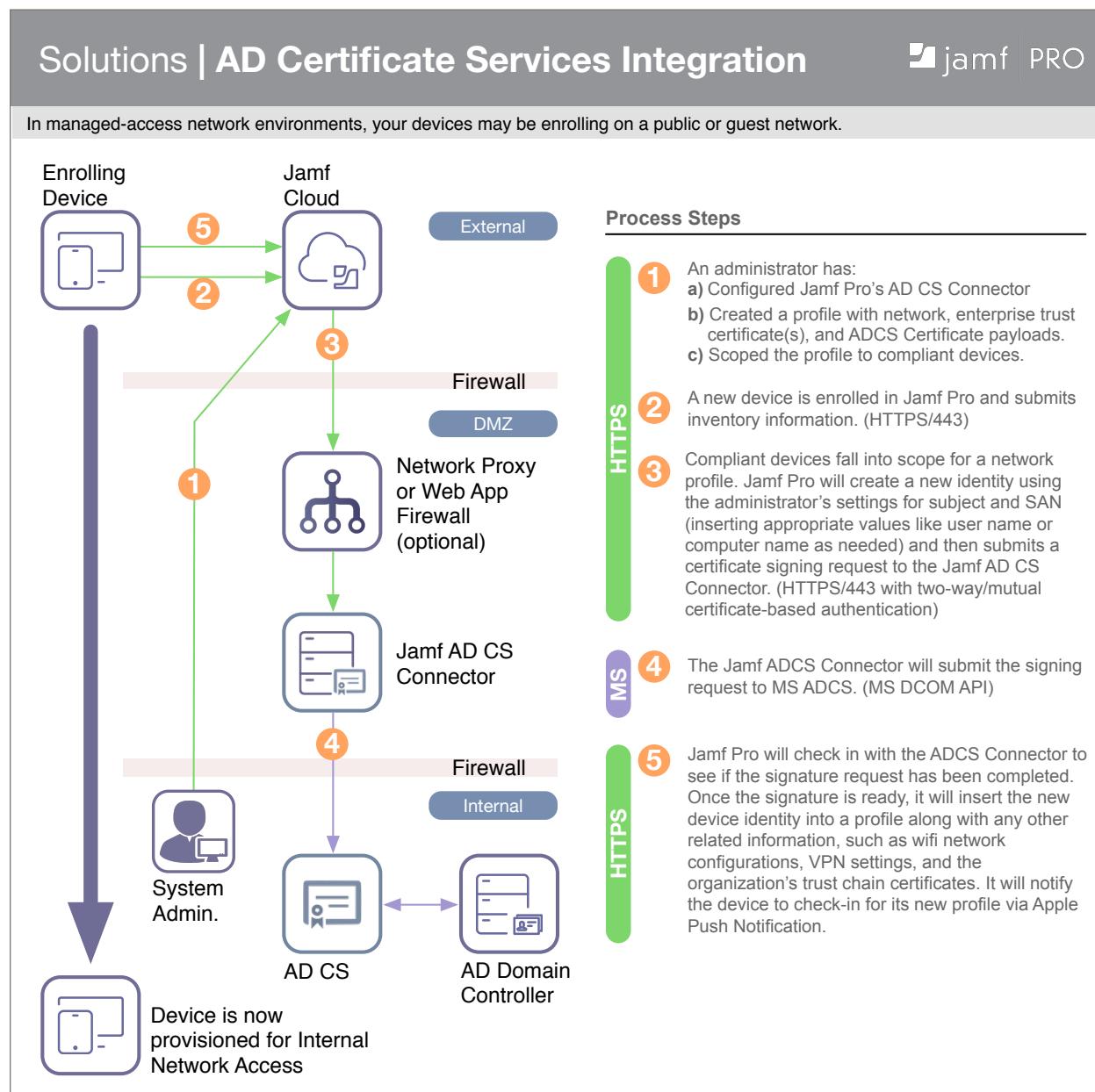
Option 3 allows the devices to use the Jamf Pro web application as a conduit for their transaction with a SCEP/Microsoft NDES service, and since the trust basis for all components is strong, this may be both acceptable and desirable, because in option 4, the Jamf Pro web application creates the private key, gets a signature from ADCS, and then delivers the completed identity to the device. In all other options, the private key is created by the managed device and never needs to leave it. The cost and level of effort to implement the SCEP/NDES approach is potentially lower because no additional server is required.

Some organizations choose ADCS Connector over SCEP Proxy because they prefer the former's certificate-based authentication, are opposed to using NDES/SCEP, or because they require the flexibility of using a variety of certificate templates. The remainder of this document deals with the Jamf Active Directory Services Connector and assumes that you have already determined that this option is the best fit for your organization.

ADCSC – Background

The Jamf Pro Active Directory Certificate Services Connector ("ADCS Connector" or "ADCSC") is an HTTP REST API running on a Microsoft IIS web server. It acts as an intermediary between Jamf Pro and Microsoft Active Directory Certificate Services ("ADCS"), submitting certificate requests to ADCS on behalf of Jamf Pro and returning completed signatures. Network communications, ports, protocols, and authentication are described in the following sections.

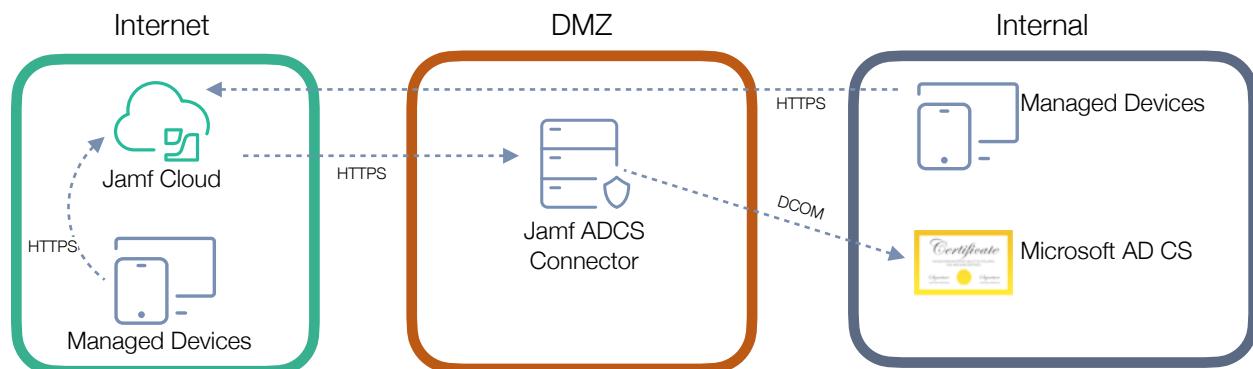
This diagram summarizes the managed device certificate deployment process using ADCS Connector.



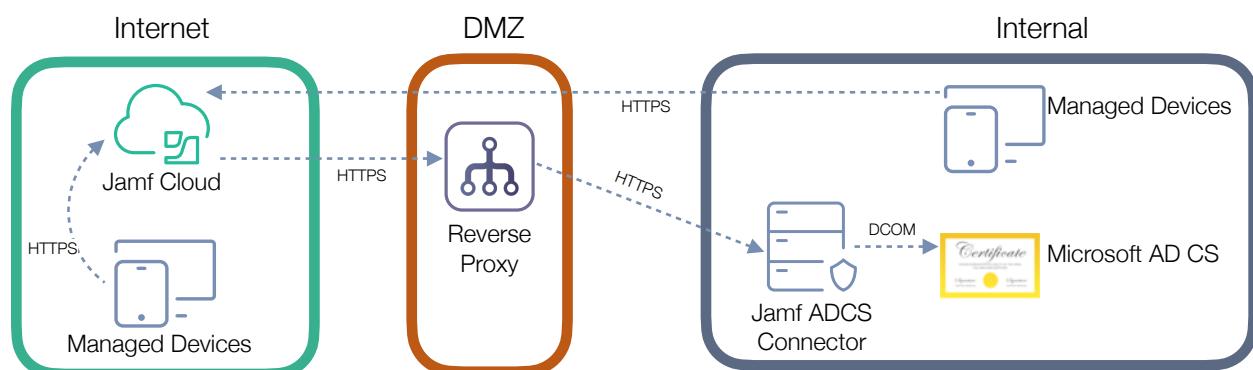
Network Connections

The following diagrams illustrate some common implementations of the Jamf ADCS Connector.

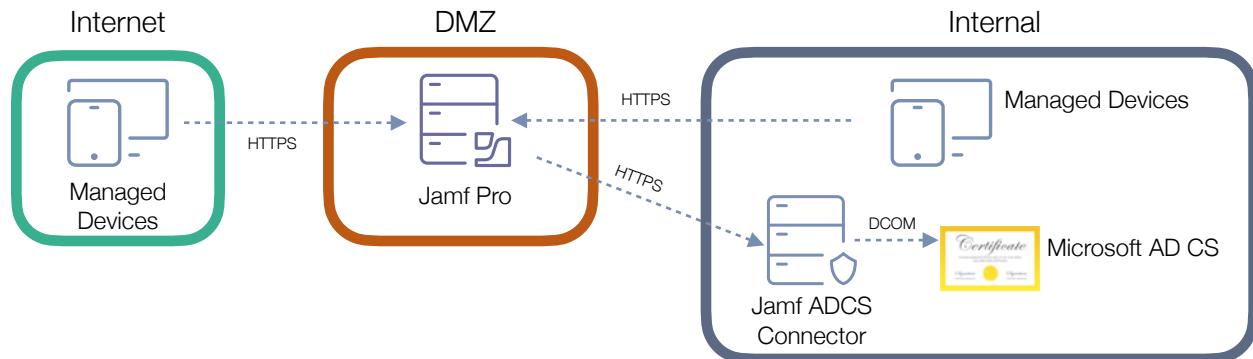
Jamf Cloud with Jamf ADCS Connector in the DMZ



Jamf Cloud with a DMZ Reverse Proxy Layer



Self-hosted Jamf Pro Server in the DMZ



Network Zones and Firewall Configuration

The ADCSC communications are encrypted and authenticated, but additional security is obtained by creating firewall rules in your network infrastructure and/or on the server/OS firewall.

The source IP addresses from which Jamf Cloud connections will originate are documented by <https://www.jamf.com/jamf-nation/articles/409/permitting-inbound-outbound-traffic-with-jamf-cloud>.

The host names, internal and external (VIP) IP addresses, or port numbers used in your internal networks can be configured as needed. Port 443 is commonly used for HTTPS connections. DCOM (Microsoft Distributed Component Object Model) connections between the ADCS Connector server and the ADCS server run on Microsoft's default ports (135 and 49152-65535), though those can be configured as well. Ref: [Default Ports Documentation](#). Please see the "Configuring ADCS to use an Alternate Port" section of this document for further considerations.

Connection	TCP Port (Typical)	Protocol	Description
Managed Devices to Jamf Pro	443	HTTPS	Apple OS-devices connect to their mobile device management ("MDM") server to receive management payloads.
Jamf Pro to Jamf AD CS Connector	443	HTTPS	Jamf Pro sends certificate signing requests and retrieves completed signatures by opening a connection to the Jamf AD CS Connector, typically on TCP port 443, but any available port can be used if preferred.
Jamf ADCS Connector to Microsoft ADCS	135: MS DCE endpoint resolution used by DCOM.	Microsoft DCOM	The Jamf AD CS Connector uses Microsoft Distributed Component Object Model (DCOM) to communicate with AD CS.
	49152-65535: Dynamic DCOM callback ports		

Security Mindset

Introduction

In many organizations, certificates are deployed to devices for use in verifying that devices or users which connect to internal networks are authorized to do so, and to allow network administrators to track who is connecting, when they connect, and which services they connect to. In other organizations, the certificate is used to authenticate to applications or for message signing. The security around the identity provisioning process must be considered in the context of the rights that identity confers and measured against the trust-basis of the provisioning process.

Regardless of the identity purpose, certain best-practices should be employed. These are discussed here.

ADCSC Communication Authentication and Trust Basis

Relationship	Authentication	Trust Basis
Devices to Jamf Pro	Message signing based on a device-specific MDM enrollment identity	This depends on the enrollment method, but for Automated Enrollment, 1) Device has been purchased by the organization and registered by Apple in Apple Business Manager or Apple School Manager, 2) An admin has accepted the device into a Jamf Pre-stage Enrollment Group, 3) The device user has authenticated with their organizational credentials on enrollment
Jamf Pro to ADCS Connector	Server and Client TLS certificate exchange	A Jamf Pro administrator will have uploaded the ADCS Connector server's public key and Jamf Pro ADCS Client Identity file (and its password) into the Jamf Pro console. Without these, no connection to ADCSC is possible.
ADCSC to ADCS	Microsoft Auth (Kerberos)	The ADCS administrator has granted permission to obtain certificates to the ADCS Connector host.

ADCSC and IT Service Security

No IT service can have perfect security. The goal of security practitioners should be to ensure that services are planned, implemented, and operated in a manner that minimizes risk. The security around the ADCS Connector is will be similar to the good security practices an organization uses with any of their network-based services.

- Administrators must have a strong understanding of the trust-basis, network connections, protocols, encryption, and authentication at each step of a communications flow.
- Vulnerabilities in the operating systems underlying an IT service are among the most frequently exploited attack vectors. Your installation should rely on the vendor's recommended practices and the vendor's updates should be applied diligently.
- Jamf patches related to security are uncommon, but action should be taken immediately in response to security notifications when the vulnerability effects a component or workflow that you are using. These will be sent to all customers via email and also posted prominently on

Jamf Nation. Jamf Cloud Standard customers are patched automatically. On-premise installations should be patched without undue delay.

- Strategies such as reverse proxies and firewalls can be used to insulate network components from attack. Proxies should be employed in a fashion that is consistent with your organization's standards and practices. Firewalls rules should allow only the minimum required connections between network zones and at the OS level.
- Security plan approval workflows, audits, or informal methods such as peer configuration review and testing can be used to verify that systems are implemented correctly.
- Anything downloaded or installed on a server should be sourced directly from a trusted vendor. For example, we would not install a network traffic monitoring utility downloaded from an untrusted repository onto a production server.
- Never use a web browser on a server to do anything unnecessary on a server. If you need to look up some information when troubleshooting, do it from your user machine.
- Strong measures should be taken to protect credentials such as private keys and service-account user names and passwords in transit and at rest. For example, we would not send user account information/passwords or a .pfx keystore and its password together in an email or copy them to a local user machine. The chain of custody of private keys should be carefully protected.
- Remote Desktop and ssh access to a server (and any management servers) should be limited only to trusted and required persons.
- Practices such as key/password rotation may be used to limit the amount of time that exposed credentials may be used to penetrate a system. Two-factor authentication ensures that exposure of a single factor (i.e. username/password) is not sufficient to gain access.
- Avoid the use of local administrator accounts on Windows servers. Domain accounts with password complexity, lockout rules, and expiration are preferred.
- Use certificate-based authentication for ssh, not username and password.

These are meant to illustrate general principles of server operation. More specific actions are generally available from network, server, monitoring, and OS vendors. These should be employed as they are with other IT services hosted by your organization.

QA Your Firewall Rules!

Once testing is complete and production rules are in place, only the Jamf Cloud IP addresses should be able to reach the connector's external host name. Use a web browser or a command line tool like netcat/openssl s_client/telnet to make sure you cannot reach the external hostname and port from the public internet, DMZ, or internal networks. Also verify that you cannot reach the Connector's internal/local host name from any of these network zones.

Private Key Chain of Custody / Data at Rest and In Transit

Introduction

The encryption, authentication, and trust basis for each step of the ADCS identity provisioning process have already been discussed.

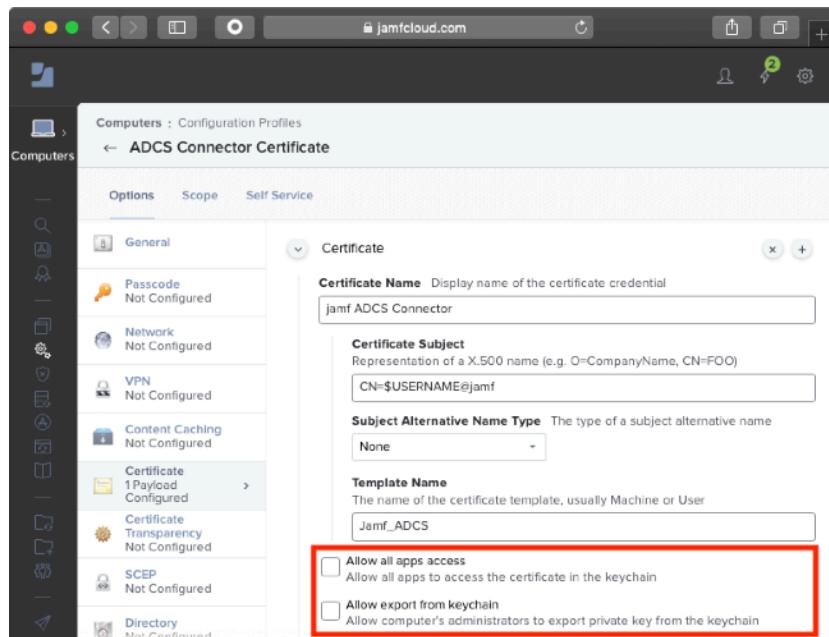
We have also previously noted that when using the Microsoft and SCEP (or SCEP Proxy) MDM payloads, the private key is generated on the managed device and never leaves. The device uses the private key to generate the CSR. With the ADCS Connector (Certificates) payload, the private key and CSR are generated by the Jamf Pro web application server. The private key is stored (AES-256 encrypted) in the Jamf Pro database to support certificate profile delivery and certificate lifecycle management. The application does not provide any facility for export or extraction of the private keys other than delivery (via MDM profile) to the intended managed device.

All HTTP/REST communications occur over TLS. Jamf Pro installers do not enable support for SSL v3.0.

Customers sometimes ask how an identity delivered via the certificates profile is secured on the device with respect to data at rest and exportability of the private key. A key point to recognize here is that in the final step where we actual deliver the identity to the managed device, we are using an Apple-standard certificate payload. How these payloads are implemented is determined by Apple.

Once the profile with a certificate payload is received by a device, the OS will pass the identity into a local keychain file. You should refer to Apple documentation for further information. For example, <https://support.apple.com/guide/security/keychain-data-protection-overview-secb0694df1a/web>.

As for the exportability question, note that both the Certificate and SCEP payloads in Jamf Pro have access and exportability flags. These settings will be included in the profile we deliver to the device and respected by Apple's OSs.



