# Implementation Note:

## Jamf Active Directory Certificate Services Connector

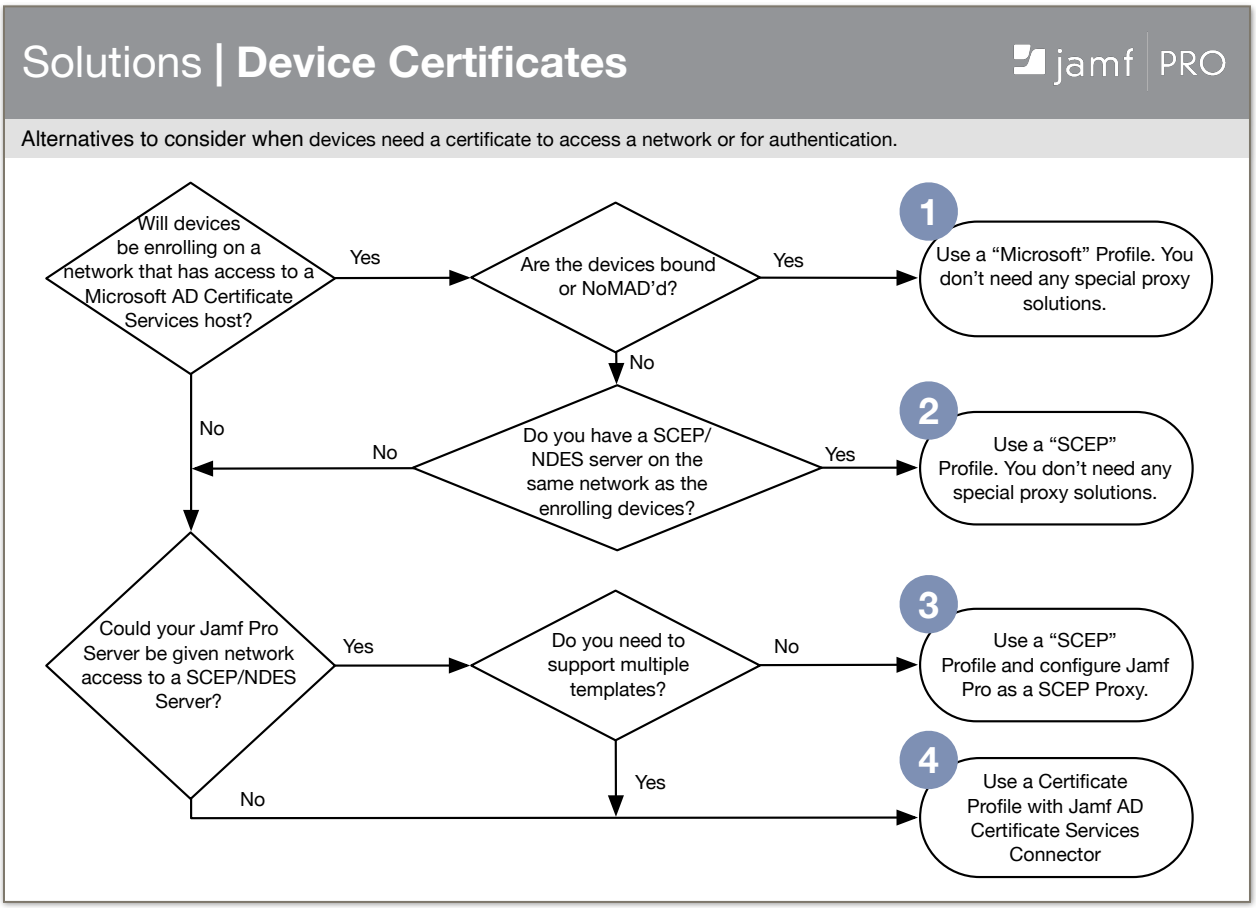
Jamf Implementation Engineering
16 November 2021/ol

jamf

CONTENTS

# 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.

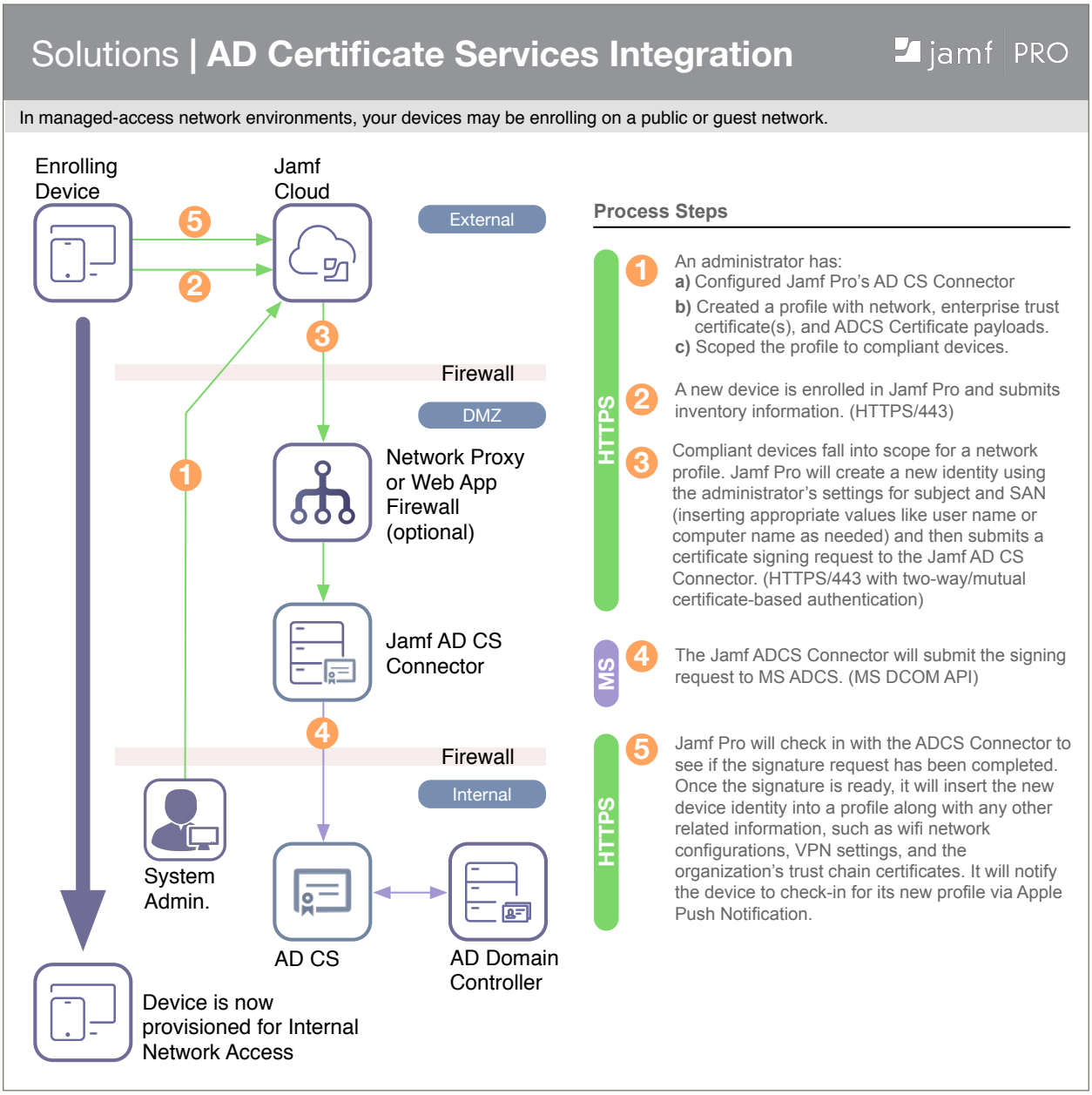
Option 3 allows the devices to use the Jamf Pro web application as a conduit for their transaction with a SCEP/Microsoft NDES service, and since the trust basis for all components is strong, this may be both acceptable and desirable, because in option 4, the Jamf Pro web application creates the private 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 leaves 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 Connector'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.



# Jamf ADCS Connector REST API

### Jamf Cloud with Jamf ADCS Connector in the DMZ

When Jamf Pro recognizes that a device needs a certificate, it generates a private key and a certificate signing request ("CSR") with the desired subject and subject alternative name ("SAN"). The CSR will need to be signed by the CA, but Jamf Pro does not have direct access to the CA, and may not even be running on a Windows server that would be able to use Microsoft's DCOM libraries so it will instead use the Jamf ADCS Connector's REST API endpoints to send a signing request and let the connector talk to ADCS on it's behalf.

**Step 1:** Check connectivity and API version: https://adcsc.my.org/api/v1/version

Here, "adcsc.my.org" is the external host name of the Connector. The port is not explicitly included in the URL in this example because port 443 is being used. This is an HTTP GET... there are no url parameters or HTTP body.

**Step 2:** Send a signing request: https://adcsc.my.org/api/v1/certificate/request

This is an HTTP POST. The data is supplied as JSON.

```
{"pkcs10": "<the csr>",
  "template": "<the name of the template to use>",
  "config"  : {
   "dc": "<the hostname of the ADCS server>",
   "ca": "<the name of the CA instance>" }
}
```

When the ADCS Connector gets that, it will use the information to connect to ADCS and submit the certificate request via standard Microsoft DCOM RPC. ADCS will reply with a certificate request ID number. The connector will return that to Jamf Pro.

**Step 3:** Retrieve the certificate: https://adcsc.my.org/api/v1/certificate/retrieve

Once it has the request, the CA will sign the CSR. That can happen very fast or can take a bit if the CA is busy, or longer if a PKI admin has to approve certificates by hand. So after a pause, Jamf Pro will ask the Connector to check in and see if the CA has issued the cert for the given request number. This is an HTTP POST. The data is supplied as JSON.

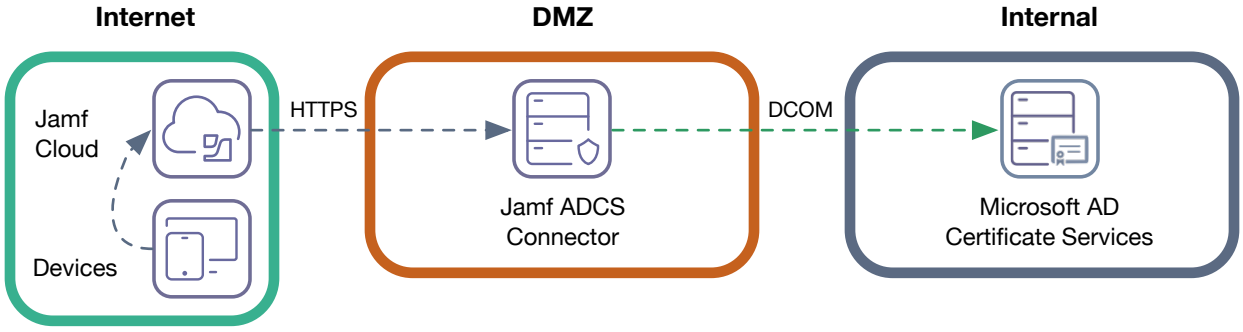
```
{"request-id": <the request id obtained in step #2>,
  "config": {
   "dc": "<the hostname of the ADCS server>",
   "ca": "<the name of the CA instance>"}
}
```

The ADCS Connector will create a DCOM message asking the CA for the completed certificate and the CA will return it if it is done. Otherwise it will return a "pending" status which will let Jamf Pro know that it should check back again later.

# Network Connection Examples

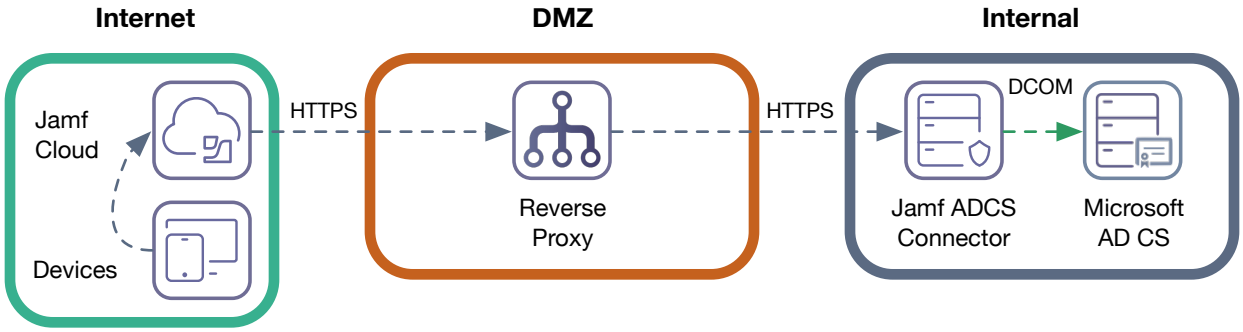
### Jamf Cloud with Jamf ADCS Connector in the DMZ

Here, the Jamf ADCS Connector is running in the DMZ, insulating ADCS from the internet.




### Jamf Cloud with a DMZ Reverse Proxy Layer

A reverse proxy may be used in the DMZ to reduce the ports open from DMZ to Internal or when a network environment does not allow DMX-based hosts to be bound to AD.



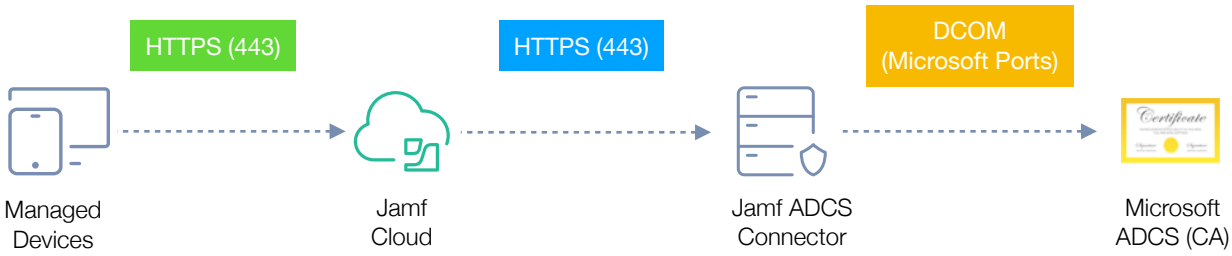
### Network Zones and Firewall Configuration

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 configuration profiles. |
| Jamf Pro to Jamf ADCS Connector | 443 | HTTPS | Jamf Pro sends certificate signing requests and retrieves completed certificates by opening a connection to the Jamf ADCS Connector, typically on TCP port 443, but any available IIS port can be used. |
| Jamf ADCS Connector to Microsoft ADCS | 135: MS DCE endpoint resolution used by DCOM.<br>49152-65535: Dynamic DCOM callback ports | Microsoft DCOM | The Jamf ADCS Connector uses Microsoft Distributed Component Object Model (DCOM) to communicate with ADCS. |



# 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's 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 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 it's 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 helps ensure 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 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. If you use a reverse proxy to cross a DMZ, also verify that you cannot reach the Connector's internal/local IP:port from internal/DMZ network zones. Nothing but the proxy should be able to reach the Connector.