Installation of ADCS Connector – Requirements Summary

Software Installation and Network Requirements

The following preparations should be made prior to installation:

- A Windows OS with .NET Framework 4.5 or greater (E.g. Windows Server 2016/2019) joined to a domain that has a trust basis with the ADCS domain.
- Port 443 open inbound from Jamf Pro to the ADCS Connector host.
- DCOM (Microsoft Distributed Component Object Model) permitted between the ADCS Connector host and the ADCS server. Ports 135 and 49152-65535 are the MS defaults.
- The DNS used by Jamf Pro can resolve the FQDN of your ADCS Connector. E.g., if you are on Jamf Cloud, The FQDN of the Connector's external VIP is available in public DNS. If your organization has a custom Jamf Cloud environment, you may request host file entries

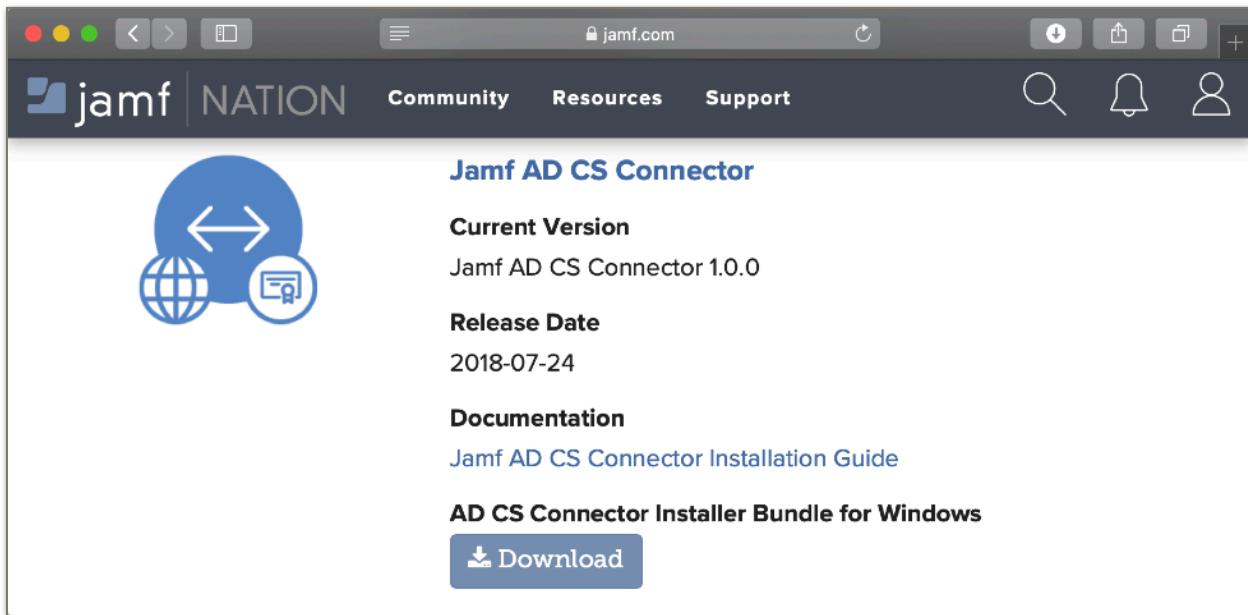
Microsoft DCOM Binding Requirement

The standard installer will configure a thread pool to run the ADCS Connector in IIS. By default, IIS runs applications as "AppPoolIdentity". When using this identity, IIS will connect to ADCS using the ADCSC's host computer's system account, and this computer account will need to be given permissions on the CA, and when using an Enterprise CA, also to one or more certificate templates. For this reason, the server running the Connector host must be bound either to the same domain as ADCS, or to a forest domain that has a trust relationship with the ADCS domain. The Connector can also be configured to run as another user, such as a domain service account. If the Connector host is in a network zone (like a DMZ) where it cannot bind to the AD domain, a local account with the same username and password as a domain account with permission on the CA and Certificate template. This will be discussed later in the document.

Installing the Jamf ADCS Connector

1. Download a copy of the installer

If you are a Jamf customer, the installation software is available under the "My Assets" section once you have logged into Jamf Nation.



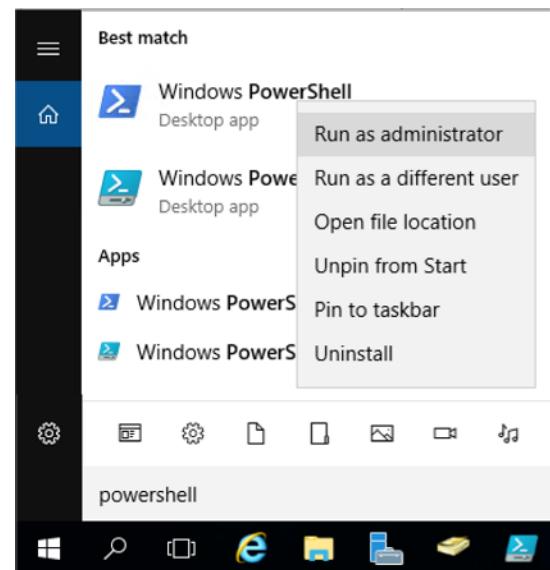
Copy the installer to the computer that will host the ADCS Connector and decompress the .zip archive.

2. Run the installer

The installer includes a Powershell script that can be called from the Windows command line or Powershell command line to unzip the ADCS Connector files and setup/configure Microsoft IIS.

Run the Windows PowerShell command line as administrator, "cd" into the folder that contains the deploy.ps1 script, and run the installer. For the -fqdn parameter, specify the host name Jamf Pro will resolve to connect to the Connector. Note that this is the external/VIP hostname, not necessarily the same as the actual host that runs the Connector. For -jamfdn, use your Jamf Pro host name. See the next page for an example.

The documentation is available from <http://docs.jamf.com/ad-cs-connector/1.0.0/index.html>.



The operation will be similar to the following:

```
PS > cd "C:\Users\admin\Desktop\ADCS Connector"
PS C:\Users\admin\Desktop\ADCS Connector> .\deploy.ps1 -fqdn adcsc.my.org
-jamfProDn my.jamfcloud.com

Enabling IIS and ASP.NET features...
IIS and ASP.NET enabled.
Removing AdcsProxyPool Application Pool...
Removing AdcsProxy Site...
Install path C:\inetpub\wwwroot\adcsproxy already exists.
Cleaning C:\inetpub\wwwroot\adcsproxy...
Unzipping site to C:\inetpub\wwwroot\adcsproxy...
Creating AdcsProxyPool Application Pool...
Creating site AdcsProxy...
Creating local user account AdcsProxyAccessUser. This user will be referenced
for IIS Client Certificate Mapping Authentication.

Created new local user AdcsProxyAccessUser with password ^\::X"+Y#bb8Wh?rC8lh

!!!NOTE - Please save this information if setting up IIS Client Certificate
Mapping Authentication manually.

Adding Windows Firewall rule to allow inbound TCP traffic on port 443
Configuring HTTPS...
Generating self-signed certificate for ms.jamf.com...
Adding adcsc.my.org to local root CA store...
Generating adcsc.my.org-signed certificate for j...
Configuring IIS Client Certificate Mapping Authentication for
AdcsProxyAccessUser...
Exporting client certificate keystore...
Client keystore exported.

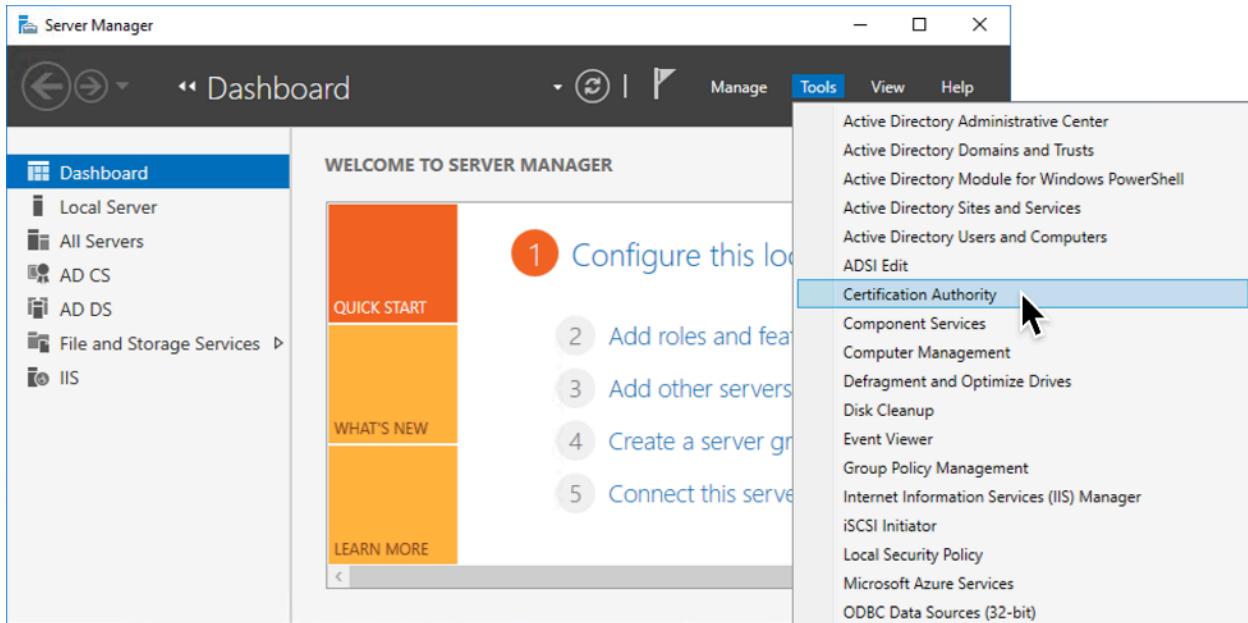
!!!NOTE - Client cert keystore password: c2sG5J5orHM3ZLP
```

Make note of the **Client cert keystore password**. You'll be prompted to enter this password when importing the client identity file ("client-cert.pfx") to Jamf Pro. If you close the Powershell window before noting the password, you'll need to re-run the installer to get a new identity generated.

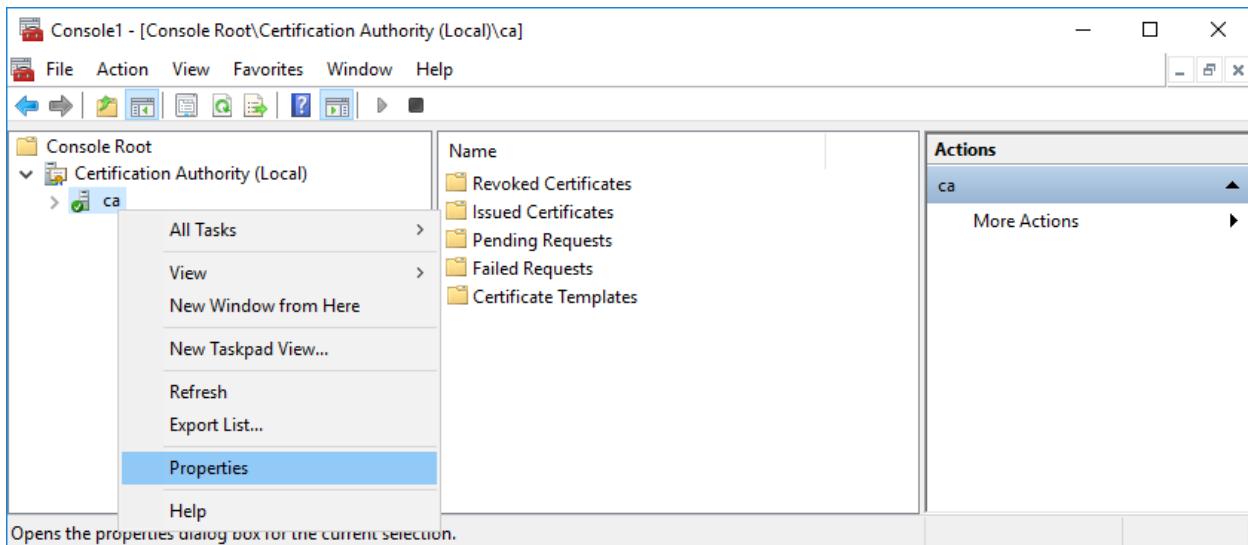
3. Give ADCS Connector permission to talk to the CA

The Connector is ready to accept certificate requests and pass them on to ADCS, but if you do it now, you'll get a "CR_DISP_DENIED" error because we haven't yet given the Connector any permissions in ADCS.

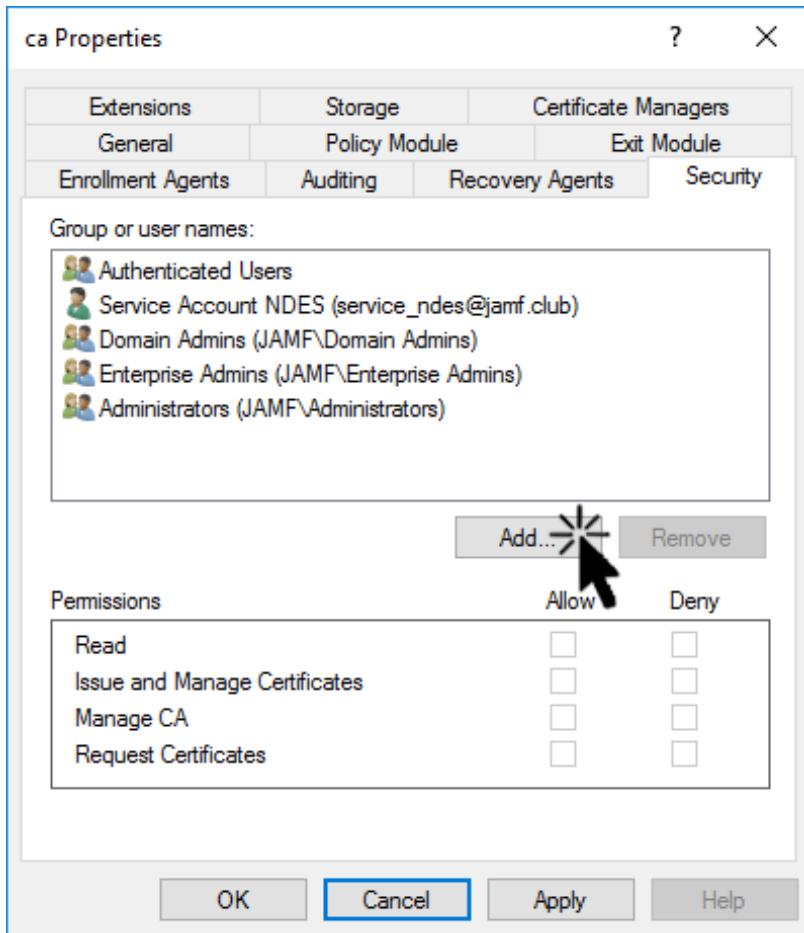
Run certsrv via cmd or as an MMC snap in. Or just select "Certification Authority" from Server Manager's Tools menu.



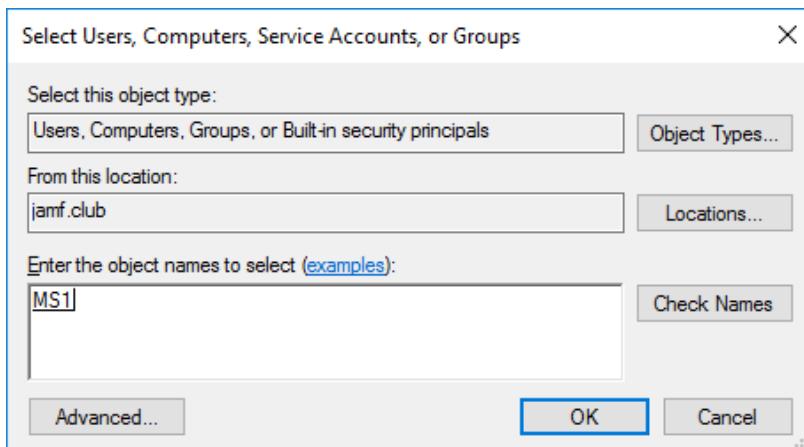
Right click on your CA's name and select "Properties..." .



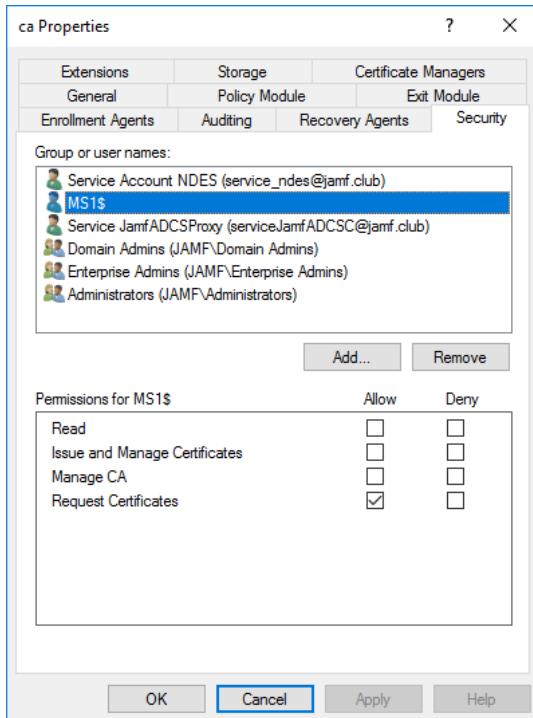
Switch to the "Security" tab and click the "Add..." button.



Make sure "Computers" is in the list of selectable object types. If it's not, use the "Object Types..." button to add it. Then add the server that's running ADCS Connector and click "OK".



You will not see the Connector Host entry in the Group and User names list. Grant the "Request Certificates" permission. It also needs "Read" permissions, but in a default ADCS setup, that permission is inherited via "Authenticated Users" so you don't need to check it here.



[Creating a Certificate Template in ADCS and Granting Template Permissions](#)

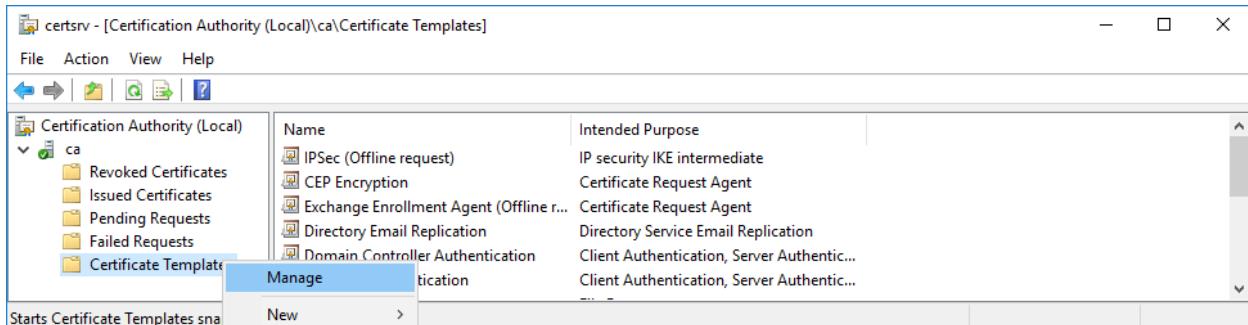
If you're running a stand-alone CA, you're done configuring permissions. If you use an Enterprise CA, you will also need to configure a certificate template.

Warning: Templates are usually set up to derive the certificate subject from the authenticating user making the request. But since we're setting up a proxy, we're going to create one that allows the requester to specify an arbitrary subject. Follow the upcoming instructions carefully so that only the Jamf ADCS Connector can obtain certificate signatures using this template.

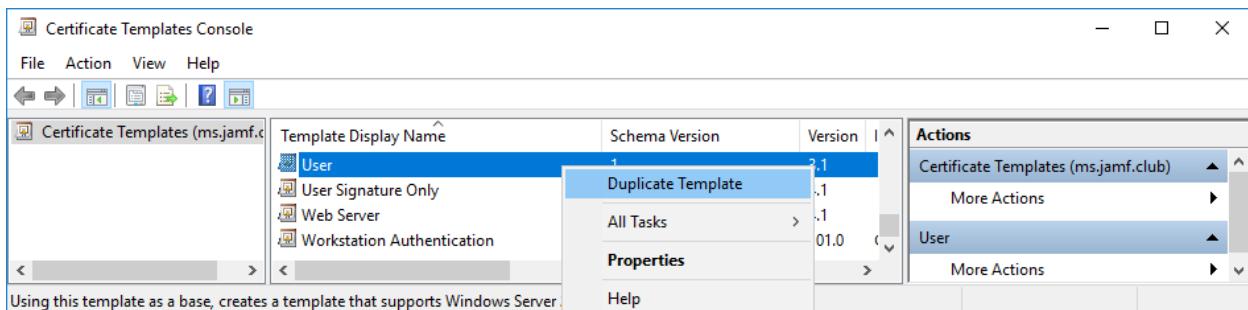
Your existing CA probably already has a template that's being used to deploy certificates to bound computers and servers. But if you used that with the ADCS Connector, every certificate would have the name of the Connector host as its subject, not the device or username for which we're actually trying to provision the certificate. We need to duplicate that existing template and set the new version up so we can specify the subject when we make certificate requests.