# 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.

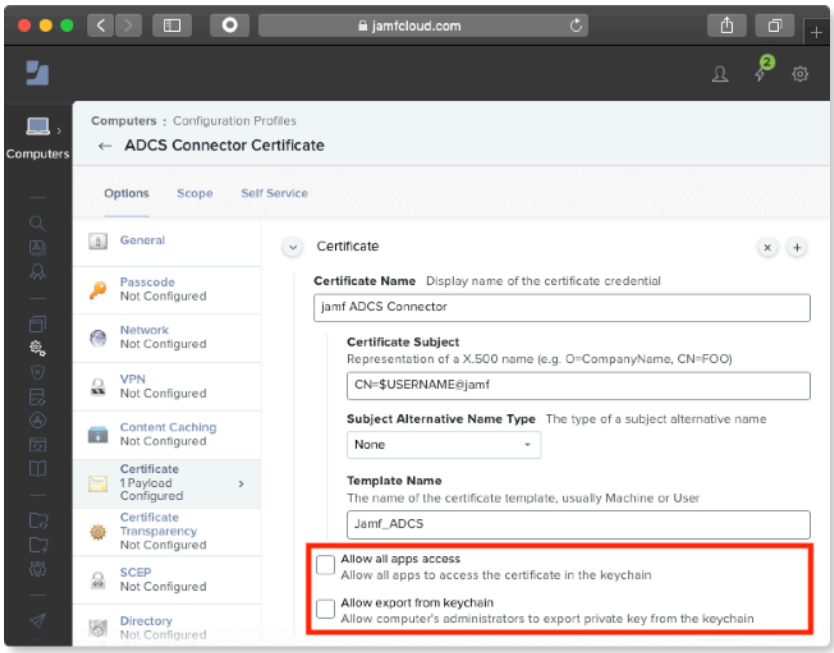
We have also previously noted that when using the Microsoft and SCEP (or SCEP Proxy) MDM payloads, the private key is generated on the managed device and never leaves. The device uses the private key to generate the CSR. With the ADCS Connector (Certificates) payload, the private key and CSR are generated by the Jamf Pro web application. Jamf Pro applies AES-256 encryption to any sensitive data at rest, including private keys and passwords fields. Additionally, Jamf Cloud databases are encrypted in their entirety in Amazon Web Services. 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 by the OS 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. Refer to Microsoft's documentation if you wish to reconfigure these ports.
- The DNS used by Jamf Pro will need to resolve the FQDN of your ADCS Connector hostname so if your Jamf Pro runs on Jamf Cloud, the FQDN of the Connector's external VIP will need to be available in public DNS. Alternately, if your organization has a custom Jamf Cloud environment, you may request host file entries. The use of an IP address for TLS communication has been depreciated. Public CAs and the newer Java libraries we use do not support certificates with IP address subjects.

### Microsoft DCOM Binding Requirement

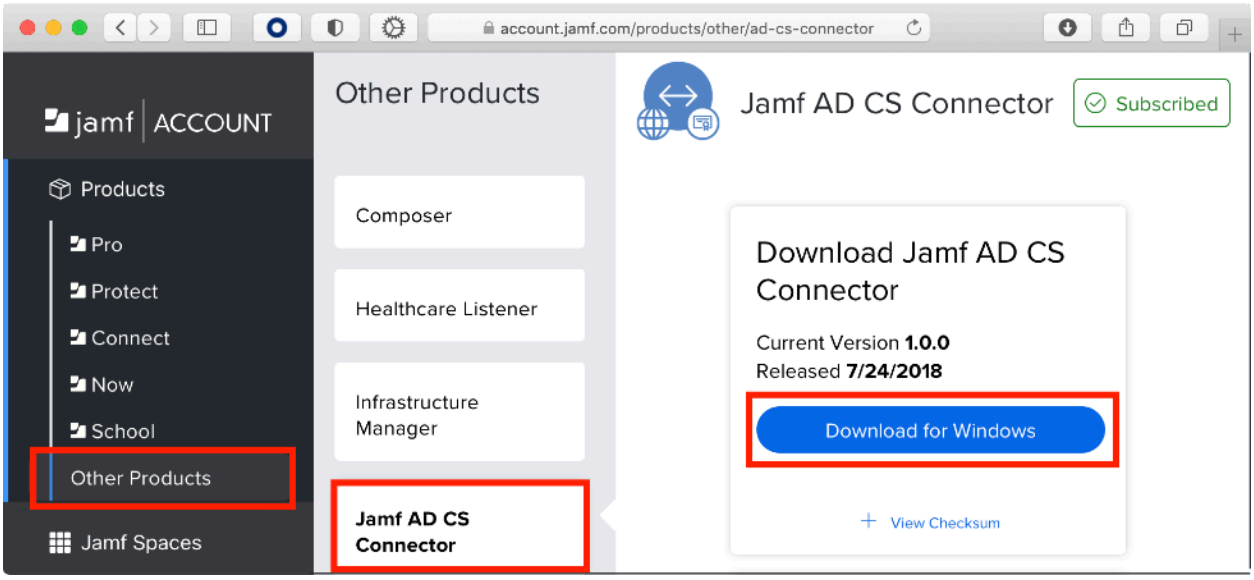
The standard installer will configure a thread pool to run the ADCS Connector in IIS. By default, IIS runs applications as "AppPoolIdentity". When using this identity, IIS will connect to ADCS using the ADCSC's host computer's system account, and this computer account will need to be given permissions 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 is typically bound either to the same domain as ADCS, or to a forest domain that has a trust relationship with the ADCS's domain.

The Connector can also be configured to run as another user, such as a domain service account.

# Step 1: Install the Jamf ADCS Connector

### a. Download a copy of the installer

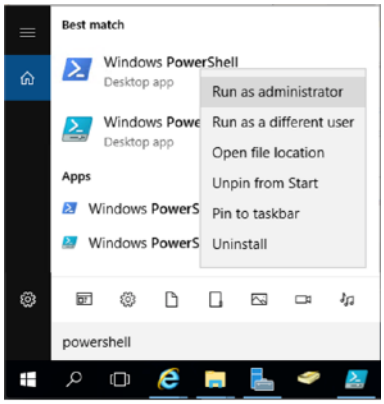
If you are a Jamf customer, the installation software is available under the "My Account" section when you log into jamf.com.




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

### b. 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 the Windows PowerShell command line as administrator, then cd into the installer and run the script...

```
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
```

For the -fqdn parameter, specify the host name Jamf Pro will use to reach 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 and also alongside the installer download.

The output will be similar to the following:

```
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
```

If you look in the script folder, you'll see that the script has saved a .p12 user certificate file and a server certificate public key file. We'll be uploading these to Jamf Pro when we configure the Connector there.

Hint: If there were no script errors but you still don't see the certificate export files in the installation script folder, you may have some unusual environment variable home directory settings or a GPO that directs file save locations to a specific spot. Look in C:\Users and inside the root of your user folders.

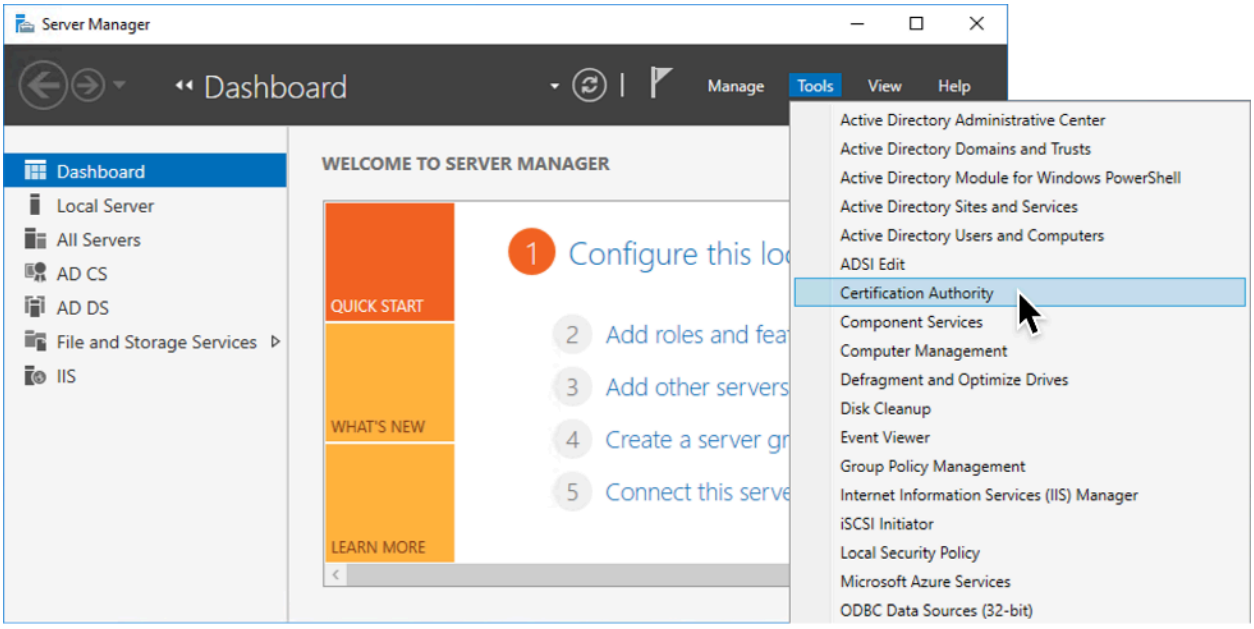
Make note of the `Client cert keystore password`. You'll be prompted to enter this password when importing the client identity file ("client-cert.pfx") to Jamf Pro. If you close the Powershell window before noting the password, you'll need to re-run the installer to get a new identity generated.

# Step 2: Give the Connector Permissions in ADCS

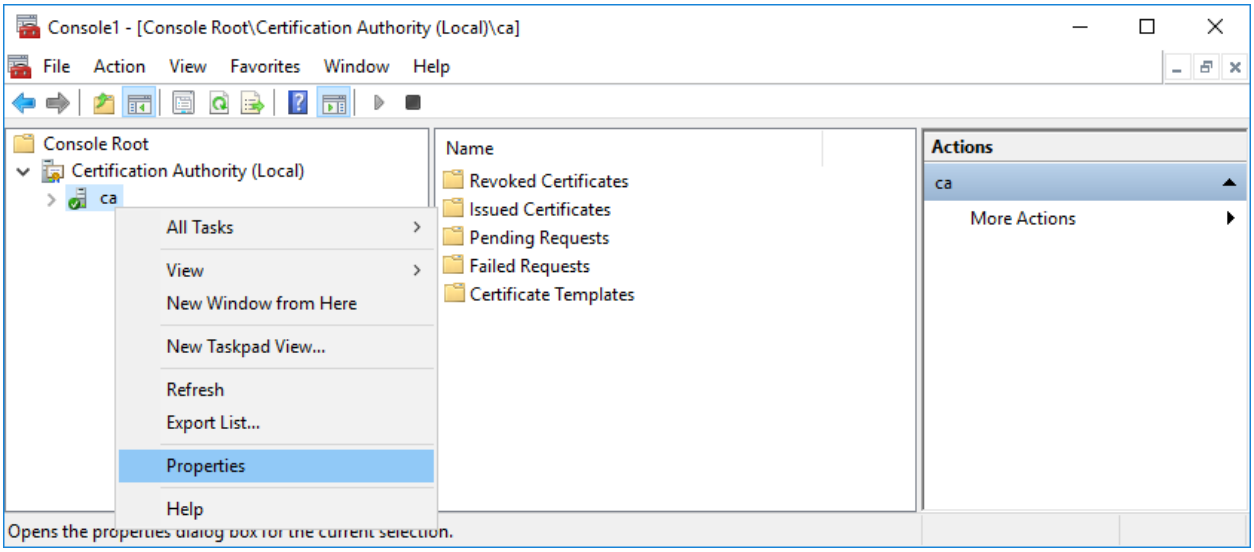
### a. Give ADCS Connector permission to talk to the CA

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.