Using a separate template dedicated to use by the Jamf ADCS Proxy will also help the CA admin to pick out the Apple templates when they look at the Issued/Failed lists in their ADCS console.

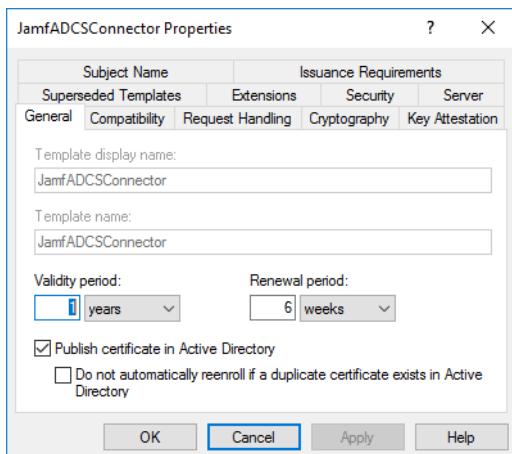
Certificate templates are managed using the Certificate Templates Console. In certsrv, right-click on "Certificate Templates" and select "Manage" from the contextual menu to run the template console. You can also access Certificate Templates Console as mmc snap-in or by running certtmpl.msc directly.



You could create a new template from scratch, but it's usually easier to duplicate an existing known-good template -- typically the same one you are already using successfully to provision certificates for domain-bound devices. Right-click the source template to duplicate it.



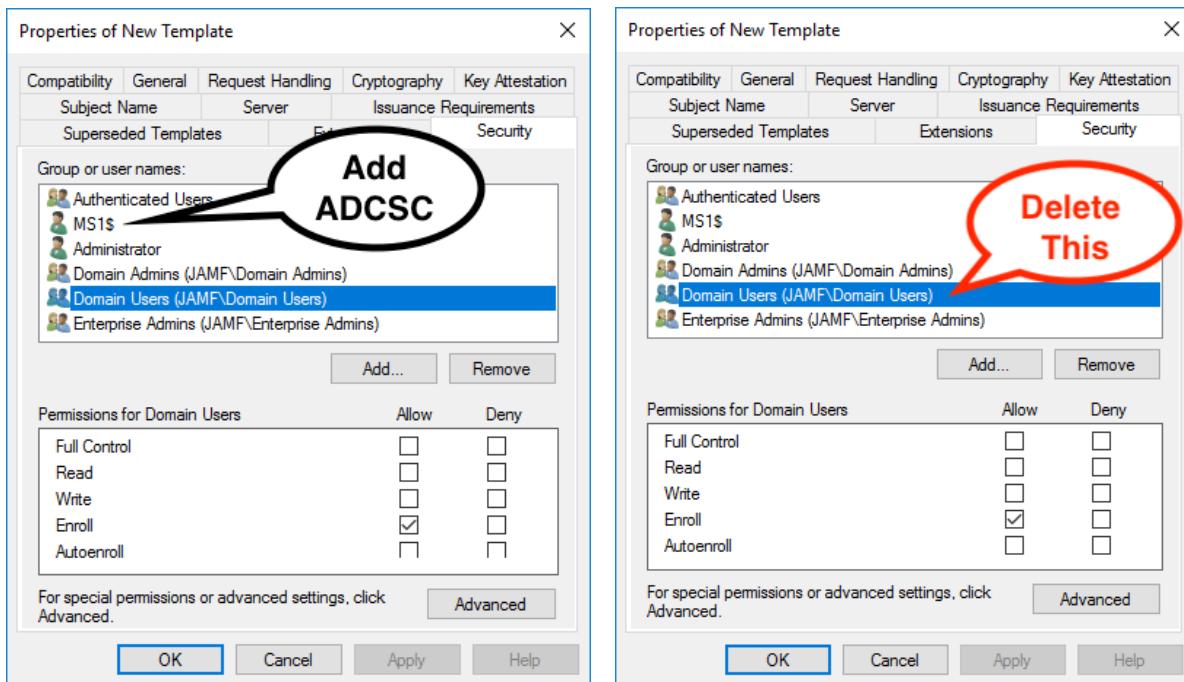
A certificate properties editor dialog will appear. In the "General" tab, give it a name consistent with whatever naming standard your CA admin prefers. Make a careful note of the name or paste it into an email/document. You'll need to enter this exact name when you're configuring Jamf Pro to work with the Connector. Note that it's the **Template Name** we need to specify, not the display name. They could be different.



In the "Security" tab, add the ADCS Connector host, just as you did at the CA-level and grant the "Enroll" permission.

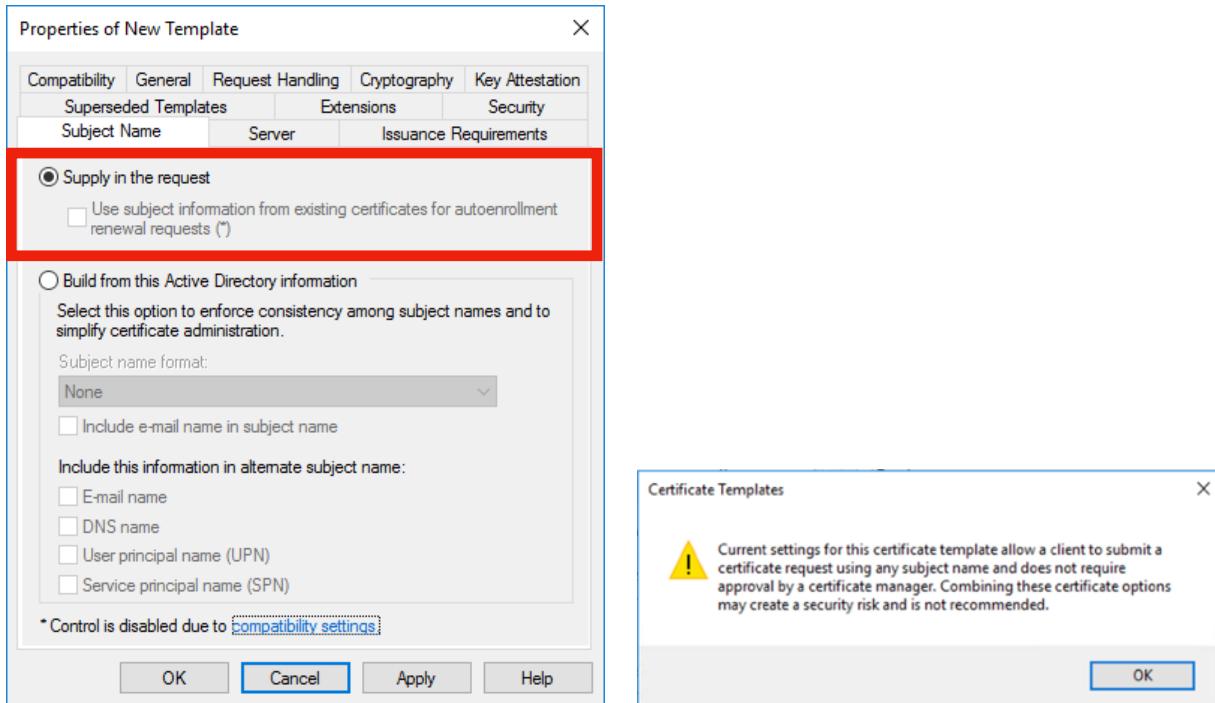
Next, select "Authenticated Users" and verify that it has Read permission and does not have the enroll permission. Some organizations will also remove "Authenticated Users" in favor of explicitly allowing each individual CA host. If that's the case, make sure the ADCS Connector host and all the subordinate CA boxes have read permission on the template.

If "Domain Users" or "Domain Computers" are in the list, delete them. It wouldn't hurt if they were there and had only read access, but you don't want to risk making a mistake and giving them enroll or auto-enroll. If you did, any user could use this template to create any certificate subject they wanted, even a wildcard for your domain!



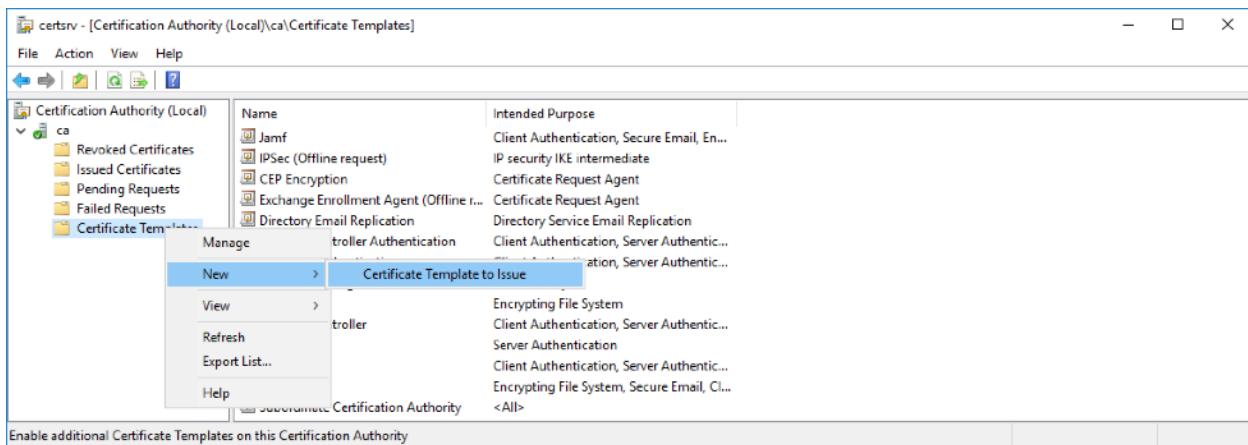
<https://itpro.outsidesys.com/2018/03/21/adcs-manage-pki-certificate-templates/> has some good discussion of this topic if you want to understand it better.

In the "Subject Name" tab, allow the request to be supplied in the request. You will see a warning when you make this change. ADCS shows this for the same reason we warned you to prevent everyone other than the ADCS Connector to have enroll permissions on the template.

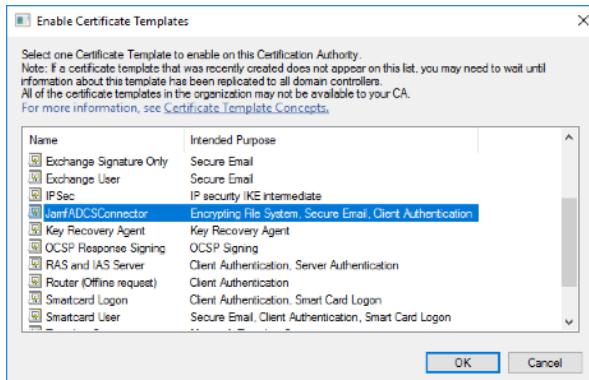


Click "OK" to exit the setup.

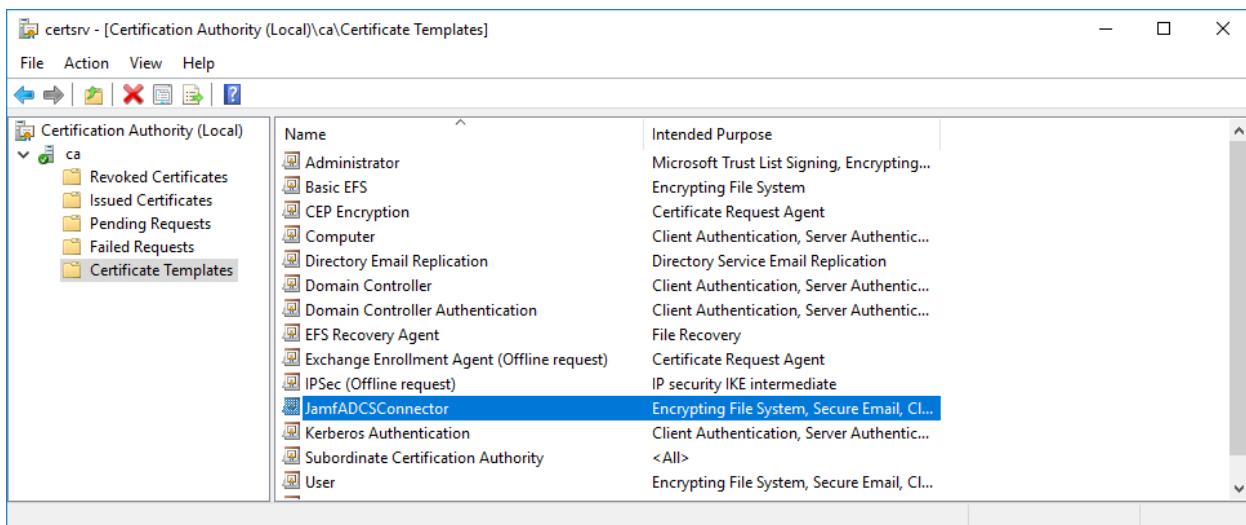
Now that we have created the certificate template, we need to tell the CA that it's available. Go back to the certsrv Certificate Authority console and right-click on "Certificate Templates" and select "New>Certificate Templates to Issue".



Select the template you created for the Connector and click "OK".



The template is now added to the CA's list of issuing templates.



Artifacts

At the completion of the installation process, you will have the following ready for configuration in Jamf Pro:

- 1) The adcs-proxy-ca.cer file, the public key of the Connectors TLS Server certificate
- 2) The client-cert.pfx file, the keypair Jamf Pro will use to authenticate to the Connector
- 3) The password needed to unlock the client-cert.pfx file.
- 4) The template name. (Again... *not the display name*, unless they happen to be the same.)

Next Steps

The following pages show how to see the changes that were made in the IIS configuration after running the installer, followed by a section that explains how to configure Jamf Pro to talk to the newly-installed Connector server.

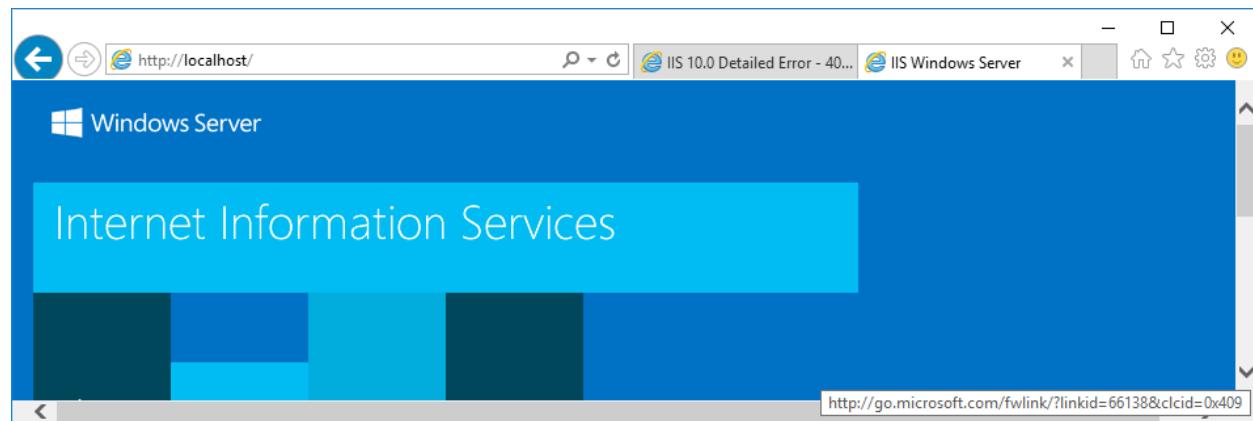
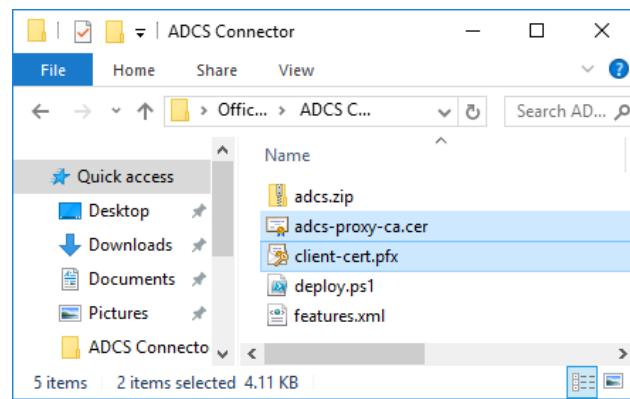
Resulting Configurations

You now have everything you need to configure the ADCS Connector in Jamf Pro. In the working directory from which you ran the script, you now have two new files.

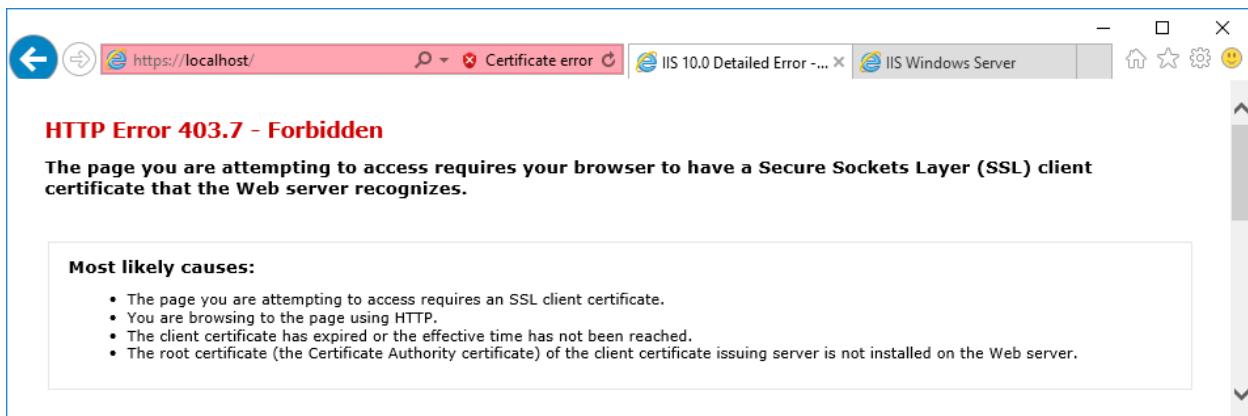
adcs-proxy-ca.cer is the public key for the identity that IIS will use when negotiating TLS when Jamf Pro tries to connect. If the server uses an identity that doesn't match up with this public key, Jamf Pro will not trust the server and the TLS handshake will fail.

Even more importantly, the ADCS Connector needs to know that the connecting client is authorized. The client-cert.pfx file is the keypair that Jamf Pro will need to present in order to successfully authenticate to IIS and reach the Connector application. This file is protected by a random password, which is shown at the end of the deploy.ps1 script's output (shown with orange highlight in the example run we showed earlier).

If we launch a web browser on the Connector host, we can see that IIS has been installed.



If we attempt to browse to the Connector and accept the self-signed server certificate warning, we will get an authentication error, showing that anonymous auth is disabled.