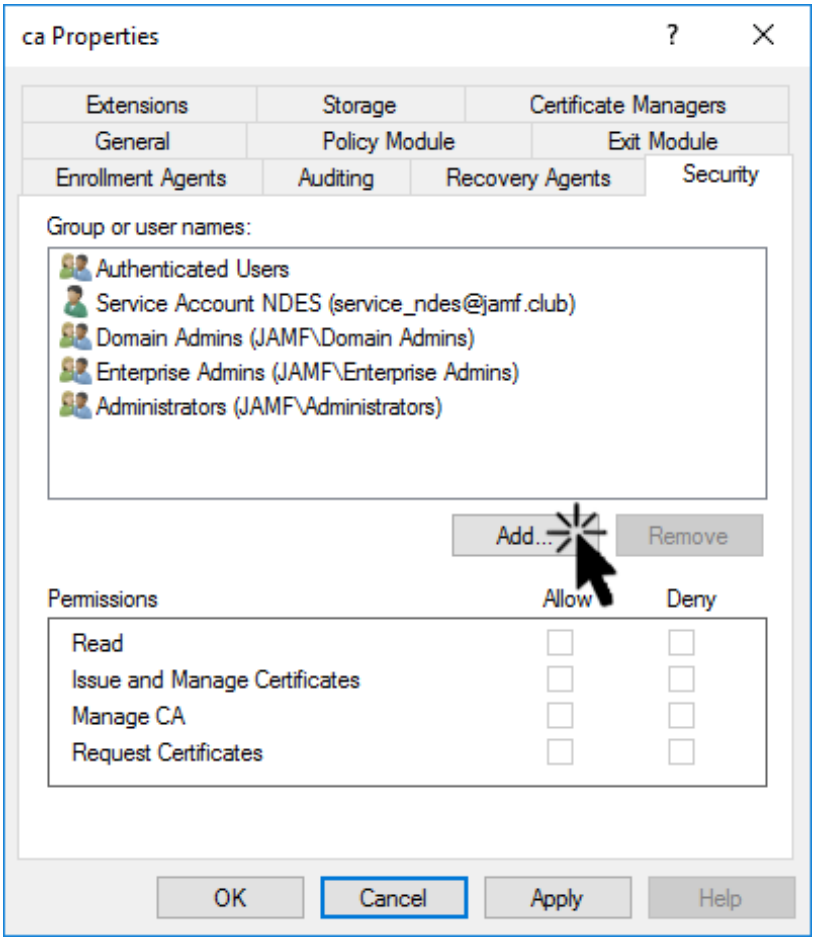
Run certsrv via cmd or as an MMC snap in. Or just select "Certification Authority" from Server Manger's Tools menu.



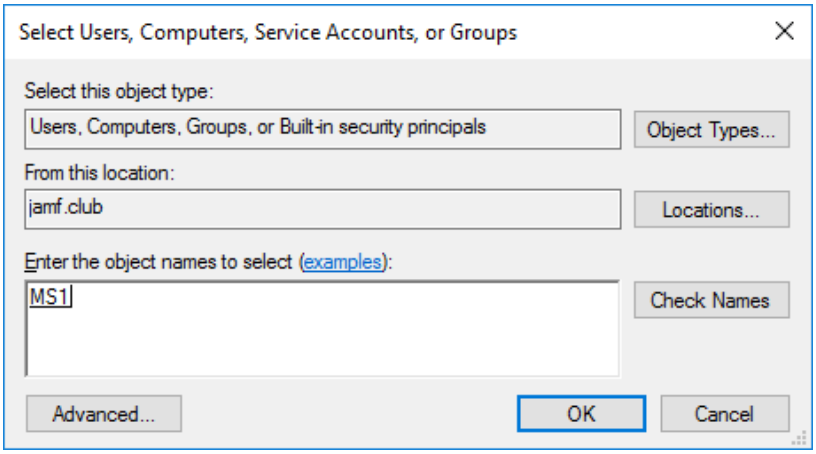

Right click on your CA's name and select "Properties...".



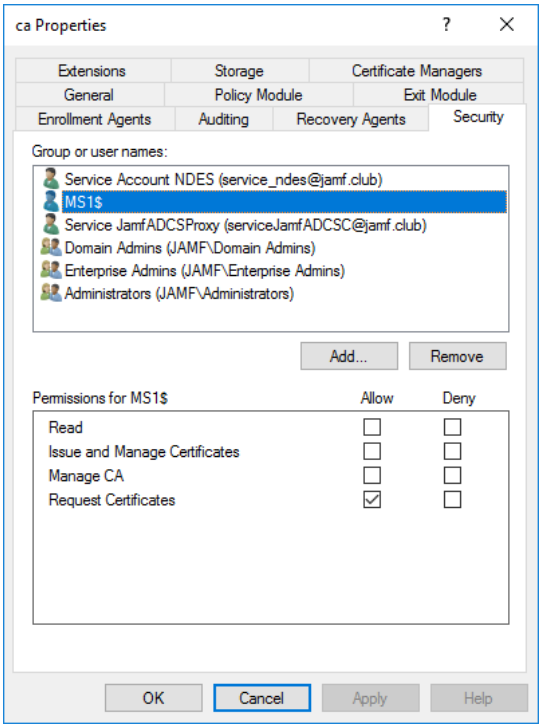
Switch to the "Security" tab and click the "Add..." button.




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 now 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.




### b. Selecting an ADCS Certificate Template

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 at least one certificate template. When we proxy certificate requests to ADCS, we do not need to create two separate templates to issue computer and username certificate subjects. You might however create two templates if you need multiple certificates with different usages.

Heads-up: Templates are usually configured with a certificate subject derived from the authentication accompanying the request, but proxied certificate requests like those from Microsoft NDES or the Jamf ADCS Connector are an exception. Proxy templates allows the requester to specify an arbitrary subject on behalf of a client they trust. Follow the upcoming instructions carefully so that only the Jamf ADCS Connector and other proxies you approve of can obtain certificates using templates that allow the requester to specify certificate subjects.

Certificate templates let PKI administrators define certificate attributes. For example, they can say who will be able to use the template, the key size, and set the "usages" -- things like

"Server Authentication" for a server certificate template or "Client Authentication" for the certificates we'll install on clients like computers and mobile devices.

The default installation of ADCS includes a variety of built in templates that can be used for different purposes. Most of these won't work with a certificate proxy like Microsoft NDES (SCEP) or the Jamf ADCS Connector because they always set the subject to the user or computer from which it receives the request, which in our case is going to be the Connector, not the managed devices themselves. For proxied certificates, we use a template that lets specifically authorized proxies specify certificate subjects.

#### b.1: Use a template you already use with NDES

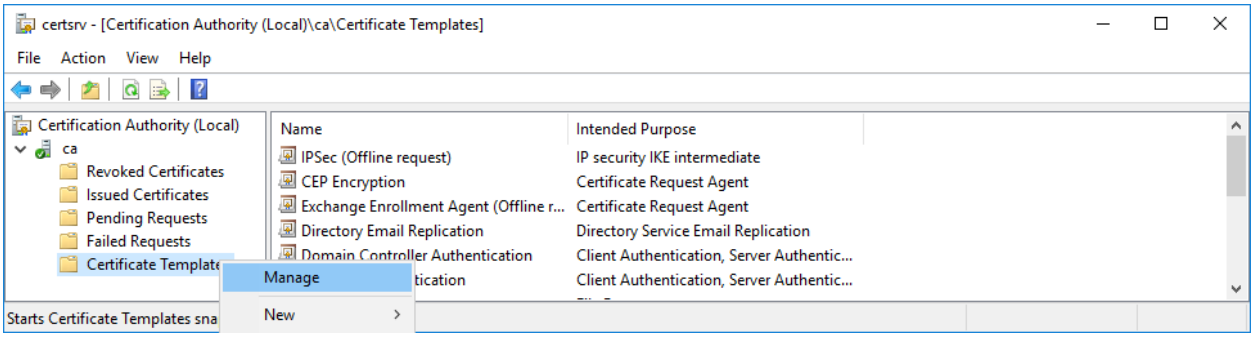
If you already have a template that you use with the Microsoft NDES role to obtain the types of certificates your managed devices need, you can use the same one with the Jamf ADCS connector because NDES templates allow the requester to specify certificate subjects. The only thing you'll need to do is add the ADCS Connector host to its permissions and grant the "enroll" right. Alternately, you can duplicate the template, remove the NDES server's rights, and add the Connector's.

#### b.2: Creating a New Certificate Template in ADCS and Granting Template Permissions

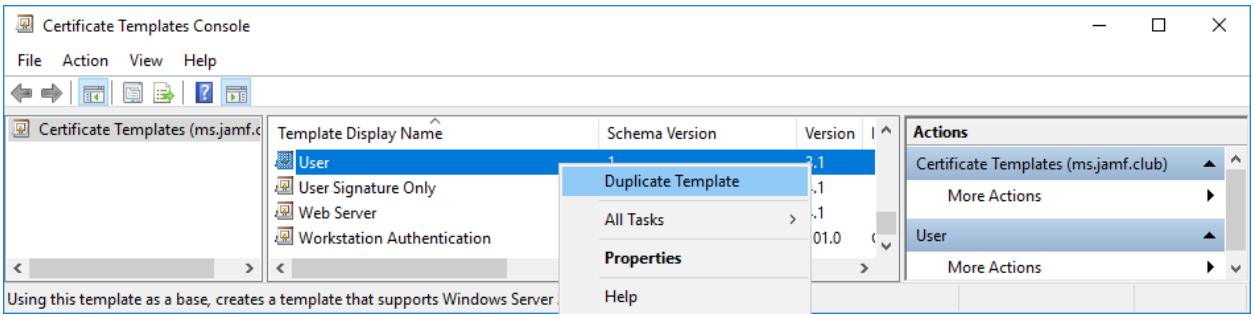
Your existing CA probably already has a template that's being used to deploy certificates to bound computers and servers, like the template you use for Windows device certificate auto-enrollment. But if you used that template with the ADCS Connector, every certificate would have the name of the Connector host as its subject because as far as ADCS knows, that's what is requesting the certificate. We need to duplicate that existing, pre-configured template and set the duplicate up so the proxy can specify the subject when we make certificate requests. That's what allows the cert to have the computer name or username we need as our certificate subjects.

Using a separate template dedicated to 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.

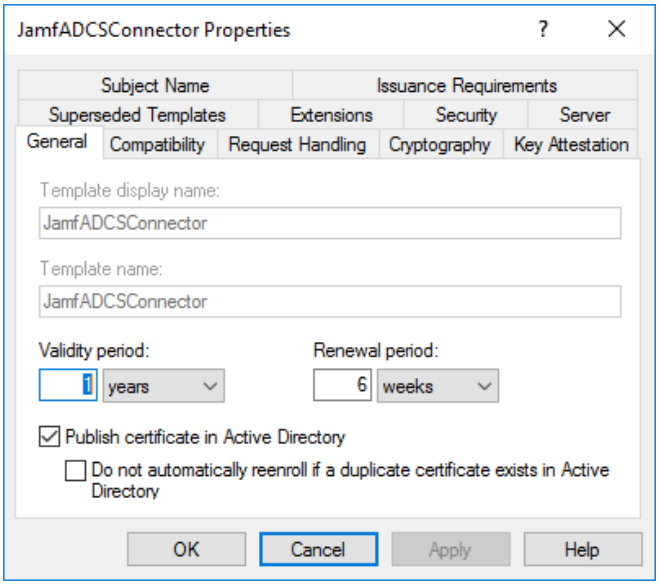
Certificate templates are managed using the Certificate Templates Console. In certsrv, right-click on "Certificate Templates" and select "Manage" from the contextual menu to run the template console. You can also access Certificate Templates Console as mmc snap-in or by running certtmpl.msc directly.




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 and select "Duplicate Template" from the contextual menu.

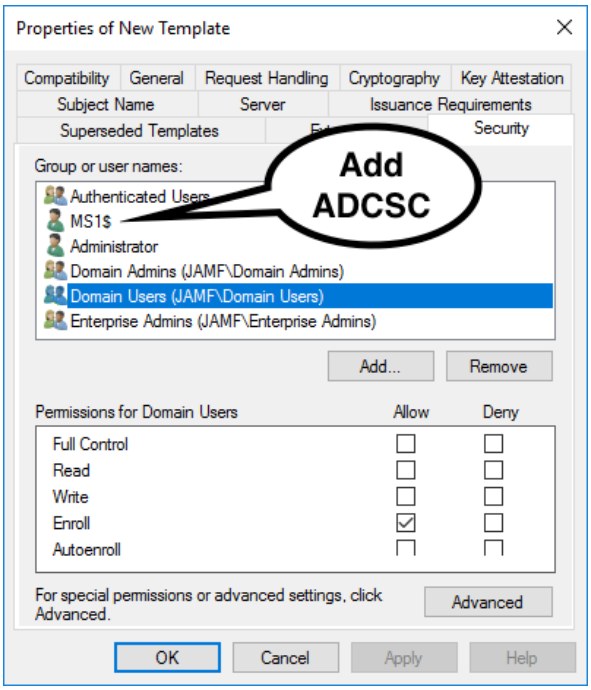


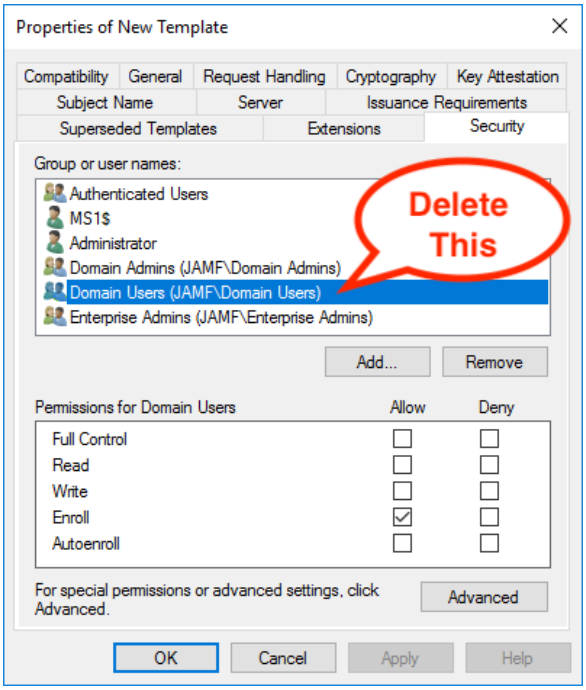

A certificate properties editor dialog will appear. In the "General" tab, give it a name consistent with whatever naming standard your CA admin prefers. Make a careful note of the name or paste it into an email/document. You'll to enter this exact name when you're configuring Jamf Pro to work with the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



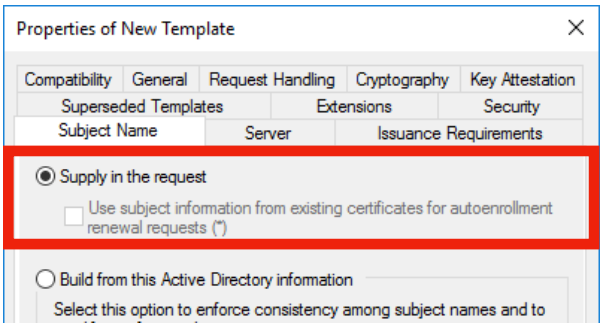
In the "Security" tab, add the ADCS Connector host, just as you did at the CA-level and grant the "Enroll" permission.