In IIS Manager, we observe that a new application pool has been created for the ADCS Connector. The ADCS Connector site will run within this pool. We see that the Connector is running as "ApplicationPoolIdentity", an identity derived from the local computer's bind to the domain. This is why we gave the computer certificate enrollment permissions on the CA and the template we created.

Name	Status	.NET CLR V...	Managed Pipel...	Identity
.NET v4.5	Started	v4.0	Integrated	ApplicationPoolId...
.NET v4.5 Classic	Started	v4.0	Classic	ApplicationPoolId...
AdcsProxyPool	Started	v4.0	Integrated	ApplicationPoolId...
DefaultAppPool	Started	v4.0	Integrated	ApplicationPoolId...
SCEP	Started	v4.0	Classic	JAMF\service_ndes

Under Sites, we'll see the corresponding site. It's listening for https on port 443.

The screenshot shows the IIS Manager interface. In the left sidebar under 'Connections', 'Sites' is selected. The main pane displays a table titled 'Sites' with columns: Name, ID, Status, Binding, and Path. Two sites are listed:

Name	ID	Status	Binding	Path
AdcsProxy	3	Started (http)	*:443 (https)	C:\inetpub\wwwroot\adcsproxy
Default Web Site	1	Started (http)	*:80 (http)	%SystemDrive%\inetpub\wwwroot

The 'Actions' pane on the right includes options like 'Edit Site', 'Bindings...', 'Basic Settings...', 'Explore', 'Edit Permissions...', 'Remove', 'Rename', 'View Applications', 'View Virtual Directories', 'Manage Website', 'Restart', 'Start', and 'Stop'. The status bar at the bottom indicates 'Ready'.

In bindings, we see that the TLS certificate subject matches the FQDN we will tell Jamf to resolve when it connects.

The screenshot shows the IIS Manager interface with the 'Site Bindings' dialog open for the 'AdcsProxy' site. The 'Actions' pane on the right includes options like 'Explore', 'Edit Permissions...', 'Edit Site', 'Bindings...', 'Basic Settings...', 'View Applications', 'View Virtual Directories', 'Manage Website', 'Restart', 'Start', and 'Stop'. The 'Edit Site Binding' dialog is centered, showing the configuration for the https binding:

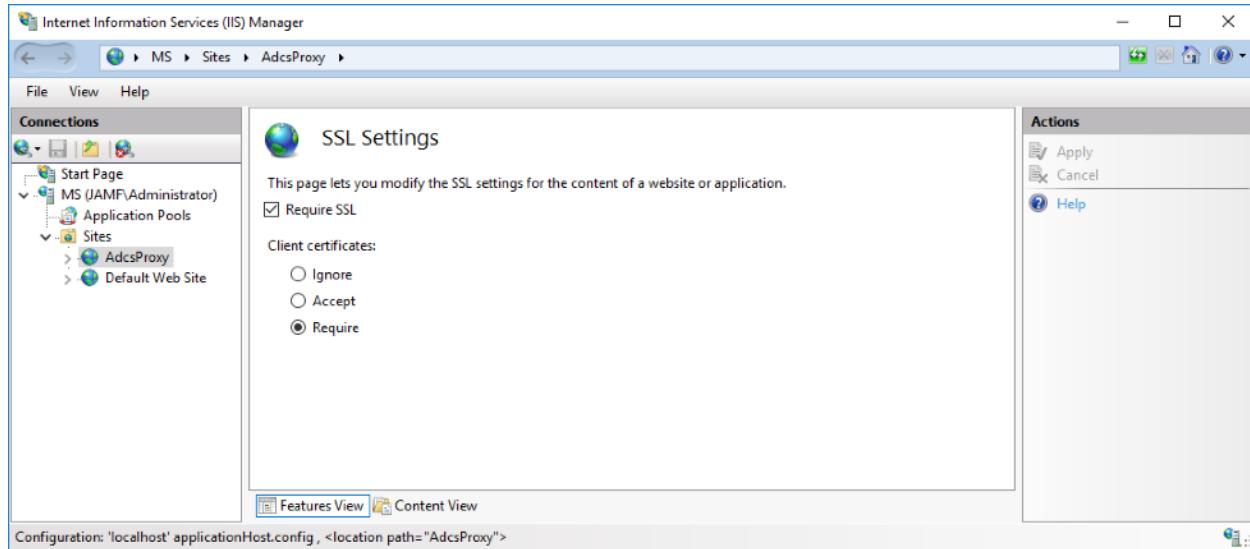
Type	Host Name	Port	IP Address	Binding Information
https	*	443	All Unassigned	

The 'Edit Site Binding' dialog contains the following fields:

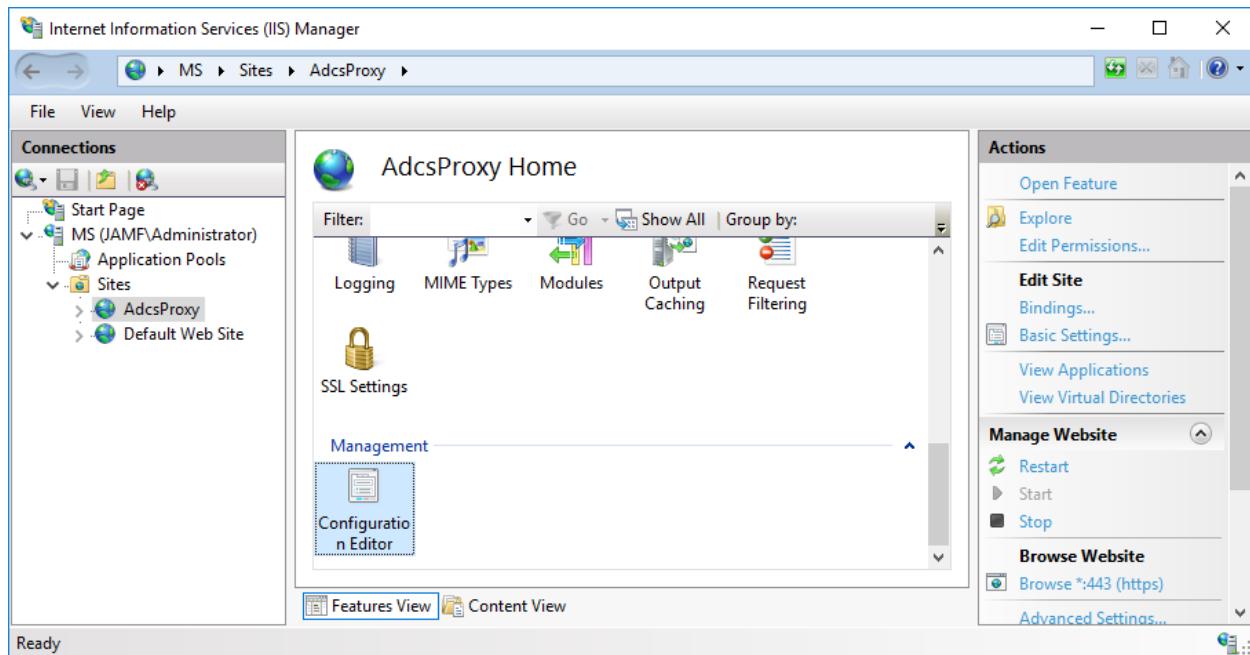
- Type: https
- IP address: All Unassigned
- Port: 443
- Host name: (empty text box)
- Require Server Name Indication
- SSL certificate: ms.jamf.club

The 'Actions' pane on the right includes options like 'Explore', 'Edit Permissions...', 'Edit Site', 'Bindings...', 'Basic Settings...', 'View Applications', 'View Virtual Directories', 'Manage Website', 'Restart', 'Start', and 'Stop'. The status bar at the bottom indicates 'Ready'.

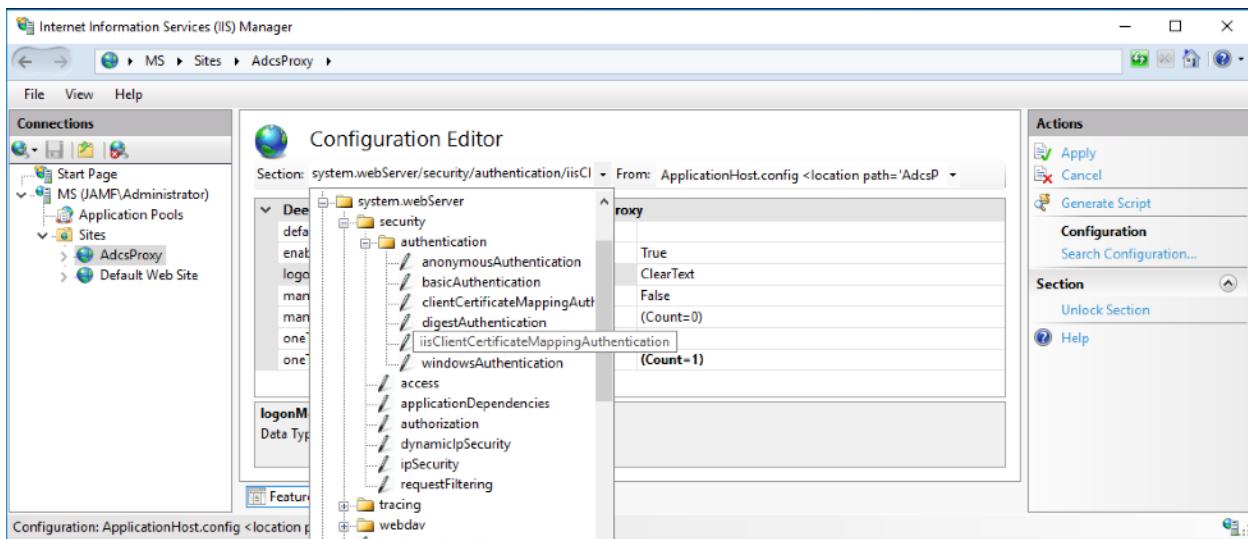
Under SSL Settings, we see that IIS will require SSL connections and that connecting computers present a client certificate for authentication.



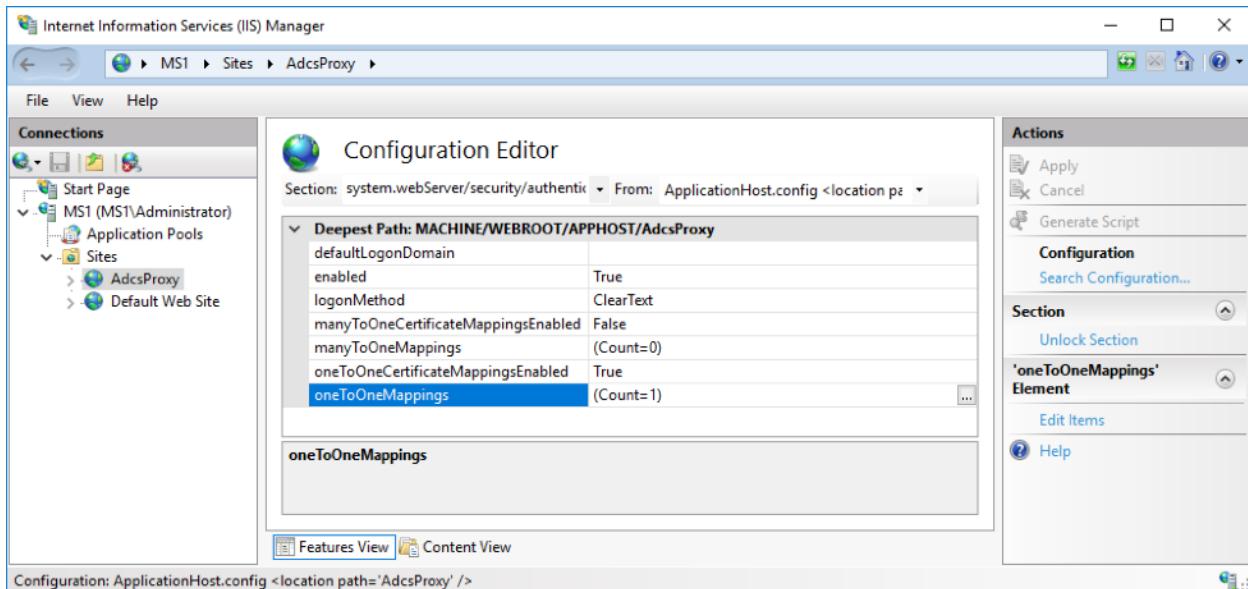
To review client certificate authentication settings, go to Configuration Editor.



Navigate to "system.webServer > security > authentication > iisClientCertificateMappingAauthentication" in the "Section" drop-down menu.



Highlight oneToOneMappings and click the "..." Button to the right of the configuration entry.



The editor will display the settings for client configuration. The certificate value is the base-64 public key for the client identity. It is used to ensure a valid identity is being used to negotiate TLS. The username and password indicate the user that will be authenticated to IIS when a valid certificate is presented.

Collection Editor - system.webServer/security/authentication/iisClientCertificateMappingAuthentication/oneToOneMappings/

Items:			
enabled	userName	password	certificate
True	AdcsProxyAccessUser	*****	MIIDLTCCAhWgAwIBAgIQVzzdY0jZsoBBKakAr/SfZzANBgkqhkiG9w0BAQsFADAYMRYwFAYDVQQDA1tcEuamFtZi5jbHViMB4XDTE5MTAz
< >			
Properties:			
certificate	MIIDLTCCAhWgAwIBAgIQVzzdY0jZsoBBKakAr/SfZzANBgkqhkiG9w0BAQsFADAYMRYwFAYDVQQDA1tcEuamFtZi5jbHViMB4XDTE5MTAz		
enabled	True		
password	*****		
userName	AdcsProxyAccessUser		
certificate Data Type:string			

Actions:

- Collection
 - Add
 - Clear All
- Item Properties
 - Lock Item
 - Remove
 - Help
 - Online Help

File Location Notes:

IIS Configuration Settings: c:\Windows\System32\inetsrv\config\applicationHost.config

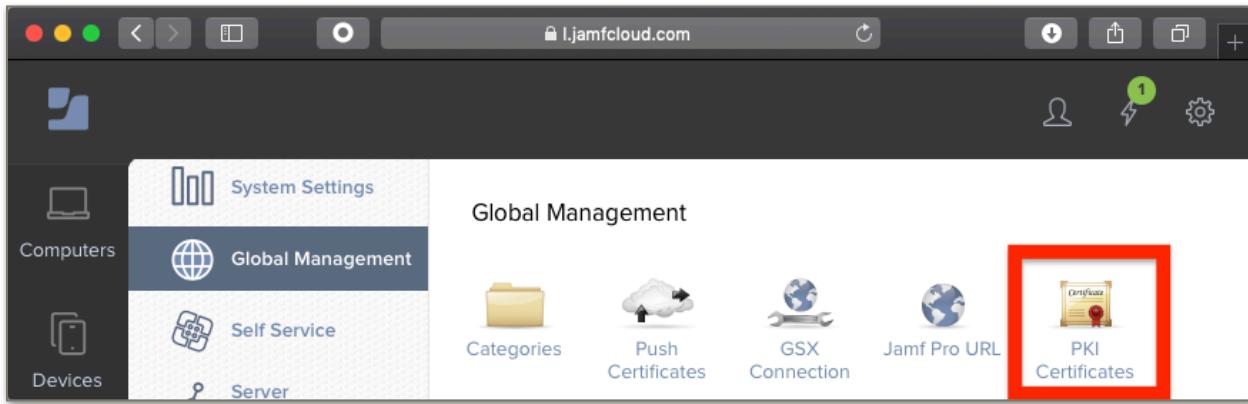
IIS Connection Logs: c:\inetpub\logs\LogFiles

Note: You'll see multiple folders here, one for each site ID. You can get ADCSC's IIS site ID from the site list in IIS Manager.

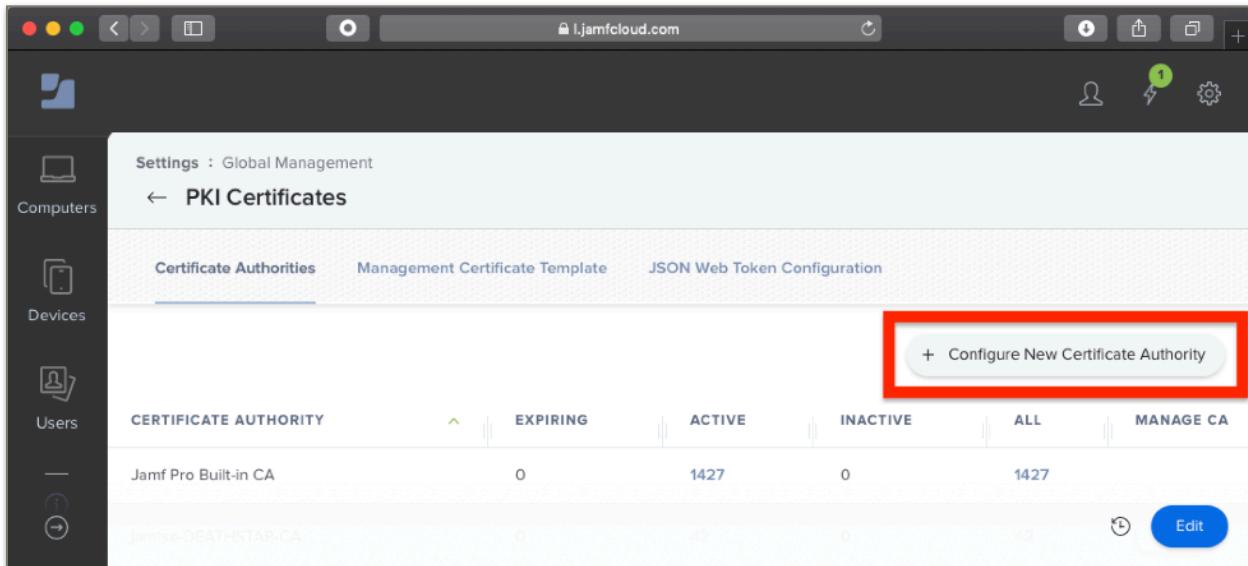
Jamf Pro Configuration

After installing the ADCS Connector Software, we will configure Jamf Pro to use it when obtaining certificates. Complete instructions for Jamf Pro Configuration are available in the Jamf Pro product documentation but we will summarize them here.

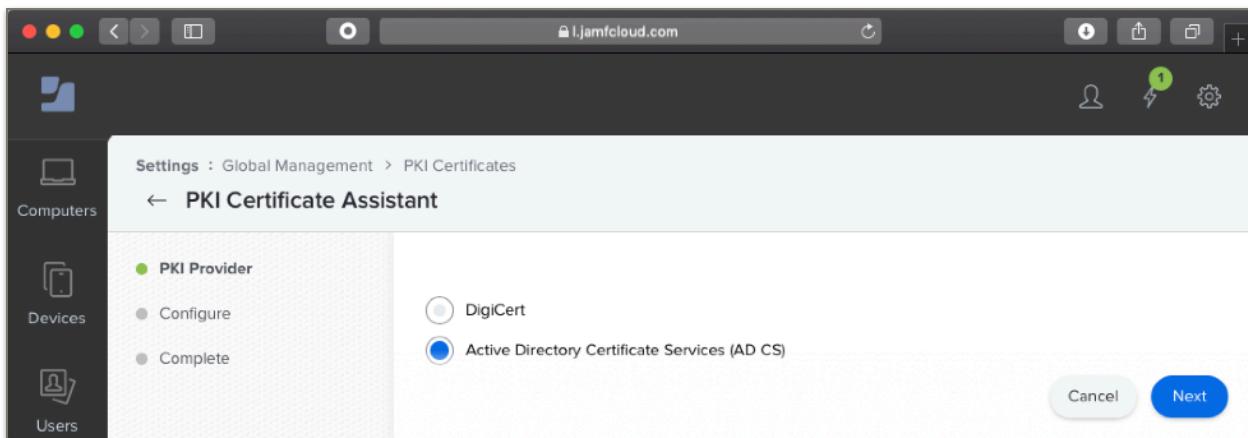
1. Login to Jamf Pro
2. Click the gearbox in the upper right corner to access the settings page
3. Go to Global Settings
4. Click PKI Certificates



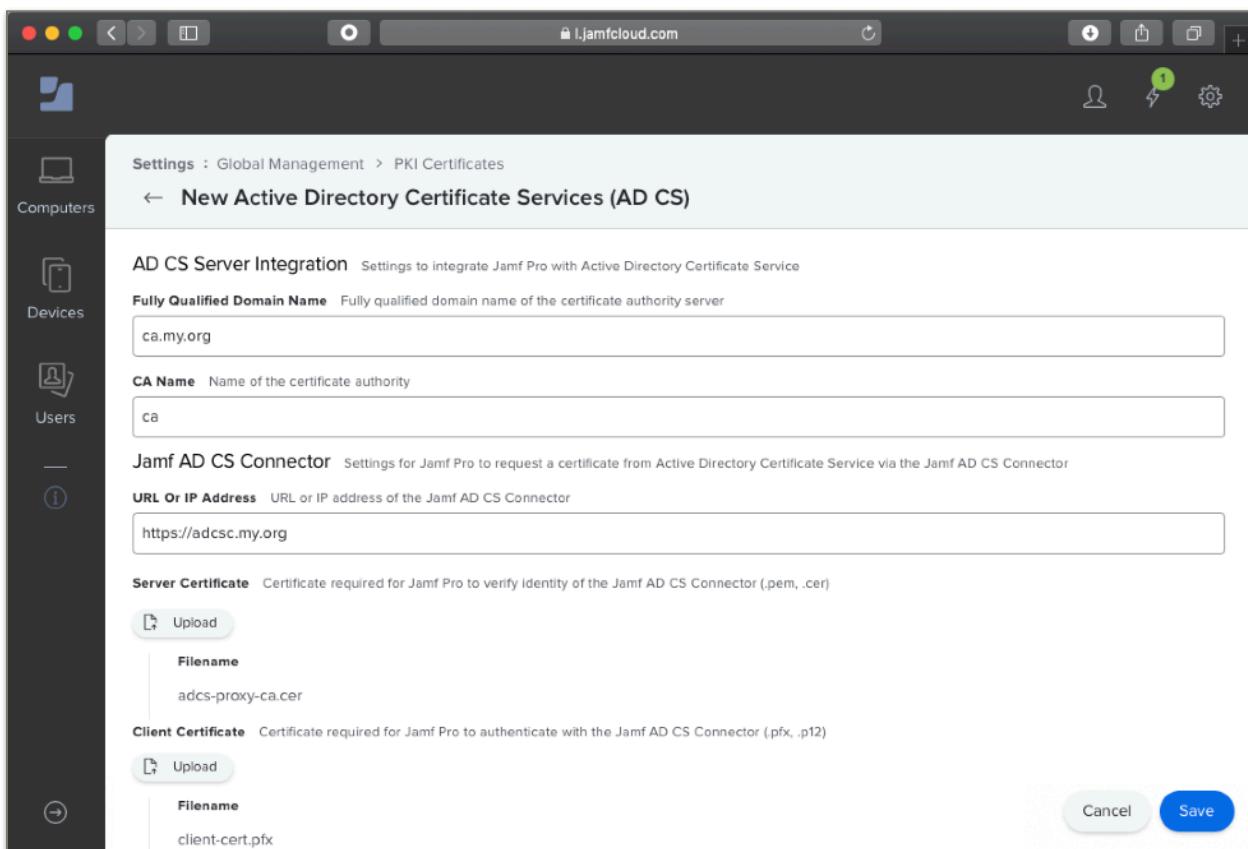
Click "Configure New Certification Authority".



Select the "Active Directory Certificate Services (AD CS)" option.



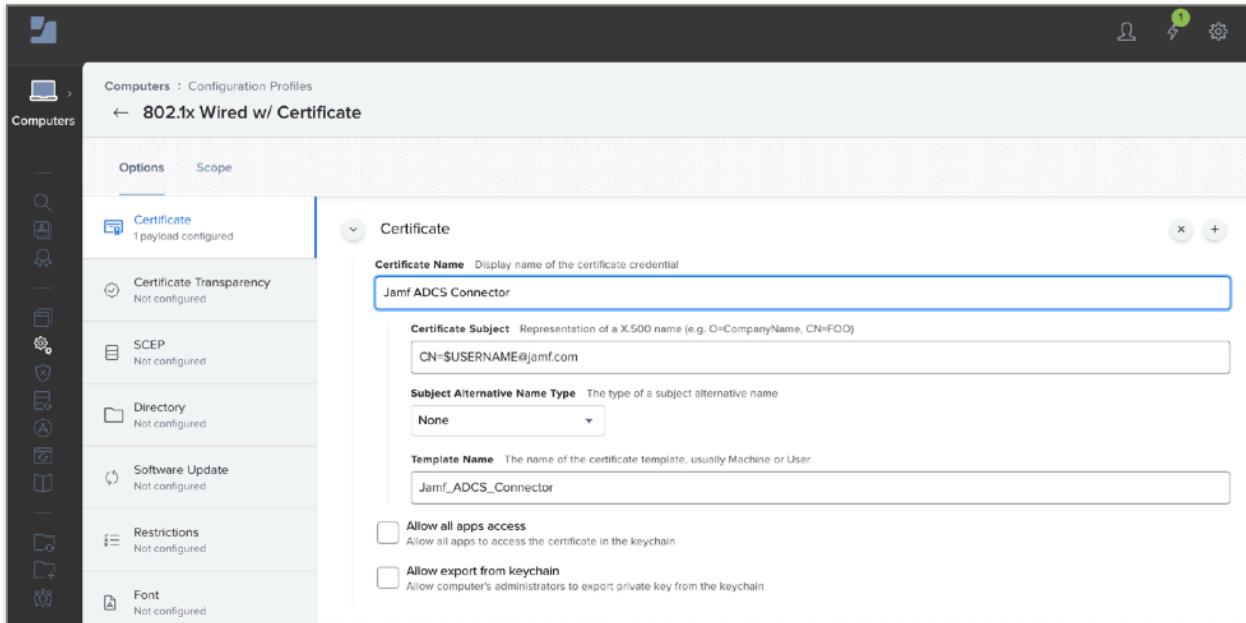
Enter the hostname of the AD CS server (what the Jamf AD CS Connector will talk to to request a certificate), the name of the CA instance, and the URL of the ADCS Connector (what Jamf Pro will talk to to request a certificate.) Upload the server certificate (how Jamf Pro will verify the identity of the Jamf AD CD Connector) and the client certificate (what Jamf Pro will present to the Connector to verify that it is authorized to request certificates).



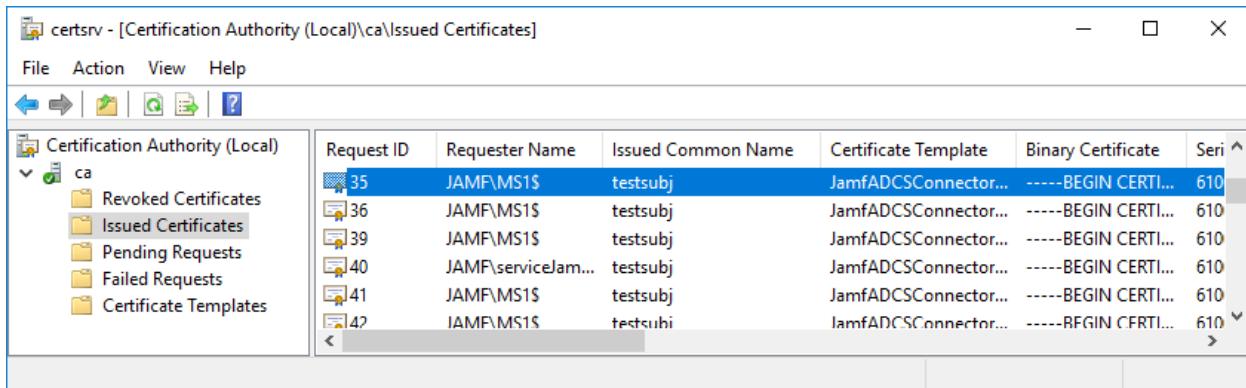
Click the Save button when finished. You can now create certificate profiles to deploy to your devices and Jamf Pro can obtain them via the AD CS Connector proxy service.

Example Certificate Profile and Issued Certificate

Once you've configured the AD CS Connector in Jamf Pro PKI settings, you'll have the option to provision certificates using the certificates payload under Computer and Mobile Device Profiles.



As devices obtain their certificates, they'll start showing up in your ADCS console. Note that the "Requester" is the Jamf ADCS Connector host and the subject will conform to the format you specified in your certificates profile payload.



Introduction to ADCS Connector Customizations

Common questions about the implementation of ADCS Connector include:

- Can we adjust things like the port used in IIS or the expiration date on the identities the installer script generates?
- Can it run in a load-balanced configuration to support high availability?
- Can it run behind a reverse proxy or web application firewall to insulate it from other network zones?
- Can we use our own server and/or client TLS identities?

We will discuss these customization options in the sections that follow.

Installation Script Customization

The installation script is written in PowerShell. Many Windows admins will already be familiar with this scripting language. The parameters section at the top of the script identifies available configuration options such as port, host names, etc. These are mainly used when we install the Connector on an existing IIS server already running other applications or sites.

```
param (
    [switch]$help = $false,
    [string]$archivePath = ".\adcs.zip",
    [string]$installPath = "C:\inetpub\wwwroot\adcsproxy",
    [string]$hostPath = "",
    [int]$bindPort = 443,
    [switch]$installIIS = $true,
    [switch]$cleanInstall = $true,
    [string]$appPool = "AdcsProxyPool",
    [string]$siteName = "AdcsProxy",
    [switch]$configureHttps = $true,
    [string]$fqdn = '',
    [string]$jamfProDn = ''
)
```

Some other configurations are easy to adjust in the script. For example, if we have an IT security rule that all service-to-service client certificates will have a validity period of one year, we would locate the client certificate line in the script and change the "10" to a "1". If you do this, set up a calendar invite to your team well ahead of the expiration so you can schedule a change to update the identity. Otherwise the system will break with the expiration is reached.

```
$clientCert = New-SelfSignedCertificate ` 
    -CertStoreLocation cert:\localmachine\my -DnsName "$jamfProDn" ` 
    -KeyExportPolicy Exportable ` 
    -KeyUsage DigitalSignature, DataEncipherment,KeyEncipherment ` 
    -Signer $cert ` 
    -NotAfter (Get-Date).AddYears(10)
```

Use a Domain Service Account when Authenticating to ADCS

Scenario

The default installation requires that the ADCS Connector host be given rights to ADCS and, in an enterprise CA, also to one or more templates. Some organizations may prefer to use a domain user service account instead of the Connector host's computer account, and IIS can be configured to support this.

If you are considering making this customization, please think through the implications. We want the highest possible security on the authentication to any template that permits an arbitrary subject to be specified in the CSR. In this case, we might prefer the Connector's default computer account authentication scheme. User service account are portable in that they can be used anywhere, and the user name and password are going to be known by some humans and transferred between them during the setup process. Using a computer account to authenticate to ADCS is not portable and its credentials are less likely to be exposed. The computer account's AD password is randomized at the time when a host is bound to AD and is never recorded or transferred in plaintext.

<https://docs.microsoft.com/en-us/iis/manage/configuring-security/ensure-security-isolation-for-web-sites> is a useful reference. The standard installer follows these guidelines in that it runs the Connector's IIS site within the standard app pool. However, if a domain user service account is required, the following instructions may be used.

Procedure: Using a Domain Service Account instead of the default Computer Account

Service accounts are typically configured with "User cannot change password" and "Password never expires". In the screenshots below, the user name is "AdcsProxyAccessUser". Be sure the password is complex -- *this account will have rights in ADCS so we need to be certain that it is well secured.*

Once you have the login for your service account, open the CA configuration console (refer back to Step 3 of the "Installing the Jamf ADCS Connector" section in this document...) and give the user "Request Certificates" permission in the CA properties Security tab, and, if using an enterprise CA, go to the Template Configuration console to give it enrollment permission on any template(s) you will be using to provision client certificates for Jamf Pro-managed devices.

If you had previously granted these permissions to the ADCSC host, you can remove those after adding the new service account.

The image contains three side-by-side screenshots of Windows management consoles:

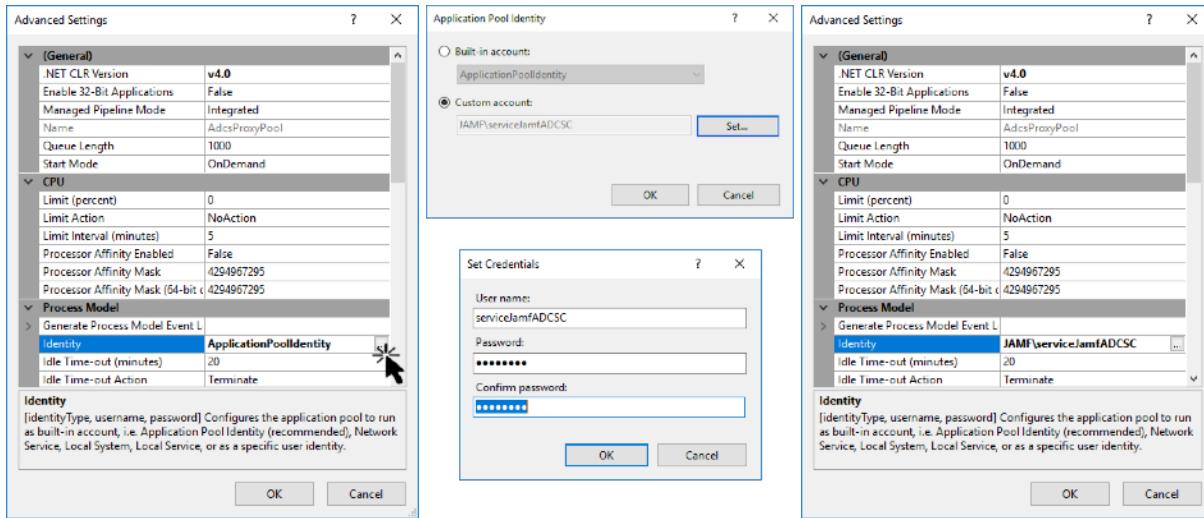
- Left Screenshot:** "AdcsProxyAccessUser Properties" dialog. It shows the "General" tab with the "User logon name" set to "JAMF\AdcsProxyAccessUser". Below it, the "Log On To" dropdown is also set to "AdcsProxyAccessUser". The "Account options" section includes checkboxes for "User cannot change password" and "Password never expires".
- Middle Screenshot:** "ca Properties" dialog. It shows the "Security" tab with the "Group or user names" list containing "AdcsProxyAccessUser (JAMF\AdcsProxyAccessUser)" and several local administrator groups. Below is a table of "Permissions for AdcsProxyAccessUser" with checkboxes for "Read", "Manage CA", and "Request Certificates".
- Right Screenshot:** "JamfADCSConnector Properties" dialog. It shows the "Security" tab with the "Group or user names" list containing "AdcsProxyAccessUser (JAMF\AdcsProxyAccessUser)" and several local administrator groups. Below is a table of "Permissions for AdcsProxyAccessUser" with checkboxes for "Full Control", "Read", "Write", "Enroll", and "Autorenew".

Then we can run the Connector as our service account and it will replace the Connector host as the identity that authenticates to ADCS. Highlight the ADCSProxyPool and click "Advanced Settings".

The screenshot shows the IIS Manager interface under the "Application Pools" section. The "Actions" pane on the right has a red box around the "Advanced Settings..." link under "Edit Application Pool". The main table lists application pools, with "AdcsProxyPool" selected and its details shown in the center pane.

Name	Status	.NET CLR V...	Managed Pipe...	Identity	Applications
.NET v4.5	Started	v4.0	Integrated	ApplicationPoolId...	0
.NET v4.5 Classic	Started	v4.0	Classic	ApplicationPoolId...	0
AdcsProxyPool	Started	v4.0	Integrated	ApplicationPoolId...	1
DefaultAppPool	Started	v4.0	Integrated	ApplicationPoolId...	1

Highlight the "Identity" setting and click the "..." button to the right of the current setting... applicationPoolIdentity. Use the "Set..." button in the dialog to switch to a custom account and enter your service account's <domain>username and password. Click the OK button and you will see your change listed in Advanced Settings. Use the OK button to close the dialog.



The ADCS template (or templates) used by the connector will need to be reconfigured so that the domain service user has the required permissions and permissions for the ADCS Connector host are removed.

[Running the ADCS Connector without Domain Binding.](#)

Some organizations do not allow connections from their DMZ network zone to the primary corporate AD that manages ADCS. In some cases, there will be a separate DMZ AD domain used to manage DMZ server access and to support Windows management, but this domain will not have any trust relationship with the internal AD running ADCS.

There are two possible approaches to this issue.

- 1) Install the Jamf ADCS Connector on an internal server and bind to the same domain as ADCS, or to one with cross-domain trust. Use a DMZ reverse proxy to allow Jamf Cloud connections to traverse the DMZ and reach the internal ADCS Connector server.
- 2) Place the ADCS Connector in the DMZ but do not bind it to the internal domain. Instead, use a *local* account with the same username and password as the domain service account. When the Connector attempts to connect to ADCS, it will not provide a domain and ADCS will assume the account is in its bound domain. Since the username and password match a domain account, the authentication will still work.

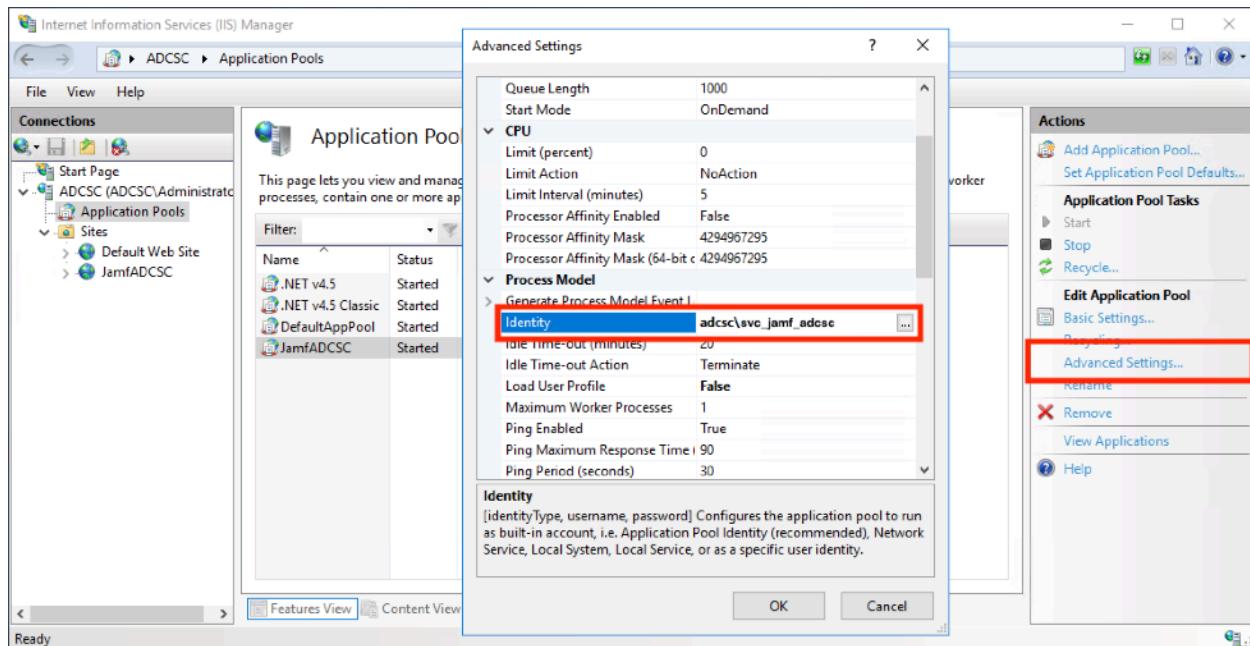
To implement option two, you will follow the same procedure described above to use a domain service account, but on the Connector, you will create a local account and tell the IIS App Pool to run as the local version of the account instead of the domain account.