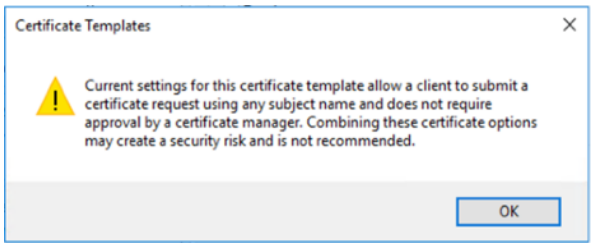
Next, select "Authenticated Users" and verify that it has Read permission and does **not** have the enroll permission. If "Domain Users" or "Domain Computers" are in the list, **delete them**. It wouldn't hurt if they were there and had only read access, but you don't want to risk making a mistake and 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! [1]




In the "Subject Name" tab, allow the subject 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 make sure only the ADCS Connector has enroll permissions on the template.

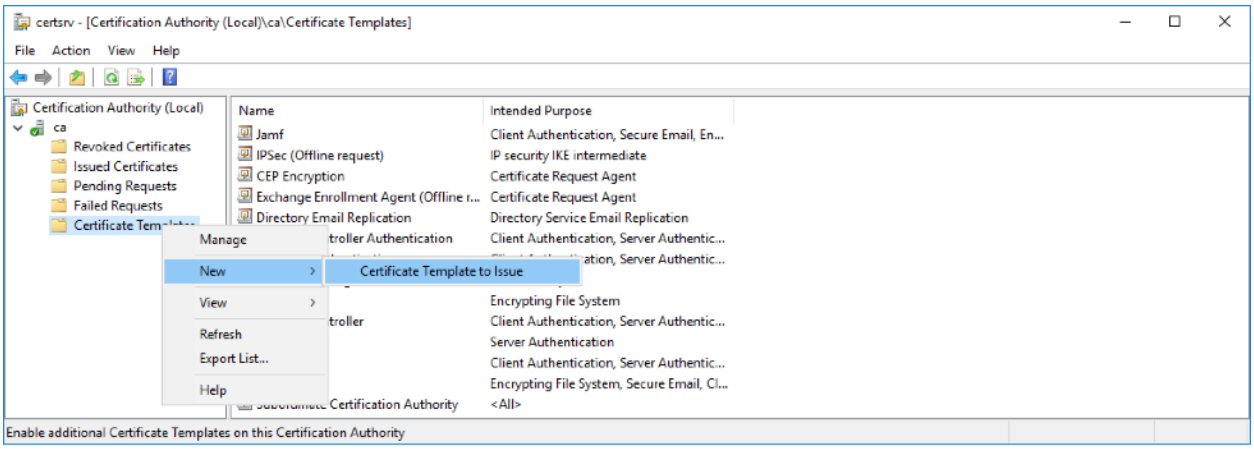


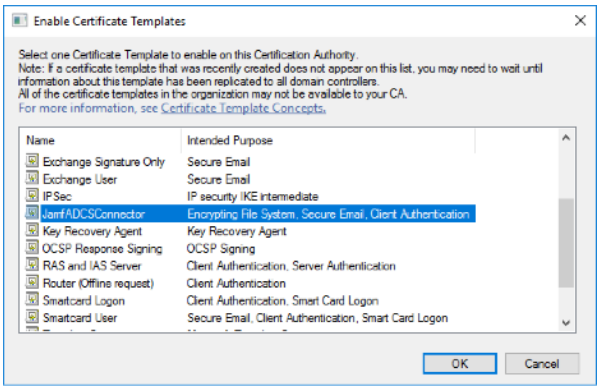

Click "OK" to exit the setup.

[1] https://itpro.outsidesys.com/2018/03/21/adcs-manage-pki-certificate-templates/ offers a good discussion of this topic.

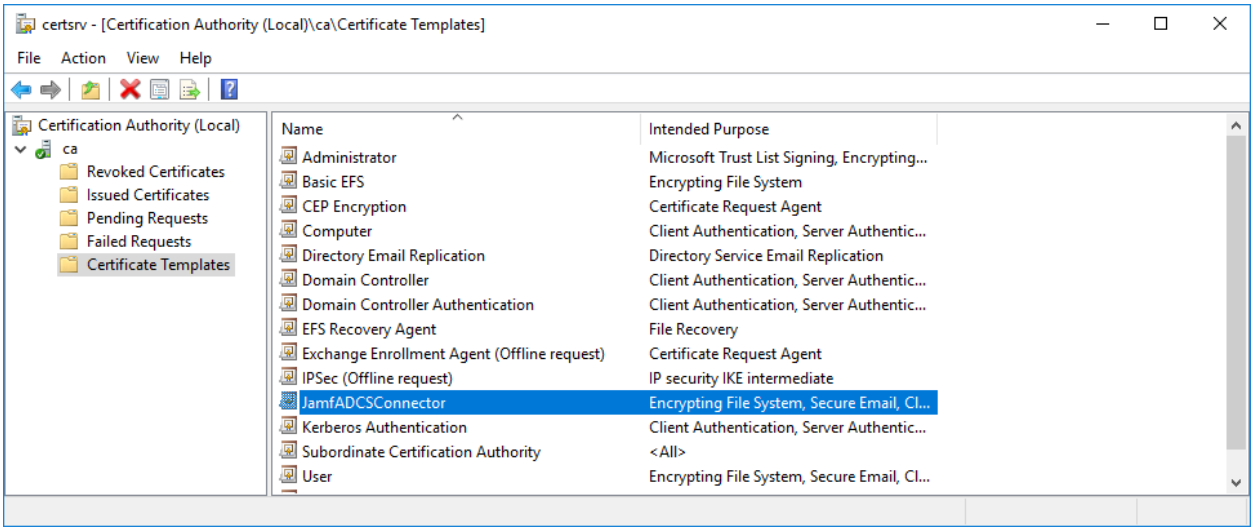
Now that we have created the certificate template, we need to publish it to the CA. Go back to the certsrv Certificate Authority console and right-click on "Certificate Templates" and select "New>Certificate Templates to Issue".




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




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



### Artifacts

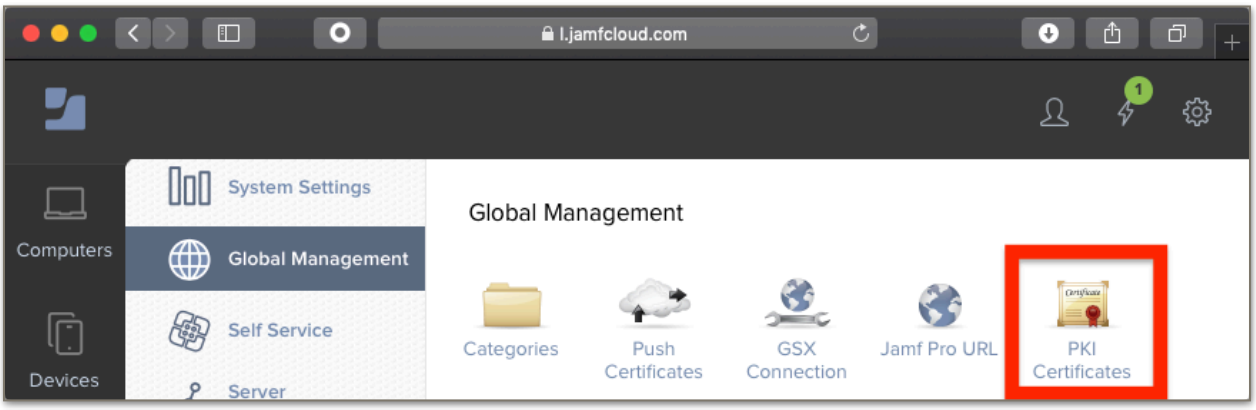
You have now configured external DNS and VIP routing/firewall rules so Jamf Pro can speak with the AD CS Connector, installed the Connector software, given the Connector permissions on the CA, and configured one of more certificate templates.  You now have the following ready for configuration in Jamf Pro:

1) The adcs-proxy-ca.cer file, the Connector's public TLS certificate
2) The client-cert.pfx file, the keypair Jamf Pro will use to authenticate itself to IIS
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.)

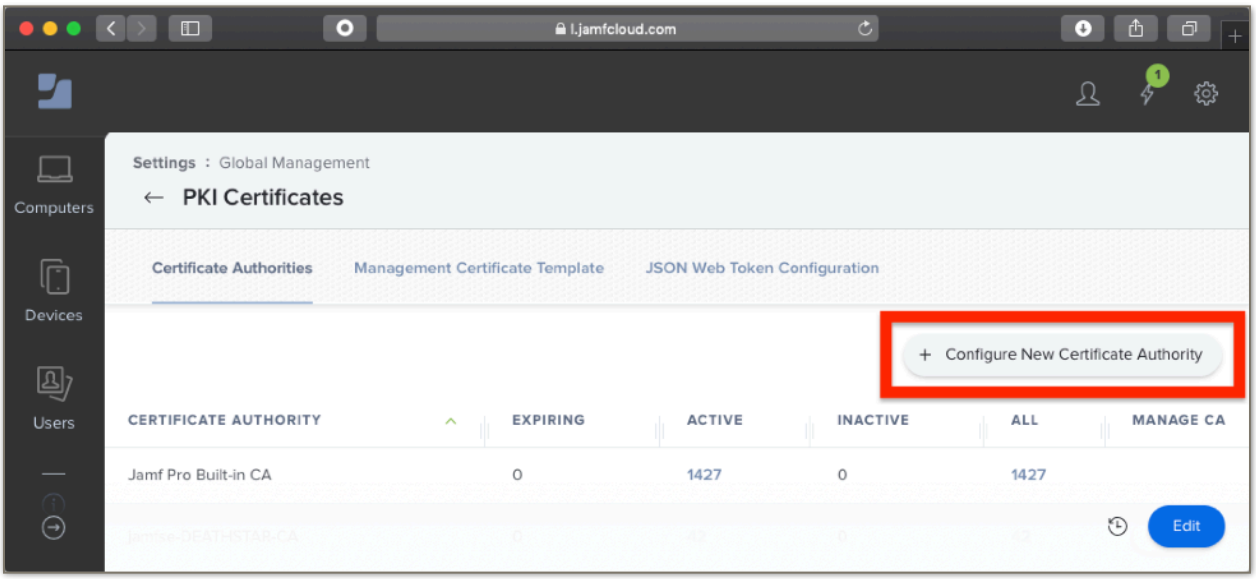
## Step 3: Configure the ADCS Connector in Jamf Pro

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.

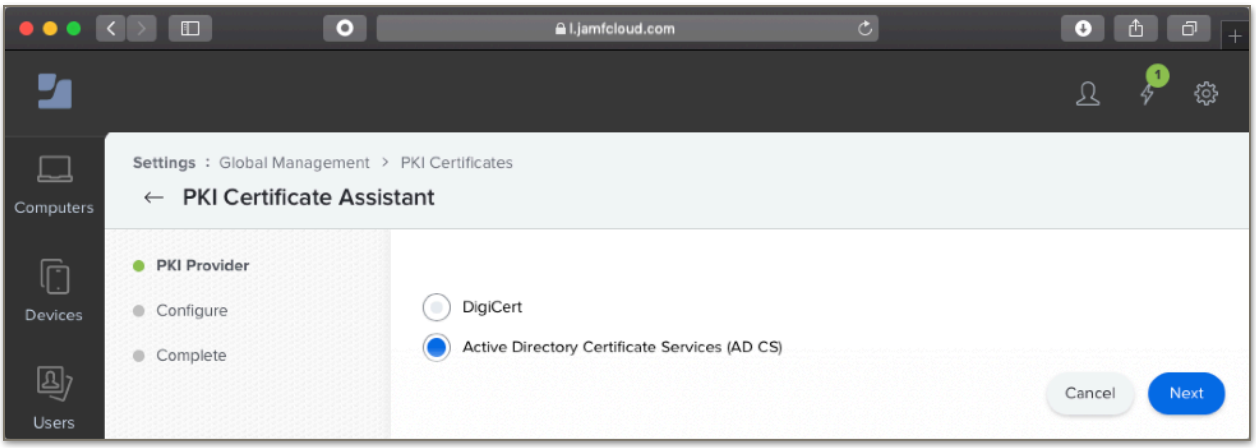
1. Login to Jamf Pro
2. Click the gearbox in the upper right corner to access the settings page
3. Go to Global Settings
4. Click PKI Certificates



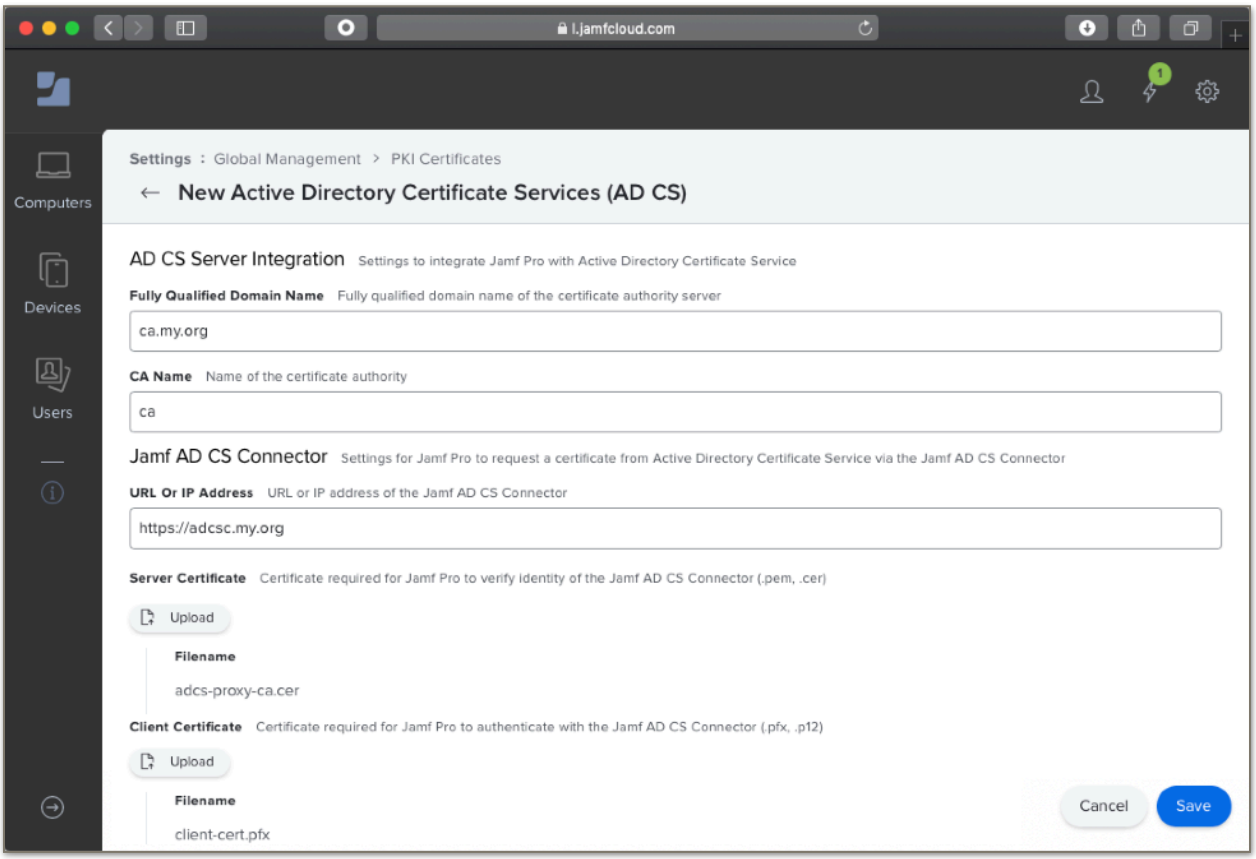

Click "Configure New Certification Authority".



Select the "Active Directory Certificate Services (ADCS)" option.




In the settings page, "Fully Qualified Domain Name" is the hostname of the Microsoft CA, the CA name is what you see when you look at it in certsrv, the URL of the ADCS Connector is the external DNS hostname Jamf Pro will use to reach the Connector) 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 login to IIS).




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 ADCS Connector proxy service.

# Step 4: Set up a Certificate Payload

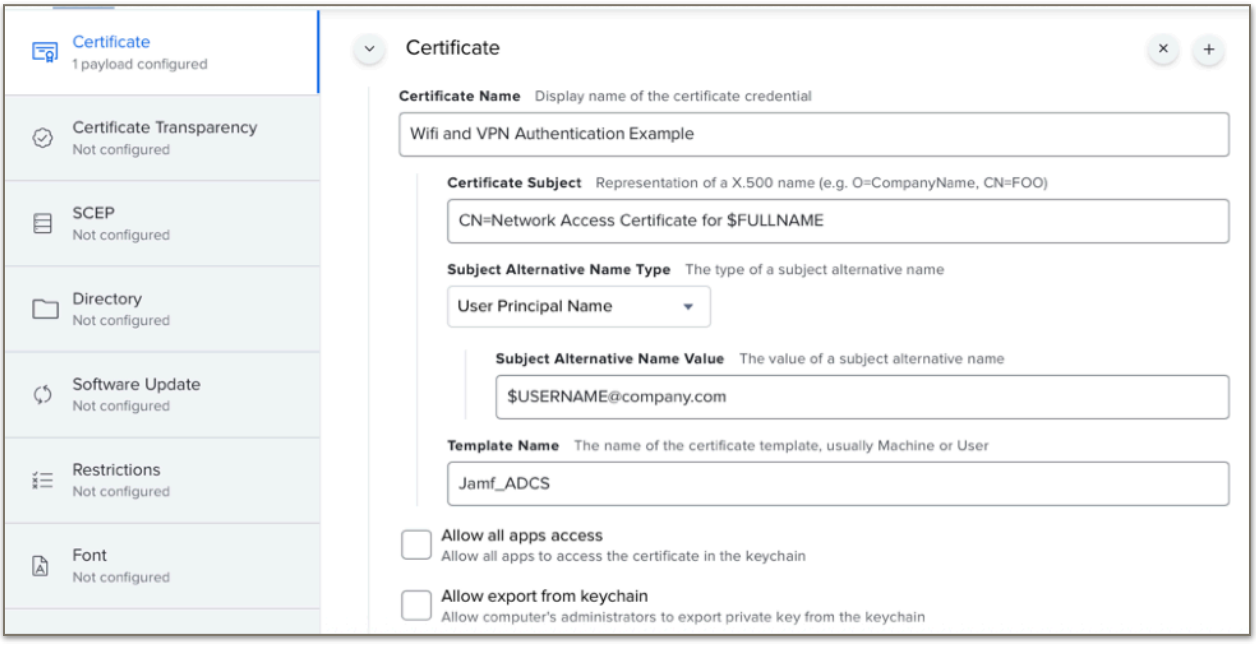
**Step by Step...**

The final step is to make a Computer and Mobile Device Profile with a certificates payload.

1. Go into Computers or Mobile Devices in Jamf Pro
2. Click **New**.
3. Use the General payload to configure basic settings, including the level at which to apply the profile and the distribution method.
4. Select the **Certificate** payload and click **Configure**.
5. Enter a display name that describes the purpose of the certificate (e.g., "Wifi Authentication")
6. Choose an ADCS instance from the **Select Certificate Option** pop-up menu.
7. Use the settings on the pane to specify the desired certificate subject(s), AD CS Template, accessibility, and exportability.
    - Certificate subject and Subject Alternative Name (“SAN”) are set based on the the requirements of the service to which the certificates will be used to authenticate
    - The subject must be specified in X.500 (“CN=”) format.
    - If you already have an ADCS template that is being used to authenticate to a service, you can replicate the Subject and Subject Alternative Name (SAN) used there.
    - Microsoft NPS and some other common RADIUS network authentication services require that certificates referencing a user account include a User Principal Name (“UPN”) SAN while certificates that reference a computer account use a SAN in Fully Qualified Domain Name (“FQDN”) format.
    - You can combine Jamf Pro replacement variables with static text to customize the subject. Lists of available variable names can be found in the following sections of the Jamf Pro Administrator's Guide:
        - Computer Configuration Profiles
        - Mobile Device Configuration Profiles
    - Enter the name (Not the Display Name) of the ADCS template you want to use. The template will determine certificate usages.
    - Do not check the “Allow all apps access” check box if your certificate is only used by Apple OS services like network and/or VPN connections.
    - Do not check the “Allow export from keychain” if you want to prevent certificates being transferred to other devices.

### Example

An admin wants managed devices to authenticate to a WiFi network via RADIUS. The administrator enters the following settings:




In the above example, we used "$USERNAME@company.com" to replicate the user's UserPrincipalName AD attribute, but we could also map the UPN attribute into Jamf Pro's directory mappings or create a directory extension attribute and used that as the variable. If the organization's email addresses and UPNs are always the same, we could have just entered "$email".

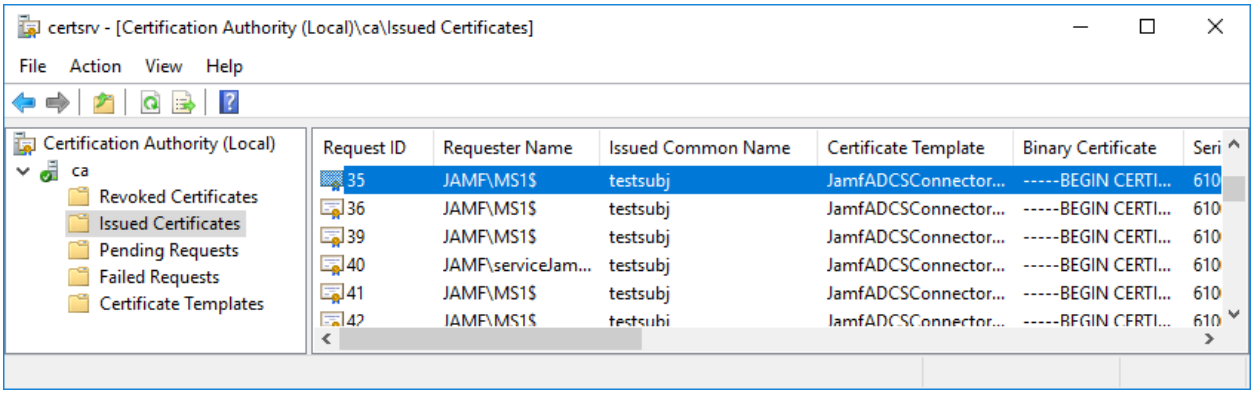
You can look at the Users and Locations tab in Jamf Pro device inventory pages to see how directory values are being mapped. You can use Microsoft's "Active Directory Users and Computers" program to view directory attribute names and values to find out what you need to map into jamf Pro, or, you can view attribute names and values in PowerShell when troubleshooting unexpected certificate subjects.

```
PS C:\> get-aduser jan.doe -Properties DistinguishedName, ObjectClass, SamAccountName,
UserPrincipalName, emailAddress

DistinguishedName : CN=Jan Doe,OU=IT,DC=company,DC=local
EmailAddress      : jan.doe@company.com
Enabled           : True
GivenName         : Jan
Name              : Jan Doe
ObjectClass       : user
SamAccountName    : jan.doe
UserPrincipalName : jan.doe@company.local
```

### Testing

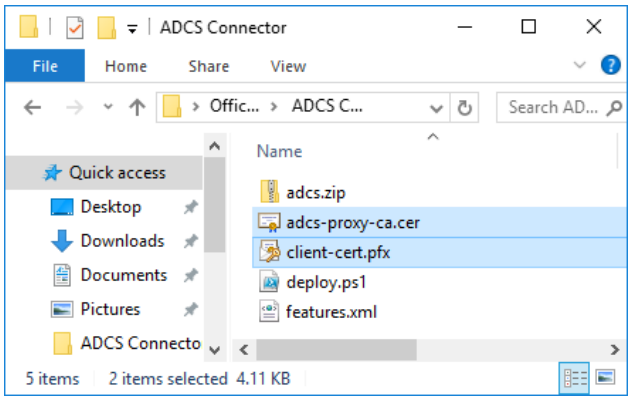
Scope your profile to a test device and select automatic delivery or Self Service as the profile delivery mechanism. It may take a few minutes for a certificate request to complete depending on APNs and ADCS processing, but sometimes it seems nearly instantaneous. You can observe the profile delivery in Jamf Pro, see it appear in the profiles list on devices, and, on Mac, look at the details of the delivered device identity certificate in Keychain Access. You can also observe certificate requests in the Issued Certificates list in ADCS' certsrv console.




Once you see identity certificates flowing to managed devices, you can upload the root and intermediate trust certificates needed for your CA by adding additional certificate payloads to the profile you're building. This is needed so your device trusts its identity certificate. You can also add a network and/or VPN payload and specify that the identity certificate should be used to authenticate to those services.

# Appendix: What you should see after running the ADCS Connector Installer

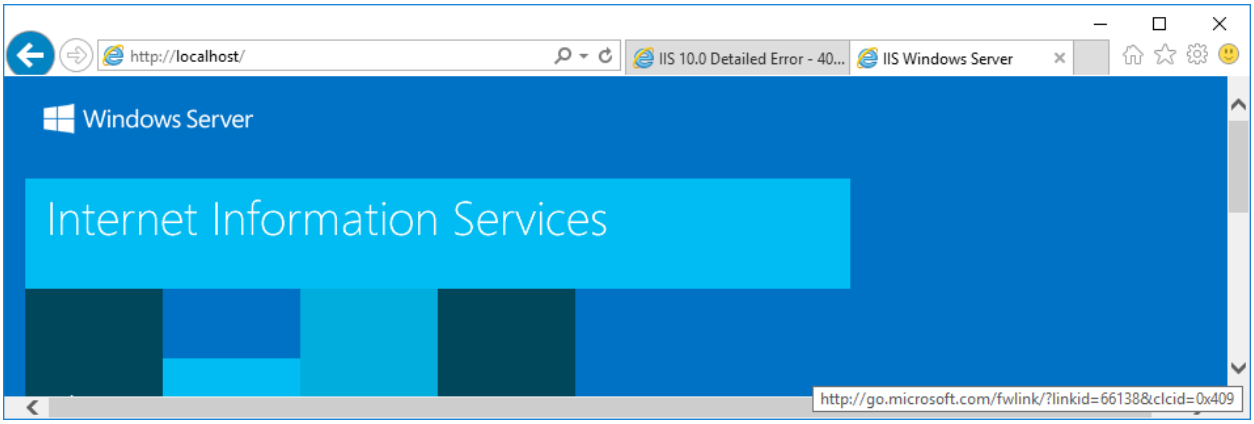
After the installation completes, there will be two new files in the working directory.



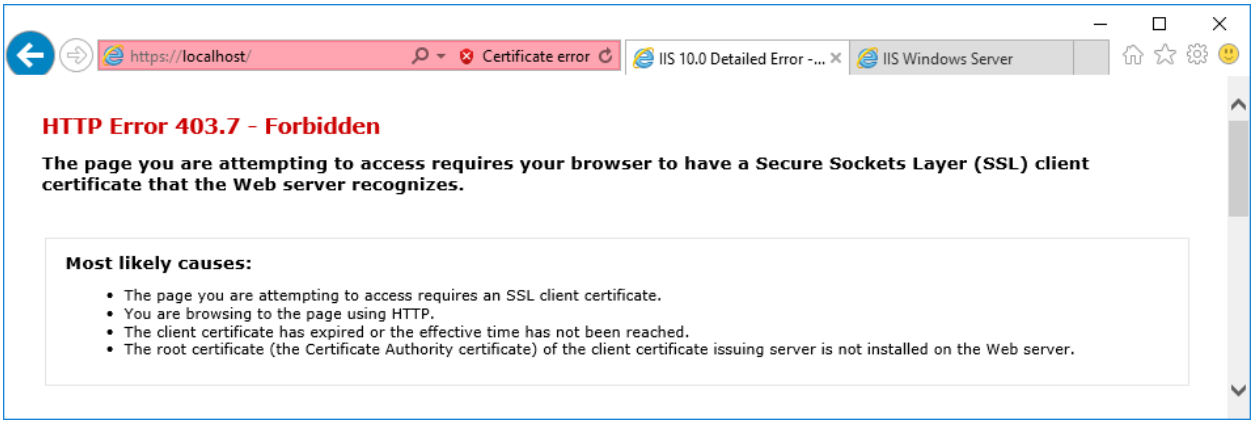

**adcs-proxy-ca.cer** is the public key for the identity that IIS will use when negotiating TLS when Jamf Pro tries to connect. If the server uses an identity that doesn't match up with this public key, Jamf Pro will not trust the server and the TLS handshake will fail.