To create the local account on the IIS server:

1. Click on Windows' Start button and open "Server Manager".
2. Click on "Tools" and select "Computer Management"

3. In Computer Management, expand “Local Users and Groups” then right-click on “Users” and select “New User”. Enter the same username and password (case-sensitive) used when you created the domain service account that has permissions to the CA and template. Ensure that “Require Password Change is NOT checked and that “Cannot change password” and “Password does not expire” ARE checked, just as you typically would for any service account you create. Click the “Create” button to create the user and then close out of Server Manager.

The remaining steps are nearly the identical to those described in the previous section. You still grant permissions to the CA and template to the domain version of the service account. However, when configuring the IIS appPool, specify the local version of the account by entering the user as <machineName>\<username>. In the sample below, "adcsc" is the server's machine name, not the name of a domain.



Configuring IIS to use an alternate Server Certificate

The server identity is used by the server when negotiating TLS connections with clients. The Connector installer instructs the Windows OS running on the Connector host to generates a self-signed identity for this purpose. Your organization may prefer to use certificates generated by your own internal PKI or from a third party public CA on its web servers instead of the self-signed ones configured by Jamf's ADCSC installer.

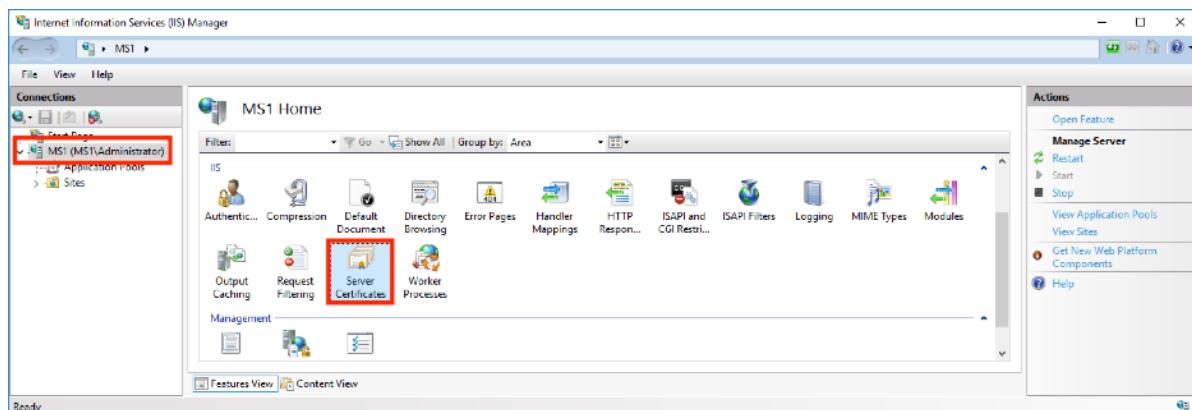
The following basic steps are used to install a server identity for IIS:

- 1) Run the ADCSC deploy script.
- 2) Obtain a new server identity from your preferred source.
- 3) Install your identity on the IIS server.
- 4) Configure the IIS site to use that identity instead of the one created by the ADCSC installation script. To do so, select the AdcsProxy Site in IIS Manger and click "Bindings...".

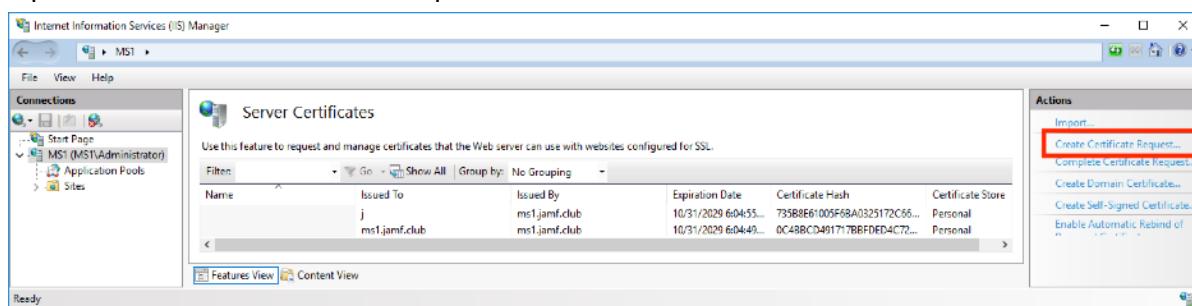
Obtaining a Certificate Signing Request

Your CA administrator or public certificate vendor will often ask that you provide a Certificate Signing Request ("CSR"). If you create this on the IIS server, the private key for the identity will remain on the server, so this is often the preferred workflow. There are many utilities for creating CSRs, including one built into IIS that is often used.

- 1) Highlight the server name in IIS Manager and click "Server Certificates".



- 2) Open the "Create Certificate Request" Wizard under "Actions".



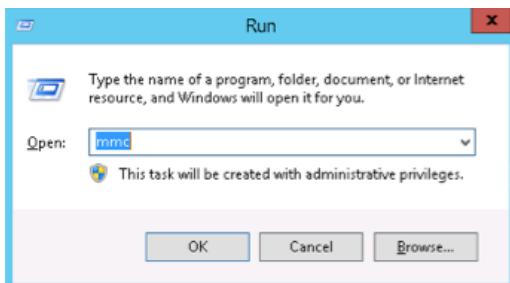
- 3) The Wizard will walk you through the rest of the process for configuring and saving the CSR. The Common Name (CN) will be the host name that Jamf Pro connects. In the case of Jamf Cloud configurations, this is the external DNS ("VIP") that resolves to your external IP address. Use the default Microsoft RSA SChannel Cryptographic Provider and a bit length of at least 2048.

Configuring IIS to Use the Alternate Server Identity

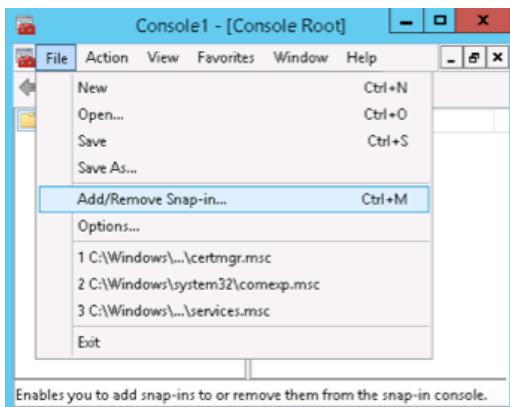
Once you already have the identity .pfx file that you want to use as the ADCSC site's SSL server certificate, it's an easy two-step process to install it. We'll add it to the Windows certificate store using the Certificates mmc snap-in, then tell IIS to use it to secure our site.

First, upload the identity file to the Windows keystore:

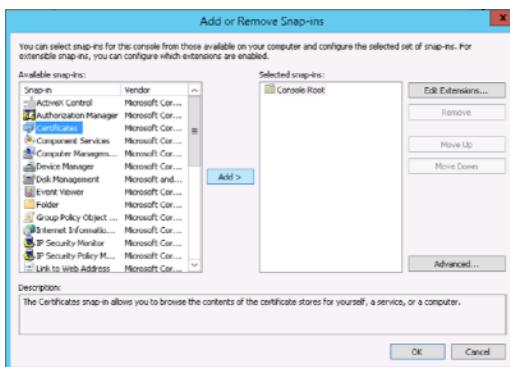
1. On the Start menu click Run and then type mmc



2. Select File > Add/Remove Snap-in



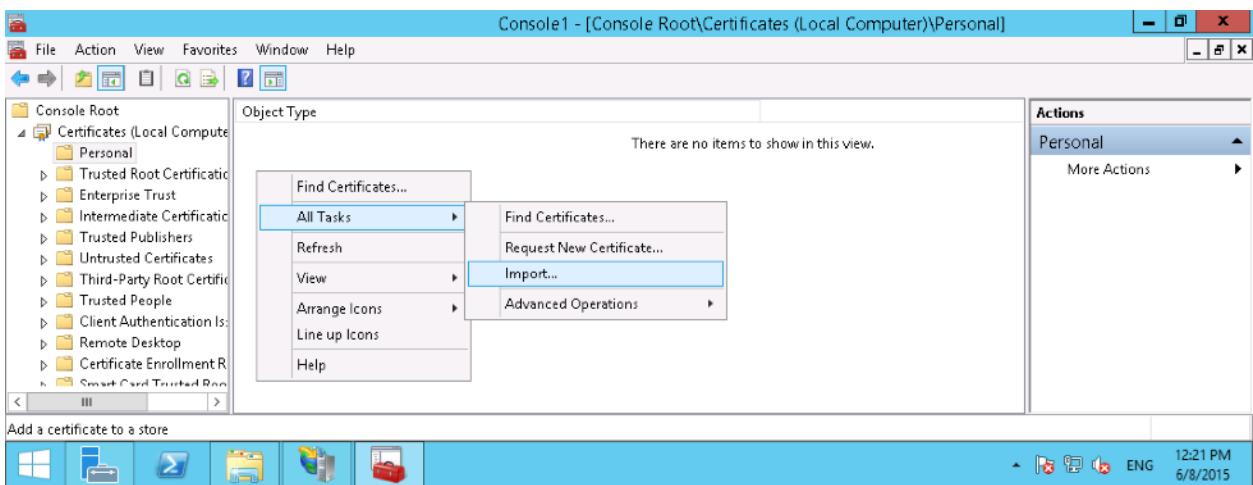
3. Click Certificates > Add



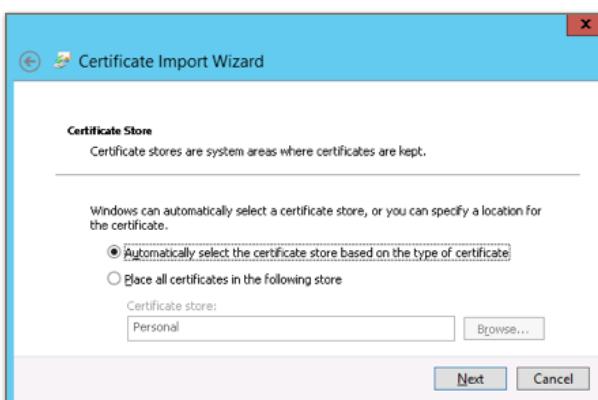
- Select Computer Account and then click Next.



- Select Local Computer and then click Finish. Then close out of the "add snap-in" window.
- Click the + to expand the certificates (local computer) console tree and look for the personal directory/folder. Right-click on the Personal certificates folder and select All Tasks > Import from the contextual menu.

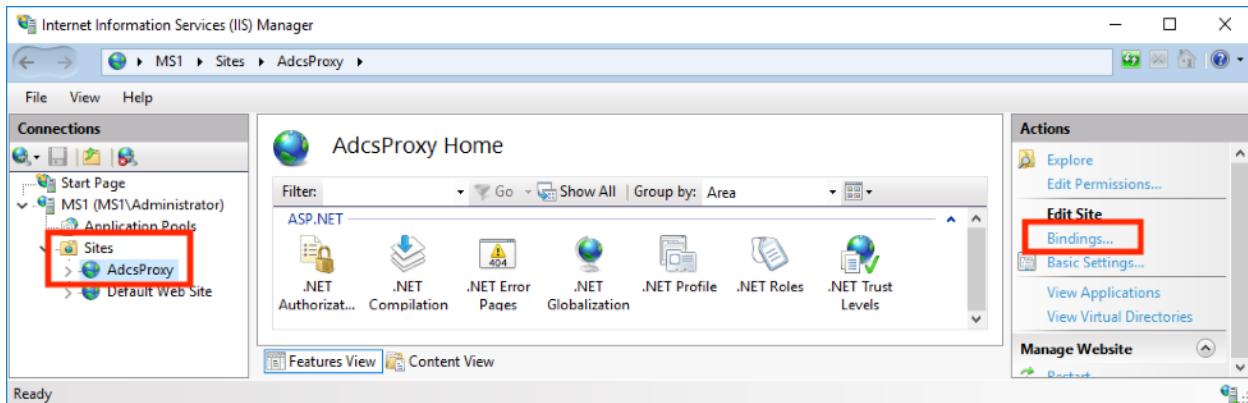


- Follow the certificate import wizard to import your primary certificate from the .pfx file. When prompted, choose to "automatically place the certificates in the certificate stores based on the type of the certificate". The advantage of that choice is that the wizard will correctly distribute the .pfx's components, putting your server cert into Personal, and any root or intermediate certs into their correct locations as well.

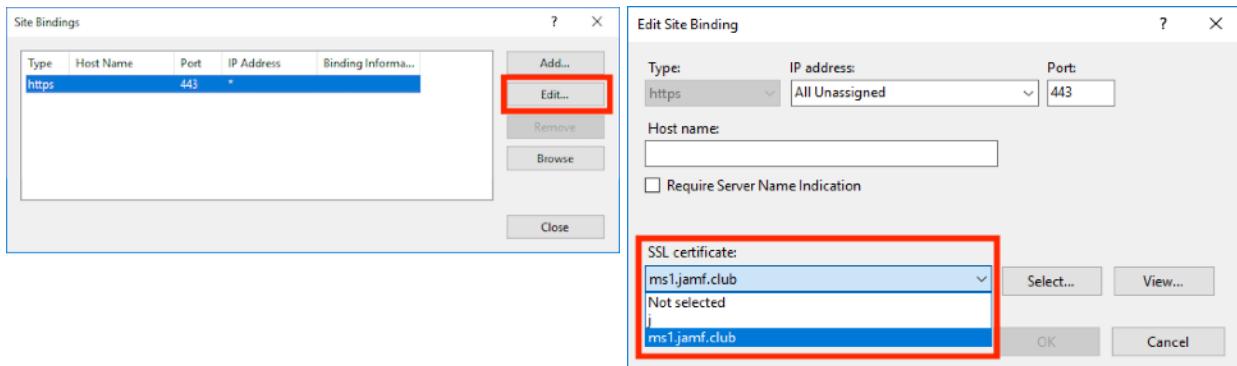


If your identity needs root or intermediate certificates for its source CA trust chain that were not included in the .pfx file, make sure they're already installed on your windows server. If not, you'll need to add them to the Windows keystore using the same procedure.

Now we just need to tell IIS to use our newly-installed identity. Highlight the ADCS Proxy site in IIS Manager, and click "Bindings..."



Edit the https binding and select the desired certificate from the SSL certificate drop-down menu.

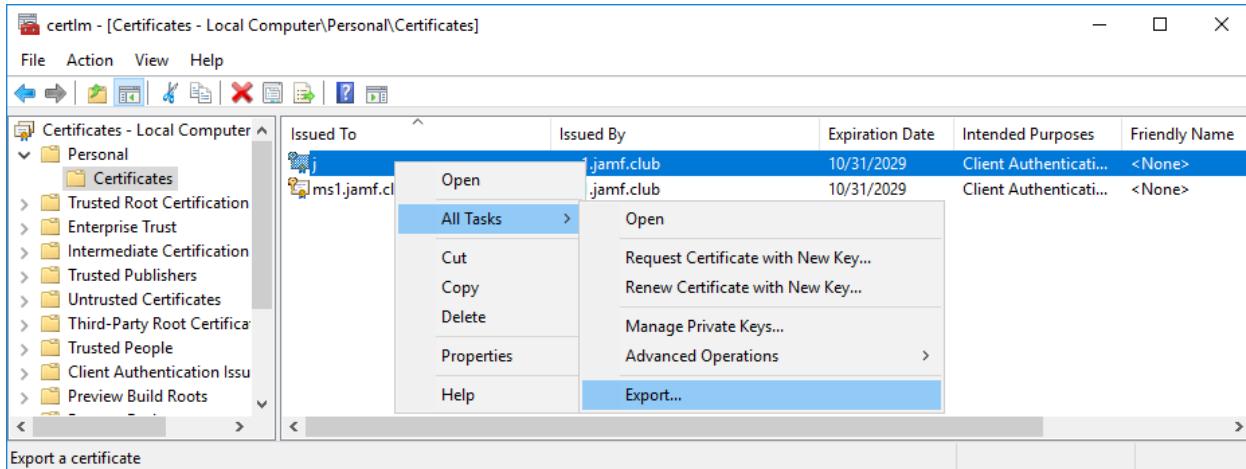


Replacing a server certificate in IIS prior to expiration

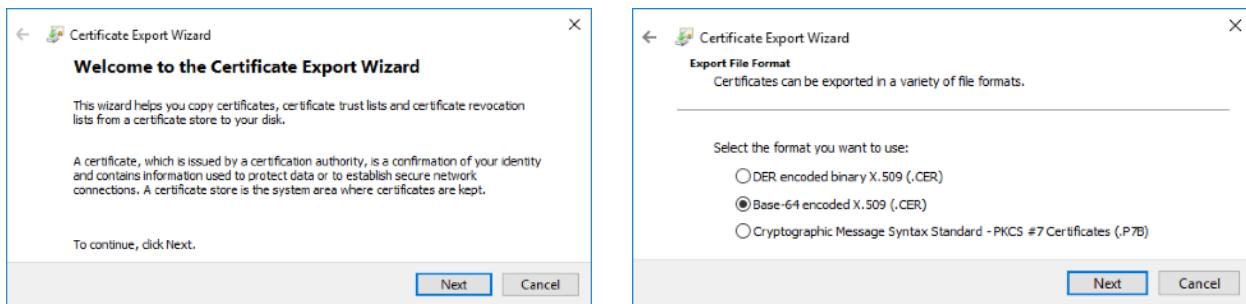
You should replace your IIS server certificate prior to expiration. If you don't, Jamf Pro will no longer be able to negotiate TLS connections once the expiration date has passed. The steps to follow are the same as the initial installation. You can install a new certificate any time you want and it doesn't matter if it's an update of the existing certificate or you create a brand new one... the only requirement is that the public key that is uploaded in the ADCS Connector PKI entry in Jamf Pro matches the server identity.

Configuring IIS to use an alternate Client Certificate

If you have another identity file (.pfx or .p12) that you want to use to authenticate Jamf Pro to IIS, you will need to use its public key in IIS's Client Certificate Mapping configuration. You can use Windows' certificates utility to export the public key. Open Computer Certificates ("certlm"), locate the client certificate you want to use, right-click on the identity and select "All Tasks > Export...".



The wizard will step you through the export process. Do not export the private key. Select the Base-64 export format.



Open the exported .cer file and copy the section between the BEGIN and END lines.

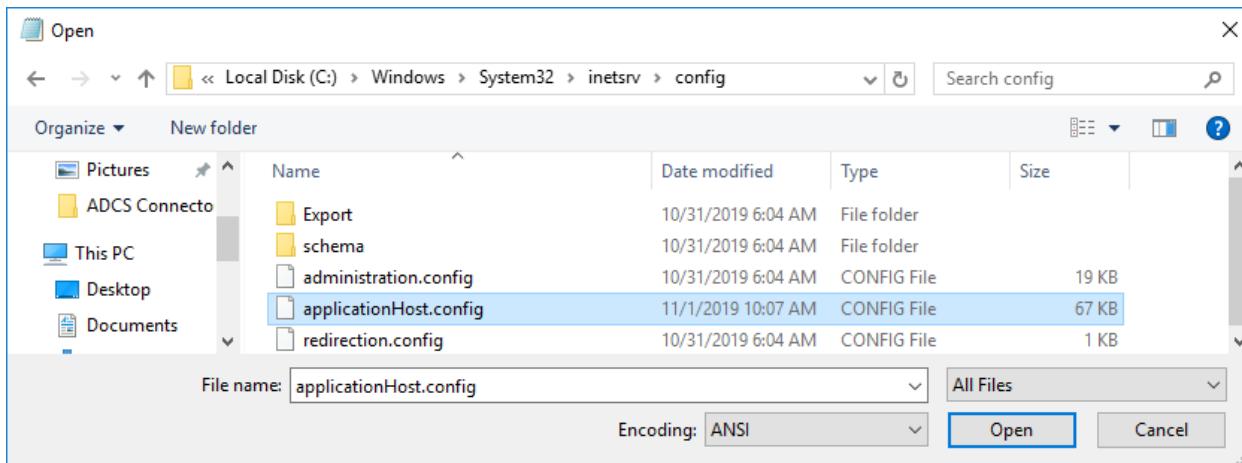
```
-----BEGIN CERTIFICATE-----  
MIIDLTCAhBgwIBAgIQUzzdY0zso80KakAr/SFZzANBgkqhkiG9w0B4QsFADY  
MRYwAYDVQQDA1tcEuanFTZ15jbhV1MB4XDTE5MTAxMTEyNTQ1MFoXDTI5MTA2  
MTEzMDQ1NVowDDE0MgGA1UEAwBaJCAISuOQYJKoZIhvJlAQEBBQAQdgEPADCC  
AQcGgEBAMVnT/CmUFNXYufmeqJ9pArvLihzw85ZB8bmMfuejwuZPSWmyLxDV  
LGx+72jP17qgPyhSHBzn+ofrIZs3d01jyINTG7k90I54vZSDvUw07abd/eN0f35  
Z/NiUlzPHL6+oeVnrlarYi/c1CV+wBC60bM10SM6xChVu1EpuNglnFhMu5n  
TgfuSuDSxdMsFgucQ2IssGrPxP-d/FANTaqcdMJuuhWt1AbpyxFET+uBytiv03yik  
h+iszaWlzqgZKUhrcOpSK6etpiE7+plwk5wklpuVEgv0no08rePxhz+GA00  
8KR4DHx9przcc4Ng2NsY1rnsY4TpC2wEAaAh/MH0wDgYDVR0PMQH/BADQ4uSw  
MB0GA1UDjQQWMBQGCCsGAQUFbwMCBgrBgfEFBQcDAT/MBgIVMREEBTADggFqMB8G  
AUU1wQYBaAFmRU1wP3xLFDcJ3mlyK40XWj0RM/B0GAIUdQnNBFRF12i+1E9F  
9Y4Wag1Tmew9/Zle4TANBgkqhkiG9w0BAQsFAOCARQEAJX+Xozi1x40q1gn81737  
AmP67qE+KaciRmV7pNzsZRn4iYYDrXrqzuCb9Z9mZ0Y3kWnJotVmocMvhzCe5  
Btk2ujNvNcIzVjAlIn-R2br/pWshEnSE9BP3qVeURe3YF5qPFGBbEwA8ck8GZ4zJH  
NGDSNe4zsCQ3ZDmub1HntS2bta8Ub5xo9oCNHv1AzhIMkAF1Lsg/co/Y13wCF77ahB4s/FPhbtVxVYzNNEY4HzmjTSvRLu6MMI+8H1FM9PLw/U8IRduKdHe0ouKx  
YYweXtDNTq1JBc1ayczasPU/jbJ7jnRHbzG0qucb1TRn421L1Kp00wFrdeSSsW1o  
au=-----END CERTIFICATE-----
```

The instructions and screen shots for navigating to IIS's client certificate authentication configuration were demonstrated in "To review client certificate authentication settings..." in the above section where installation script configurations were discussed. You can use the configuration editor screen to paste the base-64 of the new key to replace the one created by the installer.

Alternately, you can edit the IIS configuration file manually.

Navigate to "<C:\Windows\System32\inetsrv\config\applicationHost.config>" and make a backup copy before making any changes.

Run NotePad (or another text editor) as Administrator, change the file filter to "All Files" and navigating to the IIS applicationHost configuration file.



Replace the existing base-64 key with the contents you copied from the .der document and remove any carriage returns to the key is one contiguous string.

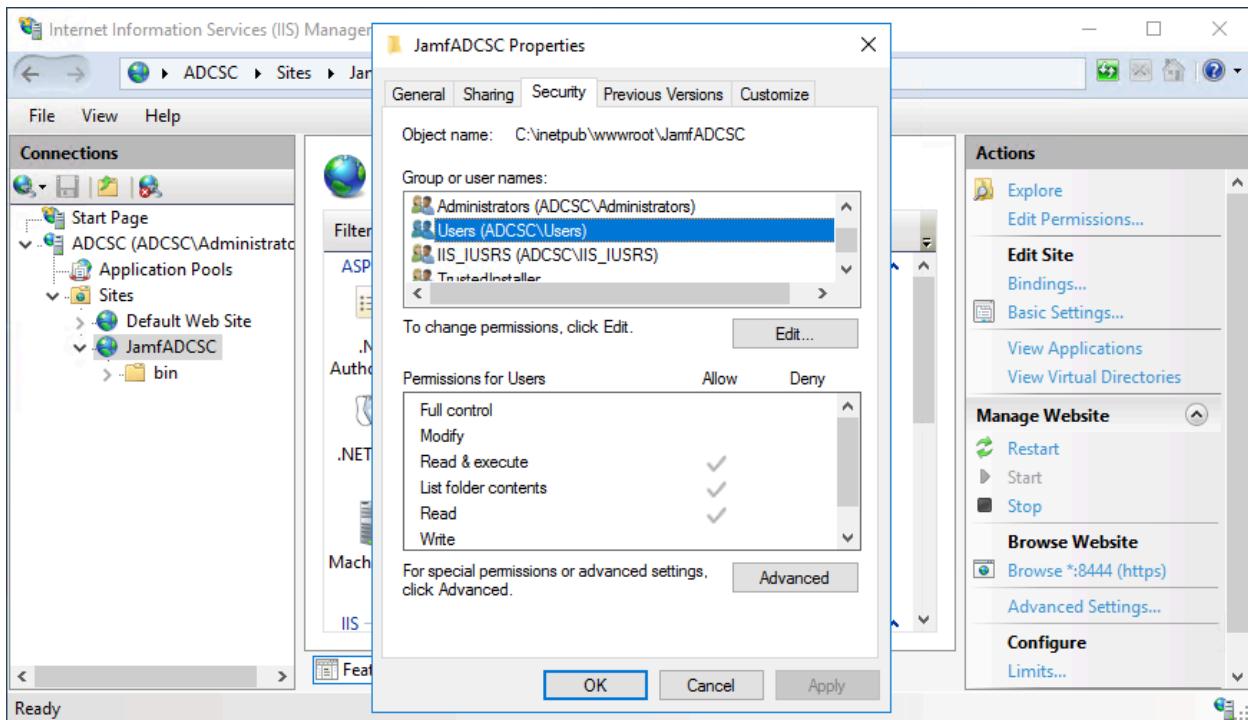
```
<location path="AdcsProxy">
  <system.webServer>
    <security>
      <access sslFlags="Ssl, SslNegotiateCert, SslRequireCert" />
      <authentication>
        <iisClientCertificateMappingAuthentication enabled="true" manyToOneCertificateMappingsEnabled="false">
          <oneToOneMappings>
            <add userName="AdcsProxyAccessUser" password="[enc:IIS_CngProvider:d0dgfQJammKTP56xq0t6fdIw9KvR9A8rb7ix1wQM1Z2gdZoyR7SFgFss1A7V7pwzs5Rn4iVYD1rXrgzwCb9ZmZDmY3kwnJotVMocMhxzCe5Btk2u:jVNvCIzVjALIrRZbr7pwshEnSE9BPJqV0uRE3YFSqPFG0bEwA8cK8GZ4zjiHNGDSHEazsCQ3ZGZnB8Jlnt852bTa8Ulxoo9C]" />
          </oneToOneMappings>
        </iisClientCertificateMappingAuthentication>
      </authentication>
    </security>
  </system.webServer>
</location>

</configuration>
```

Save the configuration file and restart IIS.

By default, the ADCS Connector installer will have mapped the authentication certificate to a local Windows user. This user can be left as is, or changed to a different user. The only

requirement is that the user account needs to have permissions to the ADCSC site in IIS. In the screenshot below, we see that all local users have permissions to the site, so any local user can be mapped to the client authentication server.



Requirements for Reverse Proxy, Load-Balanced, and Web Application Firewall Network Configuration

The Connector will be implemented in many different IT environments, each with different network layouts and practices for service deployment. Some of these will mandate high-availability, reverse proxy, or web application firewall configurations. The understanding needed to implement any of these is similar. They all work well with the Connector. Network administrators only need to understand that ADCSC is just like any other HTTPS web site running on IIS, no different than any other web services in an organization, and that it implements one-to-one client certificate authentication.

The implications are:

- The server certificate used by the proxy, load balancer, or web application firewall when negotiating TLS must have a name or SAN that matches the host name configured in Jamf Pro and must also match the public key configured in Jamf Pro.
- When a proxy or load balancer is configured for TCP pass-through, the client certificate presented by Jamf Pro will be passed transparently to IIS and IIS will do the verification.
- When a proxy, load balancer, or web application firewall is configured for TLS interception, the proxy should use the public key of Jamf Pro's client certificate to verify the authenticity of the Jamf Pro connection. Organizations can use whatever means they prefer to authenticate the connection between the proxy and IIS (e.g. NTLM) so long as both ends of the connection are configured in tandem. If certificate based authentication is used, the proxy and IIS will be set to require client certificate authentication and the proxy will use the identity whose public key has been set to verify the client certificate in IIS. This may or may not be the same client certificate used by Jamf Pro to connect to the proxy.
- When a load balancer is used, the same rules for server and client certificate authentication apply. There's no limit to the number of ADCS Connector/IIS instances that can be in a load-balanced server pool. However, you should understand that the certificate signing process in ADCS has two-steps. First, a signing request is submitted, and then a subsequent connection is made to retrieve the finished signature. ADCS will only allow retrievals by the same identity that made the request. The implication here is that if load-balanced Connectors are running with the default AppPool identity, you should use a primary/failover configuration for high availability. (The load on the connector will never be high enough to require true load balancing.) If you want to use methods like round-robin or least-load, you'll need to set the app pool identity on all Connectors in the cluster to use the same domain service account. Then any Connector instance will be able to collect signatures generated by any other.

The critical understanding here is that ADCSC is front-ended by Microsoft IIS. The Connector site itself is not involved in negotiating network connections, TLS, or authentication. Customers may route the HTTP/TCP traffic to their Connector in any way that's supported by IIS and

consistent with their own standards and practices. The configuration steps will be based on the documentation from your proxy's manufacturer and Microsoft's IIS documentation.

The following is an example reverse proxy/load balanced planning diagram:

 Internet Client DMZ -> Internal				
Owner?	Mac team	Network team	Proxy admin	Mac team
What?	REST over HTTPS (TCP Port 443 is typical but use whatever you like...)			
Auth?	TLS with client Identity cert	None (TCP Passthrough)	Verifies client's Identity cert	Anonymous? Simple? Re-encrypt+Client Certificate?
Hostname	my.jamfcloud.com	edge.my.org	proxy.my.org	us_srv_093.my.internal
IP Address	Source IPs for access rule: 54.208.14.206, 54.208.84.215, 52.1.62.94, 52.1.215.211, 52.203.216.218, 34.233.253.88, 34.234.26.211, 52.72.152.43, 52.39.2.203, 52.39.4.253	xx.xx.xx.xx <small>(This is needed when creating the A-record in external DNS)</small>	yy.yy.yy.yy <small>(This is needed to set the Firewall rule on Windows Server.)</small>	zz.zz.zz.zz
Cert	Client Identity Certificate Source: Internal CA Owner: Mac Team CN/SAN This can be anything you want, but just to make things easier to identify, we usually use the hostname of the client (E.g. my.jamfcloud.com) or a domain service account.	n/a	Server Identity Certificate Source: Internal CA. Owner: Proxy Team CN/SAN proxy.my.org	IIS Identity Certificate Source: Internal CA Owner: Mac Team CN/SAN: Whatever the proxy is connecting to. E.g. us_srv_093.my.internal or a C-Name alias to it.
DNS	AWS Route 53 managed by Jamf	External DNS A-Record edge.my.org -> <external IP address?>	n/a	Internal DNS A record already exists for the IIS host if it has an IP reservation in DHCP. us_srv_093.my.internal -> zz.zz.zz.zz
Firewall rule	n/a	Allow TCP 443 from the list of AWS VPC outbound NAT source IPs to the VIP that leads to the proxy.	n/a — Load Balancer can already hit Proxy.	Allow TCP 443 only from Proxy's IP address. Consider isolating other services from outside access as well.

Configuring ADCS to use an Alternate Port

Scenario

Since the Windows OS computer running ADCS Connector needs to be bound to a domain, it typically is on the same network as both a domain controller and the ADCS CA. For this reason, we rarely need to do any special firewall changes to have them speaking to each other on the standard communications ports used by Windows.

However, there may be cases where the ADCS server is isolated, which might be the case when if you run a stand-alone CA that only services a small number of authorized services rather than being open to all devices.

Perhaps the best approach in this situation might be to put a reverse proxy in the DMZ to forward Jamf Pro's HTTPS traffic to the ADCS Connector installed on the same isolated/internal network as ADCS.

If the ADCS Connector will run in a different network zone than ADCS, firewall admins may not wish to open all of the dynamic DCOM ports between the two zones. In this case, Microsoft allows customization of the ports.

Procedure

A summary of the steps is included here to show the basic idea, but you should use Microsoft's documentation when doing this. Reference: <https://social.technet.microsoft.com/wiki/contents/articles/1559.how-to-configure-a-static-dcom-port-for-ad-cs.aspx>

To configure the Active Directory Domain Services (AD CS) certification authority (CA) service (CertSvc) to listen on a static DCOM port

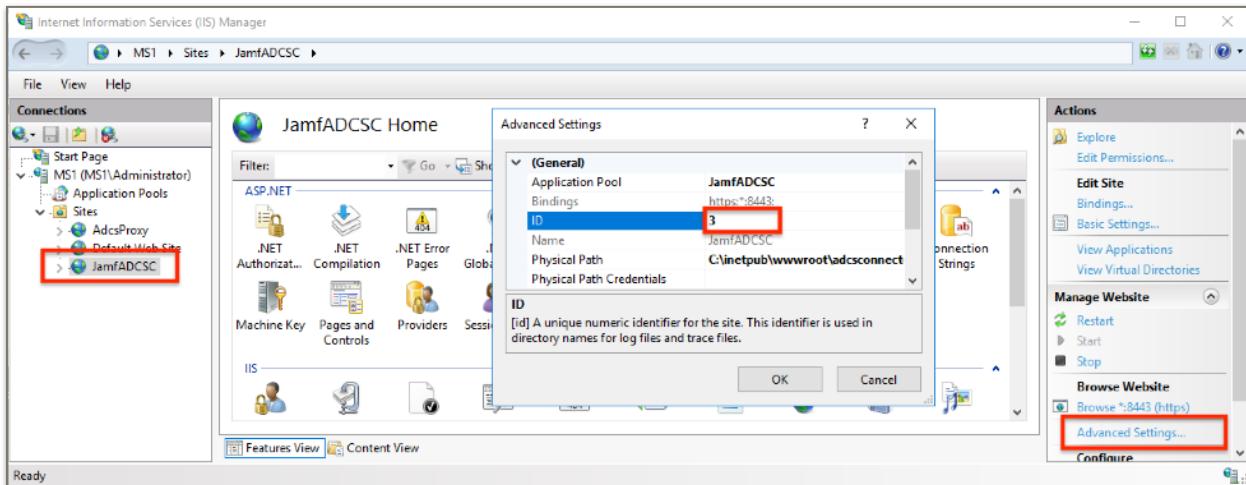
1. Log on with an account that has local administrator permission on the CA
2. Open the **Component Services** snap-In (dcomcnfg.exe).
3. In the left pane of the **Component Services** snap-In, expand **Component Services, Computers, My Computer**, and then **DCOM Config**.
4. In the right pane, select **CertSrv Request**.
5. On the **Action** menu, click **Properties**.
6. On the **Endpoints** tab, click **Add**.
7. Select **Use static endpoint**, enter an unused TCP port number, for example, 4000, and then click **OK** twice.
8. Close the **Component Services** snap-In.
9. Restart the certification authority service.

```
net stop certsVC  
net start certsVC
```

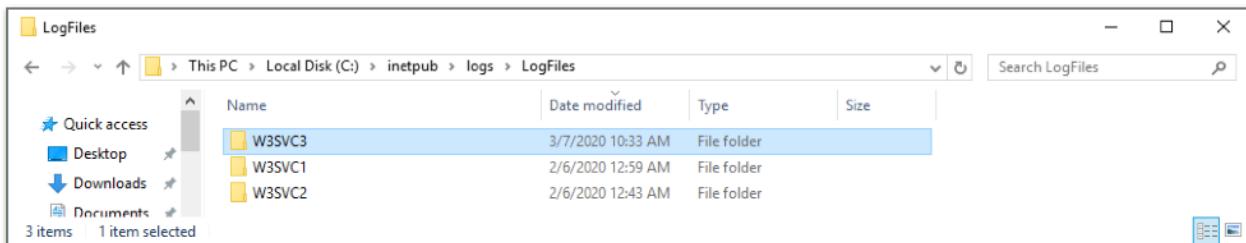
Troubleshooting Strategies

Viewing the IIS Access Logs

Go to `c:\inetpub\logs\LogFiles`. You may see multiple folders there... one for each site hosted in IIS. You might be able to click into them and figure out which one is for your ADCS Connector site based on the timestamps of the folders or their logged connections. Otherwise, get the site number to find the right folder. Click on your Connector site and click "Advanced Settings...". In the example below, we see the ID number for our Connector site is "3".



In the logs folder, this corresponds to "W3SVC3"...



Inside this folder, click on the most recent log file and you will see entries for each attempted connection.