Even more importantly, the ADCS Connector needs to know that the connections are really coming from your Jamf Pro instance. 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.

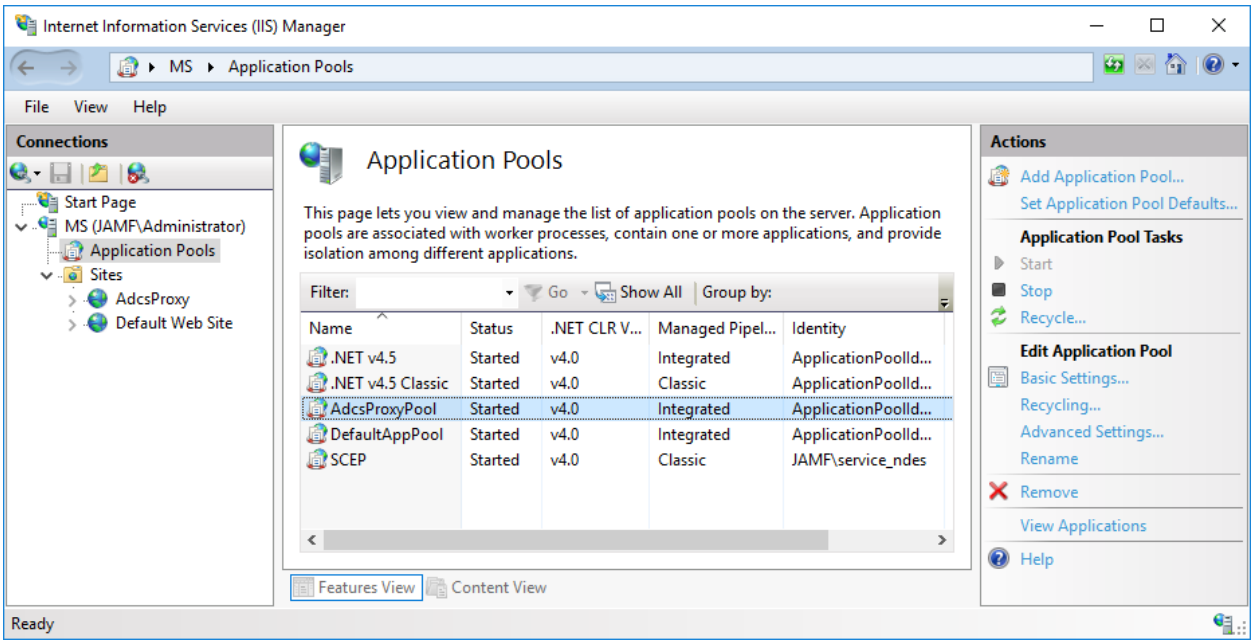
If we launch a web browser on the Connector host, we can see that IIS has been installed.



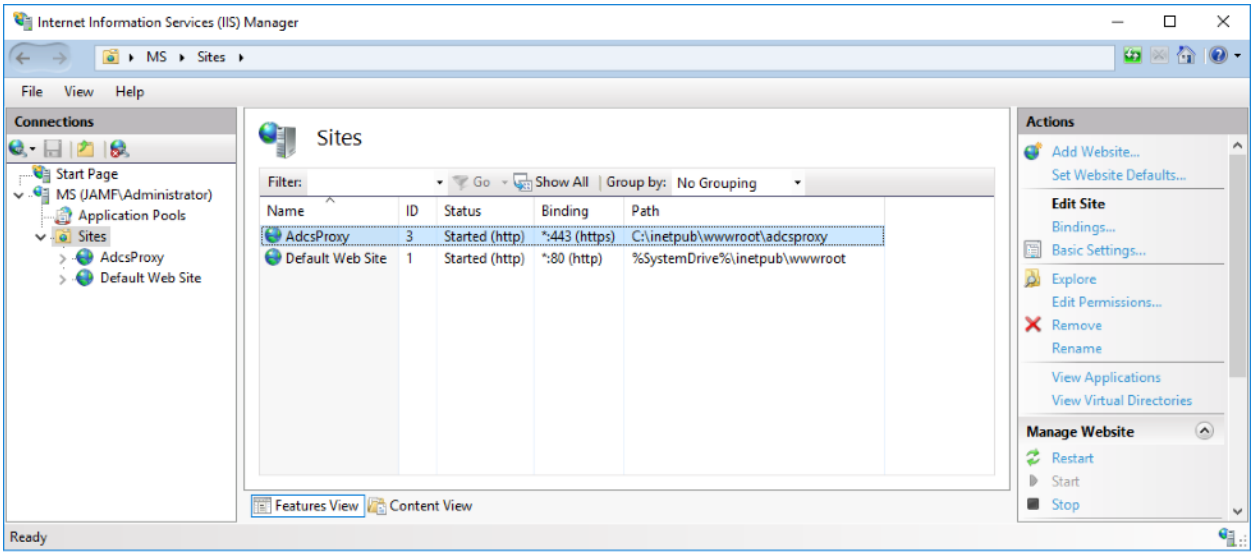
If we attempt to browse to the Connector and accept the self-signed server certificate warning, we will get an authentication error, showing that anonymous auth is disabled.



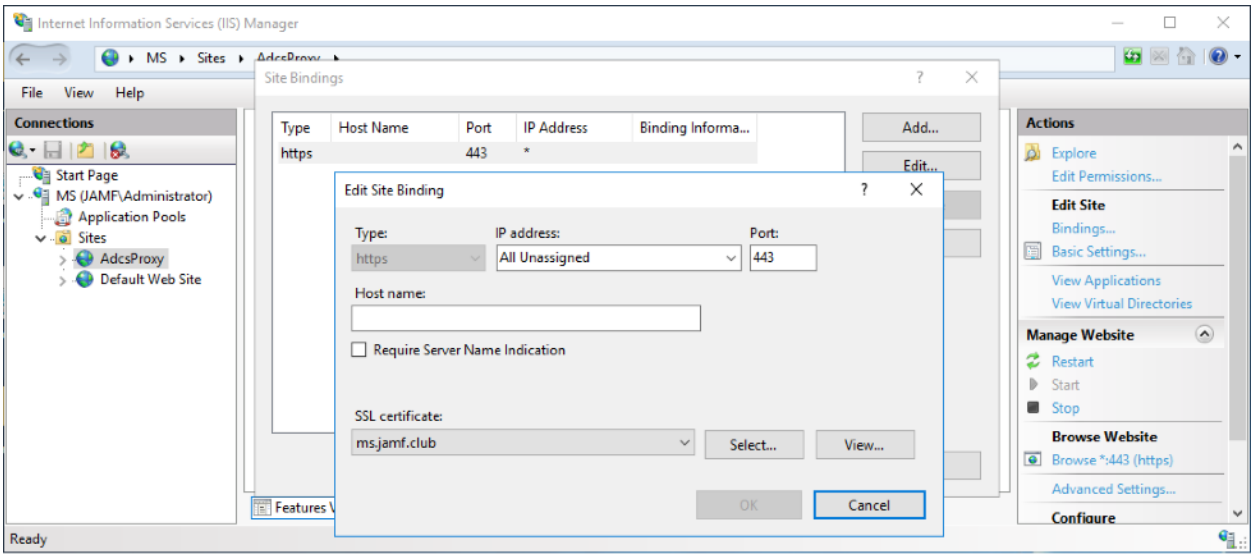

In IIS Manger, we observe that a new application pool has been created for the ADCS Connector. The ADCS Connector site will run within this pool. We see that the Connector is running as "ApplicationPoolIdentity", an identity derived from the local computer's bind to the domain. This is why we gave the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



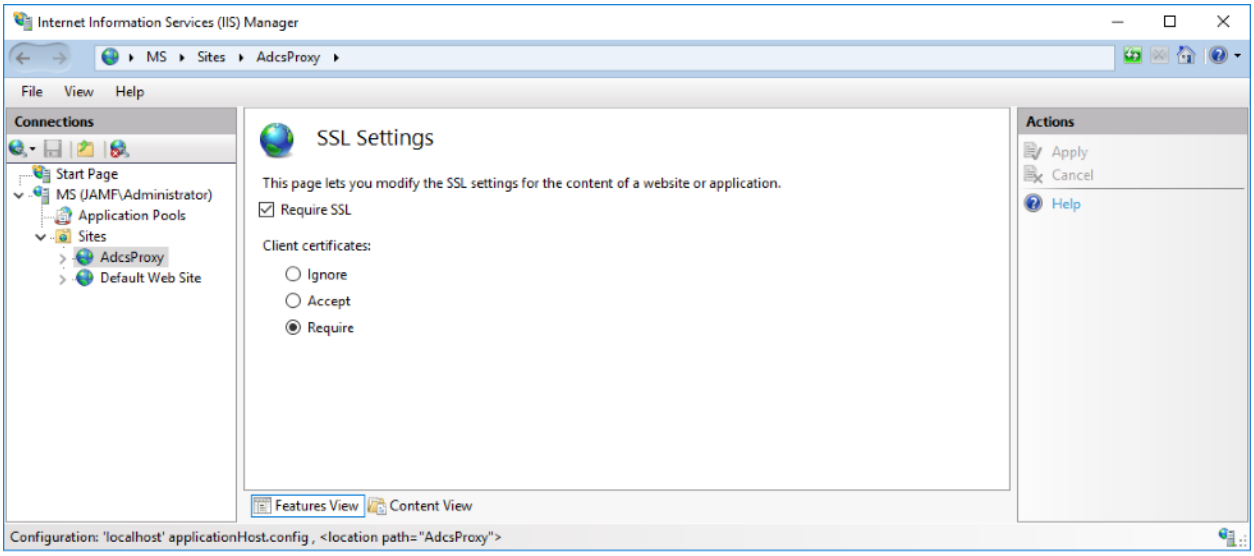
Under Sites, we'll see the corresponding site. It's listening for https on port 443.



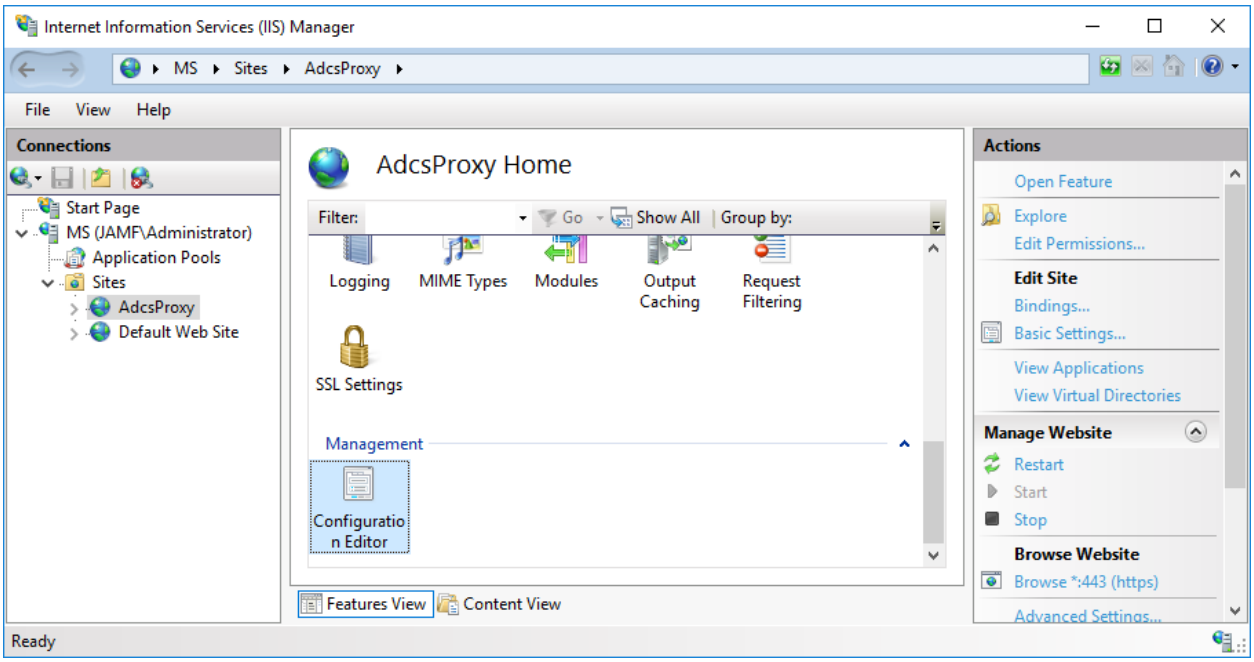

In bindings, we see that the TLS certificate subject matches the FQDN we will tell Jamf to resolve when it connects.



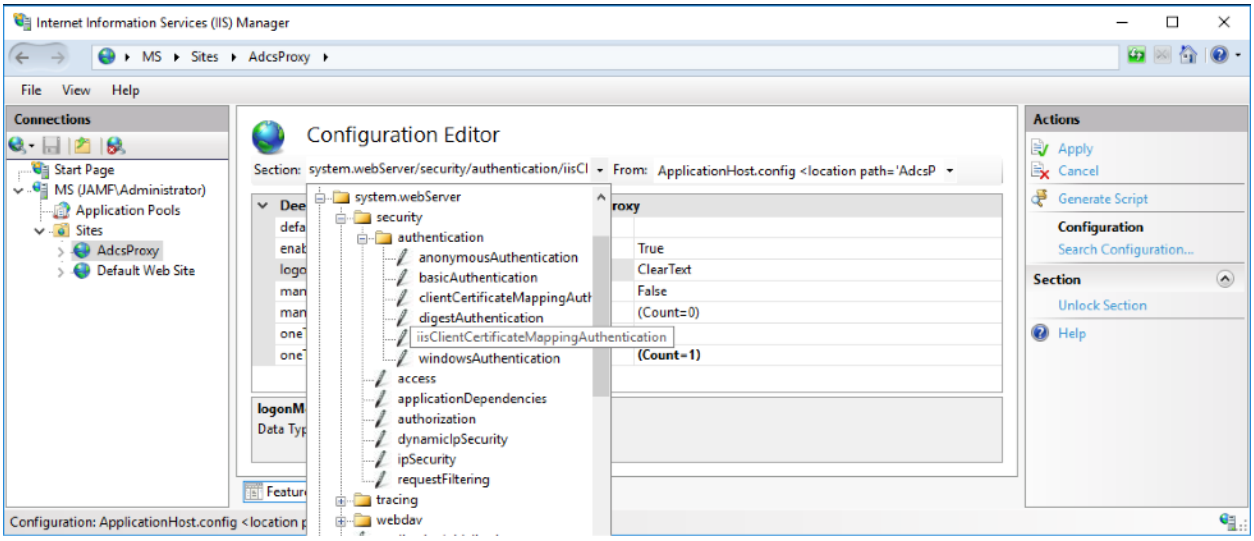
Under SSL Settings, we see that IIS will require SSL connections and that connecting computers must present a client certificate for authentication.



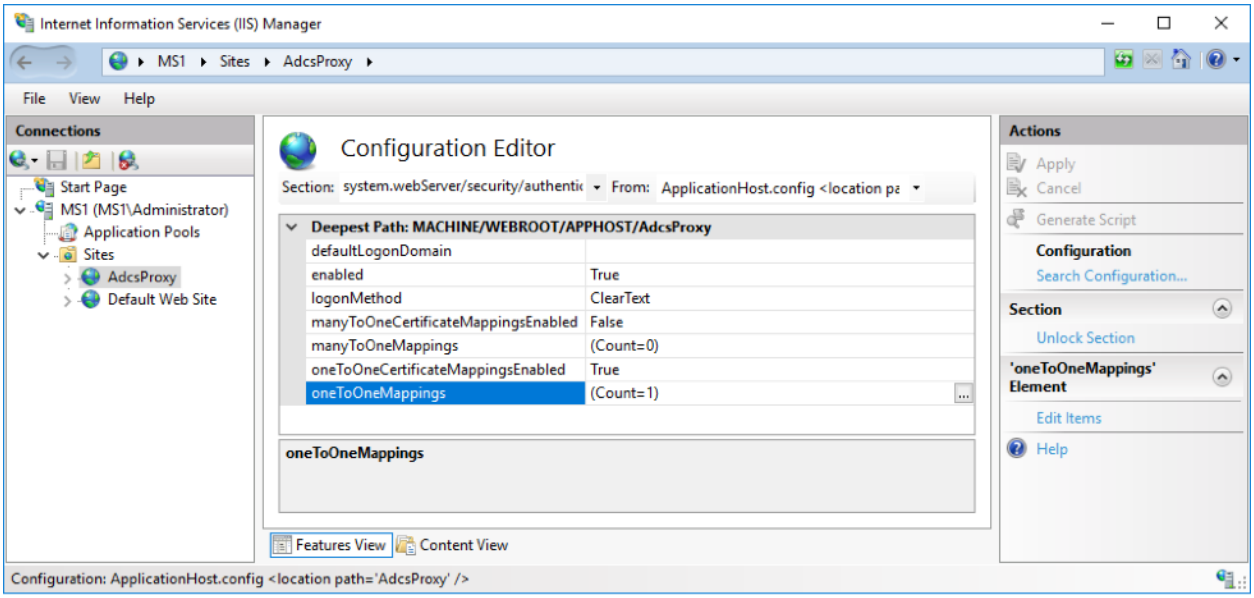

To review client certificate authentication settings, go to Configuration Editor.



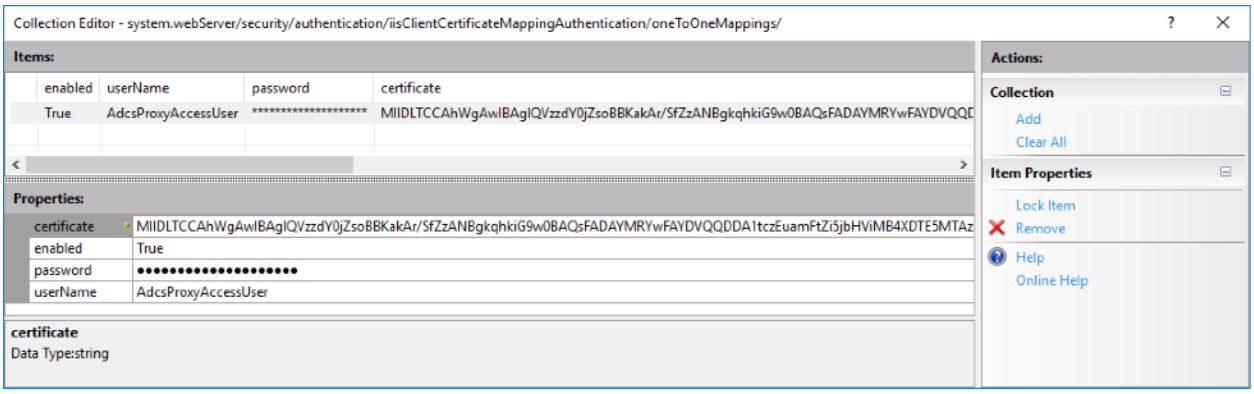
Navigate to  "system.webServer > security > authentication > iisClientCertificateMappingAauthentication" in the "Section" drop-down menu.




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.




**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.

# 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 certificates?

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

By default, ADCS Connector runs inside IIS's default system thread pool, so when it tries to authenticate to an ADCS service, it will be using the identity of the host where the Connector is running. That means the ADCS Connector host needs to have rights to the ADCS CA, and, in an enterprise CA configuration, 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.

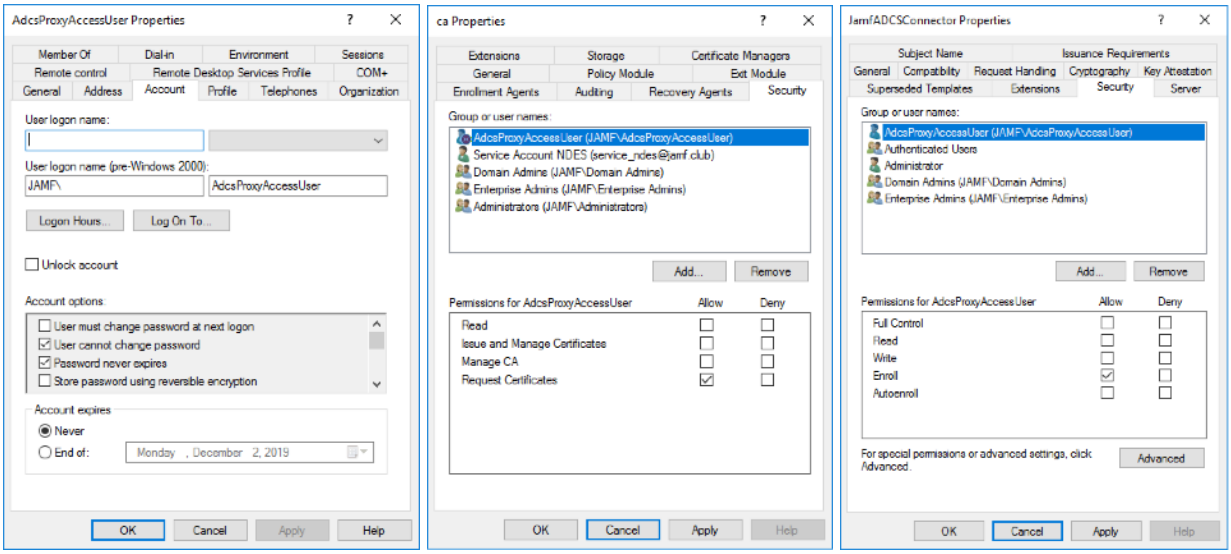
https://docs.microsoft.com/en-us/iis/manage/configuring-security/ensure-security-isolation-for-web-sites is a useful reference. The standard installer follows these guidelines in that it runs the Connector's IIS site within the standard app pool. However, if a domain user service account is required,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

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

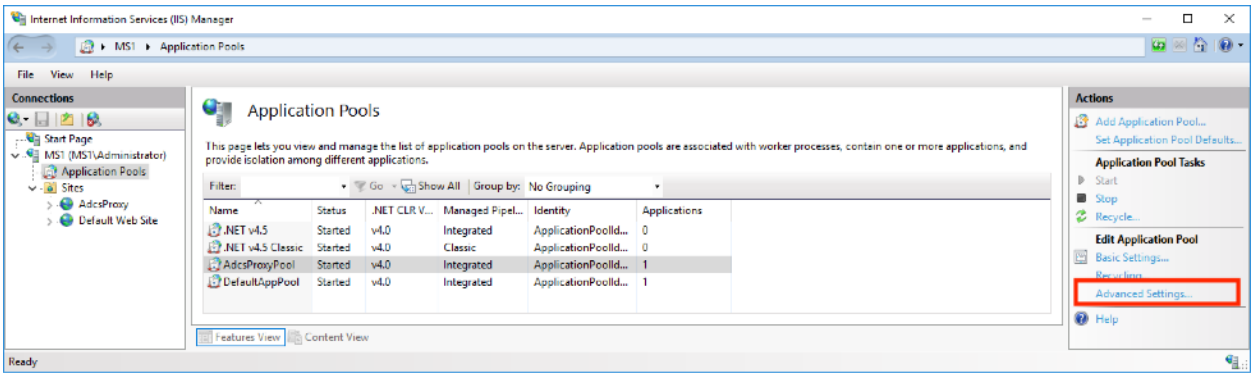
Service accounts are typically configured with "User cannot change password" and "Password never expires". In the screenshots below, the user name is "AdcsProxyAccessUser". Be sure the password is complex -- *this account will have rights in ADCS so we need to be certain that it is well secured.*

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.

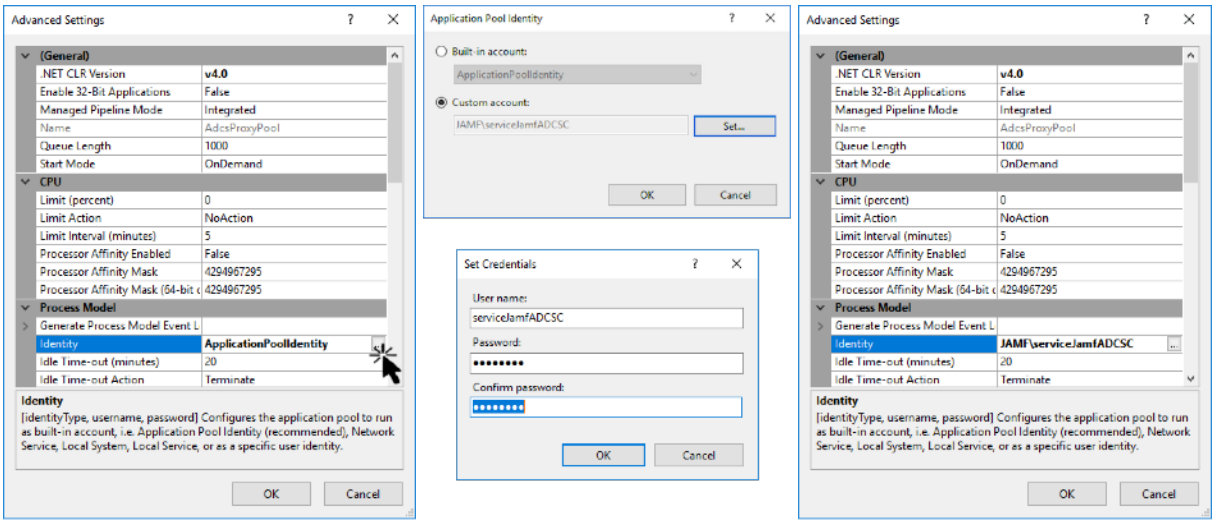
If you had previously granted these permissions to the ADCSC host, you can remove those after adding the new service account.




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".




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.

# Configuring IIS to use an alternate Server Certificate

A server certificate is presented to HTTPS clients when they try to connect. The certificate proves that the Server is who it says it is. The Connector installer instructs the Windows OS running on the Connector host to generates a self-signed certificate for this purpose. Your organization may have a rule forbids the use of self signed server certificates. Instead, server certificates must come from a third-party publicly-trusted CA or from the enterprise's CA.

There's a very good reason for this... we don't want users installing untrusted certificates into their browser or OS keystores/keychains when they load a web page from a browser. It's an open invitation to a man-in-the-middle attack. But the ADCS Connector is not a user-facing web app and it has only one client -- Jamf Pro, and Jamf Pro does not care what CA generated the server certificate. It only cares that the Server certificate is an exact match for the server public key you uploaded when you configured the PKI settings in Jamf Pro.

There is no increase in security gained by configuring a new server certificate in IIS. You should already be configuring the Windows firewall on the Connector server to only accept traffic originating from Jamf Pro (or the proxy/load balancer that's delivering Jamf Pro's traffic). That will prevent your server from showing up in InfoSec scans and raising false alarms.

You should, of course, explain the nature of this service to your security team. They may understand exactly what's going on and agree to leave the default IIS setup in place, but sometimes they just don't have the bandwidth to deal with exceptions or want to have to explain it to the auditors when the time comes. In this case, you will want to use a server certificate generated by your own internal PKI or from a third party public CA as dictated by your enterprise standards.