Example Log Entries — 403.16

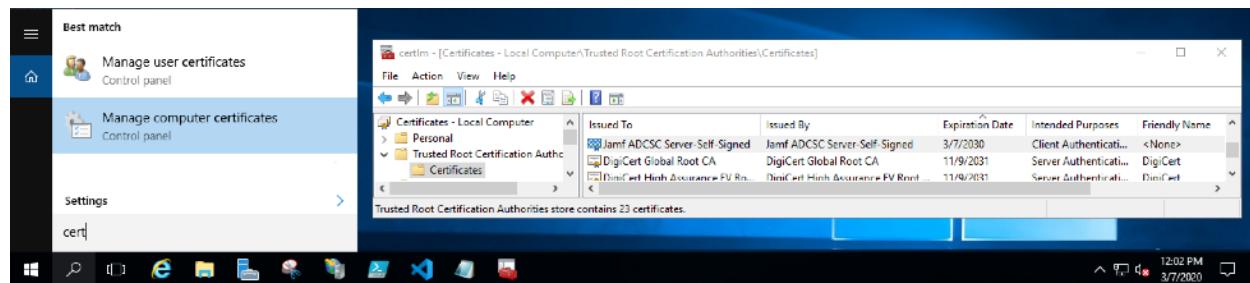
In the example below, we see that at 2020-03-07 15:33:11, a client with IP address 192.168.1.57 attempted to connect to the /api/v1/certificate/request endpoint on port 8443. An HTTP status of 403.16 was returned.

```
#Software: Microsoft Internet Information Services 10.0
#Version: 1.0
#Date: 2020-03-07 15:33:11
#Fields: date time s-ip cs-method cs-uri-stem cs-uri-query s-port cs-username c-ip
cs(User-Agent) cs(Referer) sc-status sc-substatus sc-win32-status time-taken
2020-03-07 15:33:11 192.168.1.57 POST /api/v1/certificate/request - 8443 -
192.168.1.47 curl/7.54.0 - 403 16 2148204809 1203
```

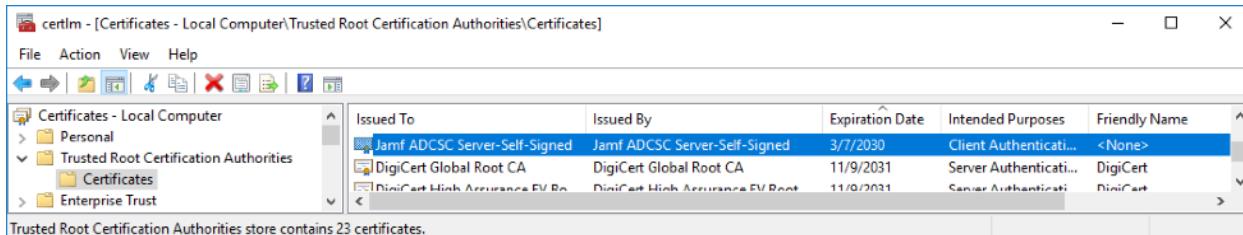
If we lookup that error, we'll find <https://support.microsoft.com/en-us/help/942061/error-message-when-you-visit-a-web-site-that-is-hosted-on-iis-7-0-http>.

So 403.16 indicates that the client presented an authentication certificate, and it probably matched up with the public key in the one-to-one client certificate mapping, but IIS couldn't establish a trust chain -- the CA cert that signed the client cert isn't in the *Trusted Root Certification Authorities* certificate store on the IIS server. If you let the ADCS Connector installer script generate the client certificate, it was signed by the server cert it used to create the site binding. If you provided a client cert from your own or a third-party CA, then you need to load the trust chain for that CA into *Trusted Root Certification Authorities*.

Bring up certlm. You can do a find on "cert" and select "Manage computer certificates". Make sure that the trust chain certs that signed your client authentication certificate are in here.



It's not uncommon for organizations to have a computer management restriction that blocks any new certs from being added to the trusted roots because if anything wrong gets in there you're a sitting duck for MITM attacks. We've seen several cases where the server where the Connector was being installed had this restriction but the SCCM admin wasn't aware of it.



Viewing the Jamf Software Server Logs

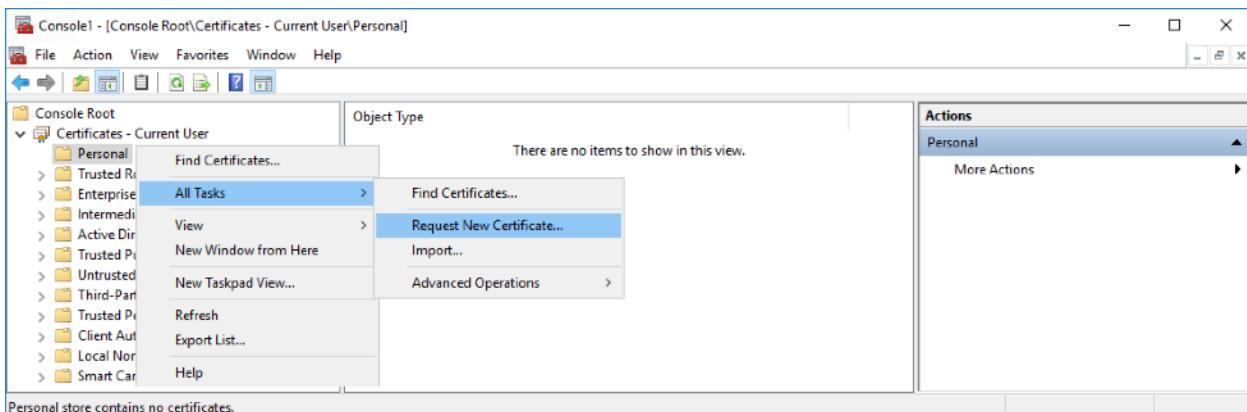
These are some errors you might see in the Jamf Pro logs and their meaning.

Log Error	Solution
<p>Caused by: com.jamfsoftware.pki.adcs.exception.AdcsConnectorCertificateNotIssuedException: INTERNAL_ERROR: System.Runtime.InteropServices.COMException – CCertRequest::Submit: The RPC server is unavailable. 0x800706ba (WIN32: 1722 RPC_S_SERVER_UNAVAILABLE)</p>	<p>The wrong hostname has been entered for the AD Domain Controller or the server is unreachable from the Connector host:</p> <p>Settings > Global Management > PKI Certificates > ca</p> <p>AD CS Server Integration Settings to integrate Jamf Pro with Active Directory Certificate Service</p> <p>FULLY QUALIFIED DOMAIN NAME Fully qualified domain name of the certificate authority server <input type="text" value="domainserver.jamf.com"/></p>
<p>Caused by: org.springframework.web.client.ResourceAccessException: I/O error on GET request for "https://192.168.1.198:8444/api/v1/version": Certificate for <192.168.1.198> doesn't match any of the subject alternative names: [adcsc.jamf.com, 192.168.1.198];</p>	<p>The server certificate uploaded to Jamf Pro does not match the subject or SAN of the ADCS Connector host name you entered in Jamf Pro.</p>

Troubleshooting with the Certificates MMC and Certreq

If you are having trouble getting certificates and suspect the CA or template configuration, you can remove Jamf entirely from the picture and test the backend configuration with native Microsoft tools.

To do so, just run the certificate snap-in on the Jamf ADSC Connector host and request a certificate from the template you want the Connector to use. If you can obtain a cert, you know that the host's computer account has the right permissions.



Microsoft's certreq utility uses the same DCOM interface as the Connector so that is another option that can be useful for testing.

Verifying Connectivity between Jamf ADCS Connector and Microsoft ADCS with Certutil

You have to wait for a few steps to complete if you want to use Jamf Pro to test ADCS Connector. You'd need to put a device into scope for a certificate profile, wait for the interaction with Apple Push and the device, then the communication with the Connector and from the Connector to ADCS. This is an important test but not efficient. It's easier if we can test one connection at a time.

If you suspect network or ADCS permissions issues, a simple approach is to take Jamf completely out of the picture and use microsoft-native tools for testing. Your network and CA admins will already be comfortable using them.

Microsoft offers a utility called "**certutil**" that uses the same communications interface and many of the same APIs as the Jamf ADCS Connector. It can be used to verify connection between the Connector and ADCS.

Procedure:

1. SysInternals' **psexec** command allows you to run a command prompt as the computer account. If you don't already have it installed on the host running the Connector, [download Sysinternals from Microsoft](#) and unzip the contents.
2. Verify that the ADCS Connector host computer account has permissions on its ADCS template. This was described earlier in this document.
3. Run cmd.exe on the ADCS Connector computer.

Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

You're logged in as a user. In this example, "administrator".
C:\Users\Administrator>whoami
adcsc\administrator

Check DNS resolves the ADCS hostname...
C:\Users\Administrator>nslookup ms.jamf.com

```
Address: 192.168.1.112
```

```
If your network allows ICMP to the ADCS host, you can ping it...
C:\Users\Administrator>ping ms.jamf.com
```

```
Pinging ms.jamf.com [192.168.1.112] with 32 bytes of data:
Reply from 192.168.1.112: bytes=32 time<1ms TTL=128
Ping statistics for 192.168.1.112:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
^C
```

```
Pinging the remote ADCS service will fail if you run as the current user because that
user does not have any template permissions. (Use your ADCS hostname and CA Name
instead of "ms.jamf.com\CA")...
```

```
C:\Users\Administrator>certutil -ping -config ms.jamf.com\CA
Connecting to ms.jamf.com\CA ...
Server could not be reached: The permissions on this certification authority do not
allow the current user to enroll for certificates. 0x80094011 (-2146877423
CERTSRV_E_ENROLL_DENIED) -- (63ms)
```

```
CertUtil: -ping command FAILED: 0x80094011 (-2146877423 CERTSRV_E_ENROLL_DENIED)
CertUtil: The permissions on this certification authority do not allow the current
user to enroll for certificates.
```

```
Cd into sysinternals dir (if it's not already in your path)...
C:\Users\Administrator> cd "C:\Users\Administrator\Downloads\PSTools\"
```

```
Elevate the cmd session to run as the computer account...
C:\Users\Administrator> psexec \\localhost -s cmd
```

```
PsExec v2.2 - Execute processes remotely
Copyright (C) 2001-2016 Mark Russinovich
Sysinternals - www.sysinternals.com
```

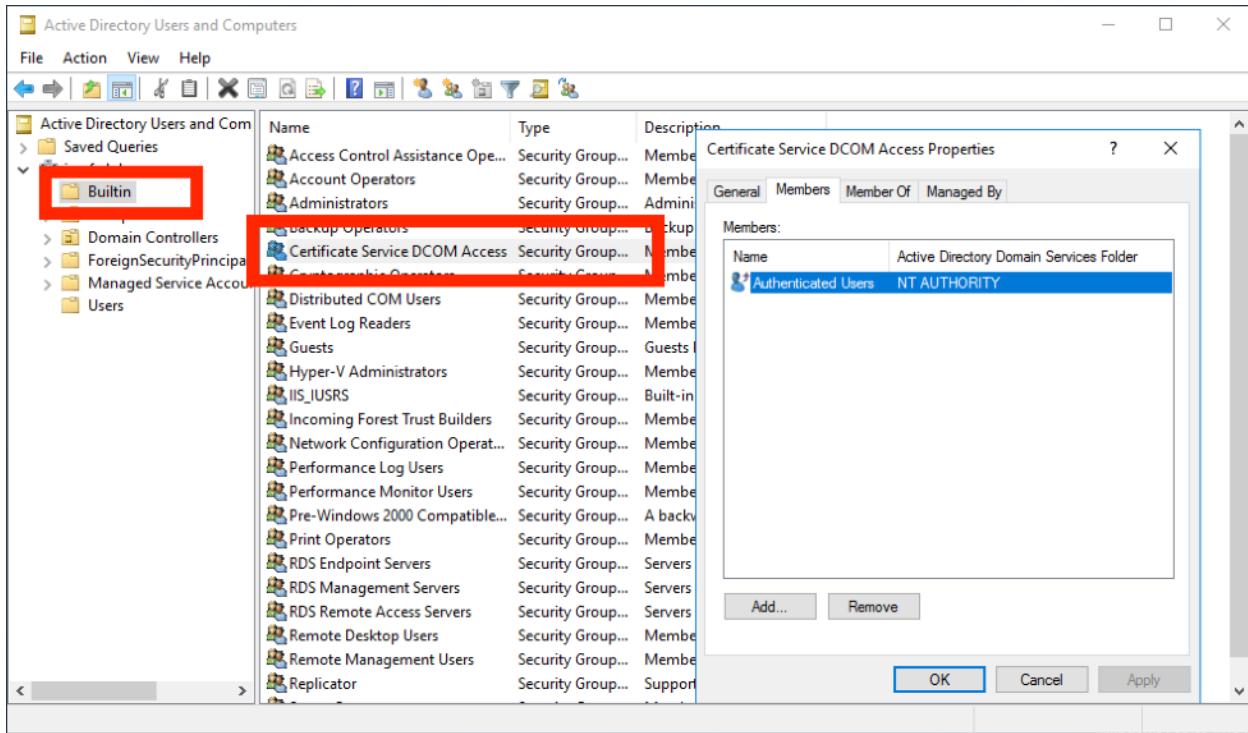
```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.
```

```
Now you're running as the computer account...
C:\Windows\system32>whoami
nt authority\system
```

```
Use certutil to test connection to ADCS. (Use your ADCS hostname and CA Name instead
of "ms.jamf.com\CA")...
C:\Windows\system32> certutil -ping -config ms.jamf.com\CA
Connecting to ms.jamf.com\CA ...
Server "ca" ICertRequest2 interface is alive (15ms)
CertUtil: -ping command completed successfully.
```

Check the DCOM Access Group

Windows' has a domain local group called Certificate Services DCOM Access. By default, Authenticated Users will be a member of this group and the connector will automatically be a member of the Authenticated Users group since it's bound to the domain. So it inherits all the permissions it needs. However, if your PKI admin has made this group more restrictive, you can explicitly add the Connector to the DCOM Access group.



Check the DCOM Access Restriction GPOs

It's very rare, but an administrator may configure Group Policy values under "Computer Configuration\Policies\Windows Settings\Security Settings\Local Policies\Security Options\Other":

DCOM: Machine Access Restrictions in Security Descriptor Definition Language (SDDL) syntax

DCOM: Machine Launch Restrictions in Security Descriptor Definition Language (SDDL) syntax

This will cause the "Edit Limits" button under Access Permissions and Launch and Activate Permissions on the COM Security tab to be greyed out. Ask your Windows admin to remove this restrictions if needed.

Frequently Asked Questions

I have Prod, Test, and QA Jamf Pros. Can I use one ADCS Connector to service all of them?

Yes. Just create the ADCS Connector entry in each Jamf Pro using the exact same information and certificates. Note that the subject for a client certificate can be arbitrary, so you can just make it something generic, like "JamfPro".

Why does Jamf Pro say the password for my client key file is wrong when I try to upload it?

You likely entered the password incorrectly. To verify that you have the correct password, try opening the file in KeyChain Access on a Mac or in the Certificates snap-in in Windows mmc. You can also use openssl, entering the keystore password when prompted.

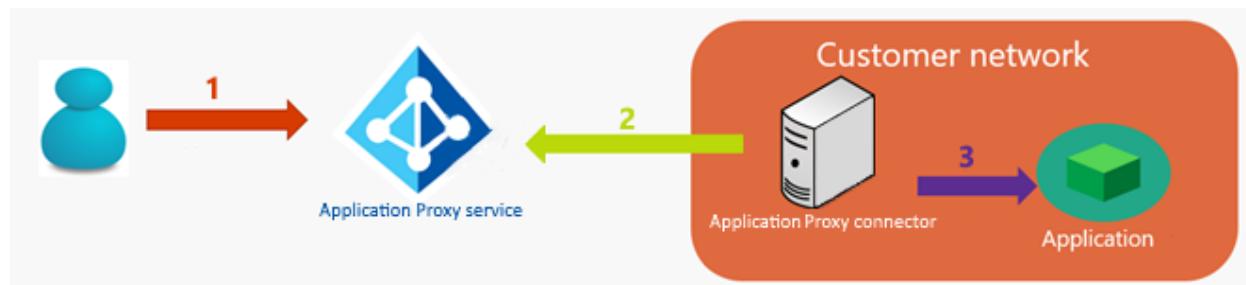
```
openssl pkcs12 -in mypfx.pfx -noout
```

If you don't have the correct password, you'll need to re-run the ADCS Connector installer to obtain a new key file and password.

Can I use Azure Web Application Proxy?

Organizations that prefer to use this Azure service should consider using the SCEP Proxy method for their certificate deployment. Microsoft NDES (SCEP) supports form-based authentication, which is supported by Azure Web Application Proxy.

Many organizations that adopt Azure's technology stack will be accustomed to using Web Application Proxy when connecting users to their web applications. It lets them take advantage of the Single-Sign-on ("SSO") authentication methods available with Azure AD and translate that authorization to other mechanisms (like Windows Integrated Auth) that might be required by their legacy applications. Even better, it uses an on-premise Connector component that initiates connections outbound to Azure. In this way, no inbound firewall rules are required into your internal network, making the solution much more agreeable to your networking and IT security teams.



Ref: "Network topology considerations when using Azure Active Directory Application Proxy" <https://docs.microsoft.com/en-us/azure/active-directory/manage-apps/application-proxy-network-topology#traffic-flow>

Microsoft recommends the use of Application Proxy to give remote users access to internal resources. Ref: <https://docs.microsoft.com/en-us/azure/active-directory/manage-apps/application-proxy>. You should consider that this is perhaps not relevant in the case of the Jamf ADCS Connector because users and their devices will never connect to ADCSC directly. The identity certificates they require will be obtained on their behalf by Jamf Pro itself, and then Jamf Pro will send the certificate to the clients.

ADCS Connector is not a user-facing app and therefore does not use SSO when it authenticates to a remote service. It is a web service for server-to-server communications. It is possible to pass Jamf Pro's HTTPS communications to ADCSC through the Azure App Proxy with "authentication disabled". The IIS instance would simply need to be configured to use whatever authentication method is used by the connecting service, in this case, the Web Application Firewall.

This would be a simple configuration if MS WAP would pass Jamf Pro's client-authentication certificate, or, alternately, would verify Jamf Pro's client certificate and then use any of its supported down-stream authentication methods when connecting to IIS. However, Azure Application Proxy decrypts and re-encrypts the traffic it proxies and the Azure/cloud version doesn't have native support for the client-certificate based authentication used by Jamf ADCD Connector. Microsoft states:

Can I publish a web application with client certificate authentication requirement?

No, this scenario isn't supported because Application Proxy will terminate TLS traffic.

Ref: <https://docs.microsoft.com/en-us/azure/active-directory/manage-apps/application-proxy-faq#can-i-publish-a-web-application-with-client-certificate-authentication-requirement>

However, using Web App Proxy with Jamf ADCS Connector is not impossible. You would, however, be trading one security approach for another. To do so, you would use Windows firewall (or another firewall and a VLAN containing only the ADCSC) to ensure that only Microsoft's Application Proxy Connector can reach the ADCSC host via the HTTPS port (typically 443) and turn off the certificate-based authentication for the ADCSC site in IIS and instead allow anonymous auth. Then configure your Azure Web App Proxy with no pre-authentication (passthrough). The service will never request the client certificate when Jamf Pro negotiates TLS, so the client certificate you provided to Jamf Pro would never be used. In this way, you gain the ability to use only outbound connections from your internal network, but at the significant cost of removing client certificate-based authentication. Given the sensitivity of the service being exposed (ADCS), please consider the implications of alternative security arrangements very carefully. Jamf is not recommending this approach.