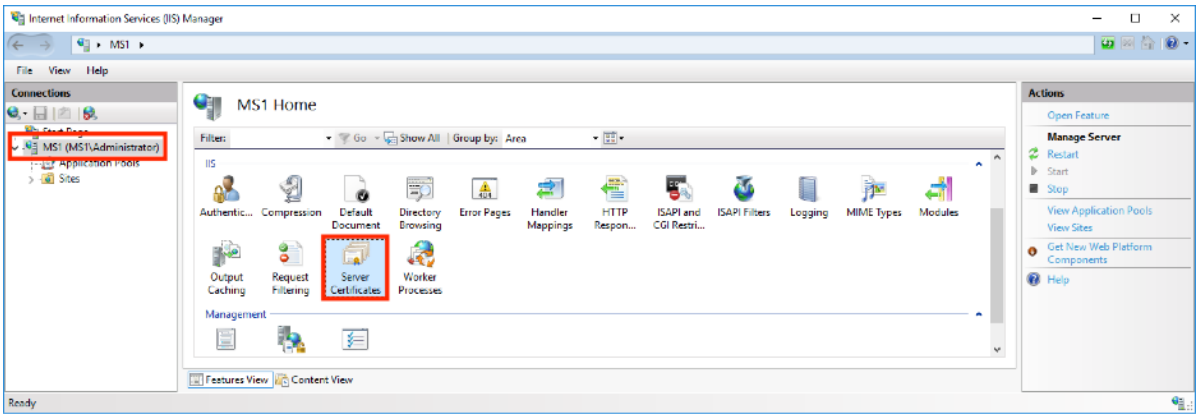
The following basic steps are used to install a server identity for IIS:

1) Run the ADCSC deploy script.
2) Obtain a new server certificate 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.
5) Export the public key for your server certificate and upload it to the Connectors PKI settings entry in Jamf Pro.

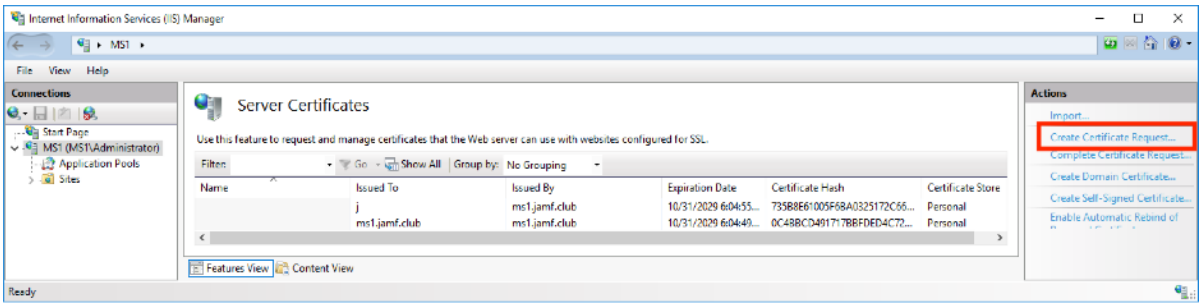
### Obtaining a Certificate Signing Request

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 the one built into IIS.

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




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



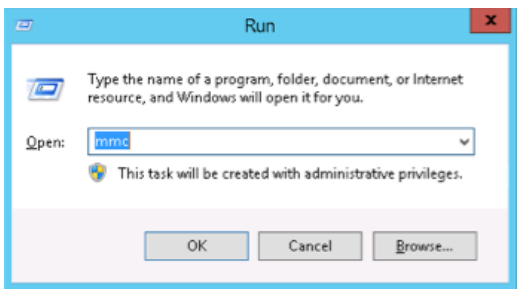

3) The Wizard will walk you through the rest of the process for configuring and saving the CSR. The Common Name (CN) will be the host name that Jamf Pro connects. In the case of Jamf Cloud configurations, this is the external DNS ("VIP") that resolves to your external IP address. Use the default Microsoft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

### Configuring IIS to Use the Alternate Server Identity

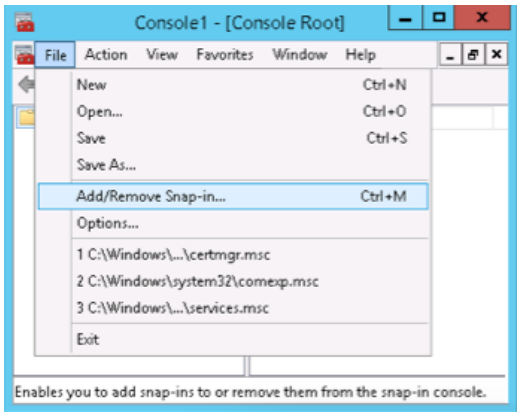
Once you already have the identity .pfx file that you want to use as the ADCSC site's SSL server certificate, it's an easy two-step process to install it. We'll add it to the Windows certificate store using the Certificates mmc snap-in, then tell IIS to use it to secure our site.

First, upload the identity file to the Windows keystore:

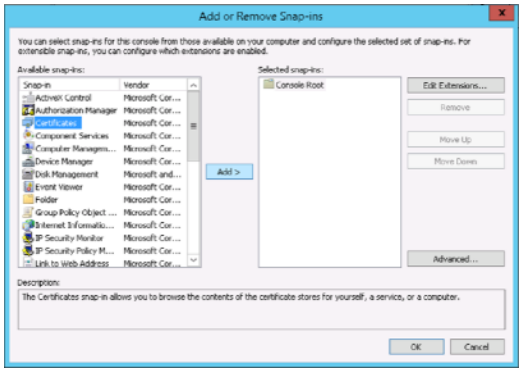
1. On the Start menu click Run and then type mmc



2. Select File > Add/Remove Snap-in

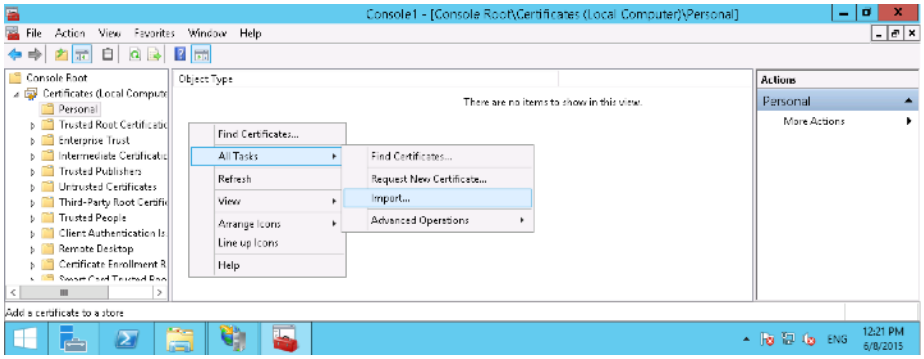



3. Click Certificates > Add

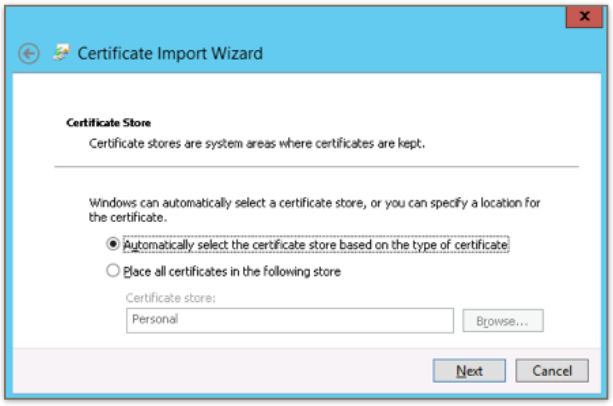



4. Select Computer Account and then click Next.


5. Select Local Computer and then click Finish. Then close out of the "add snap-in" window.
6. 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.

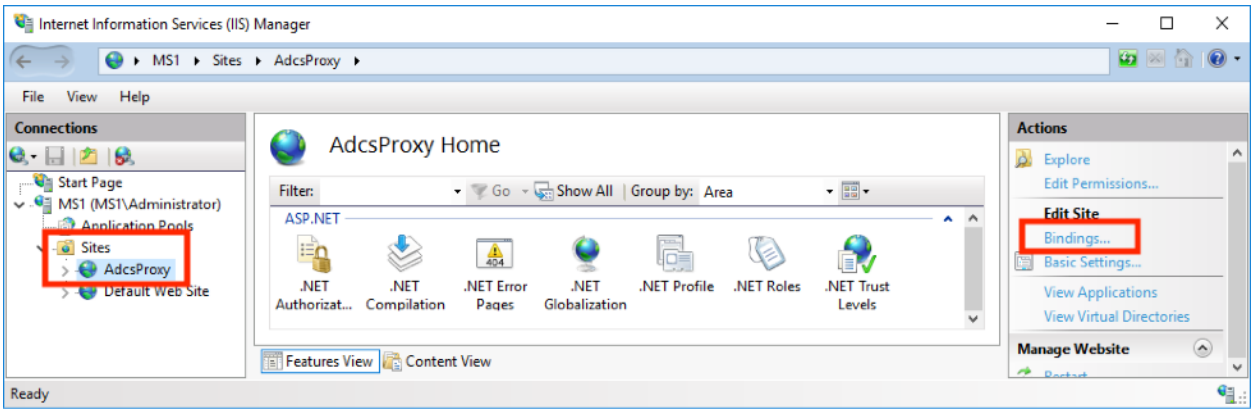


7. Follow the certificate import wizard to import your primary certificate from the .pfx file. When prompted, choose to "automatically place the certificates in the certificate stores based on the type of the certificate". The advantage of that choice is that the wizard will correctly distribute the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

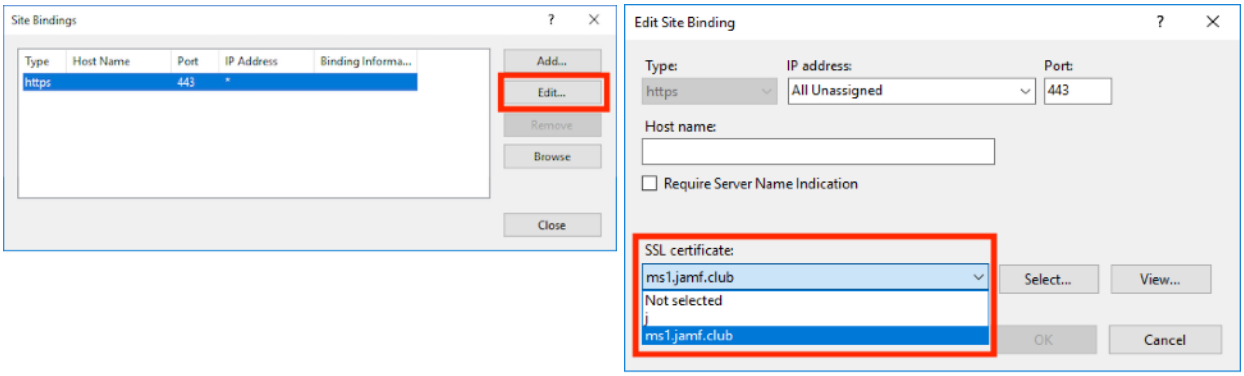



If your identity needs root or intermediate certificates for its source CA trust chain that were not included in the .pfx file, make sure they're already installed on your windows server. If not, you'll need to add them to the Windows keystore using the same procedure.

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.

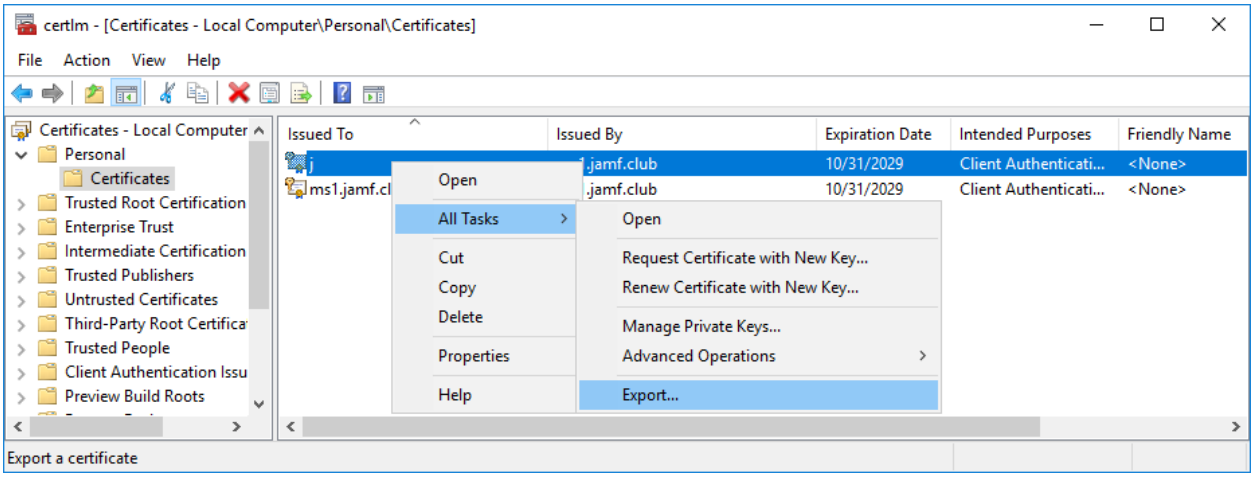



### Replacing a server certificate in IIS prior to expiration

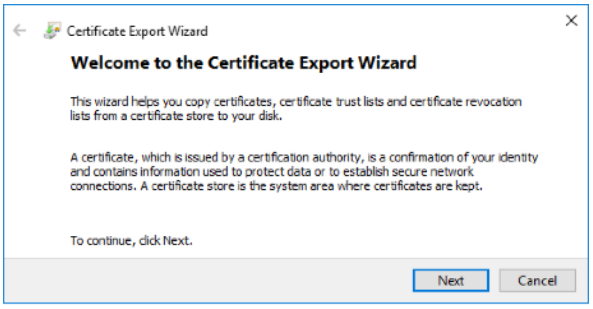
You should replace your IIS server certificate prior to expiration. If you don't, Jamf Pro will no longer be able to negotiate TLS connections once the expiration date has passed. The steps to follow are the same as the initial installation. You can install a new certificate any time you want and it doesn't matter if it's an update of the 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's certificate.

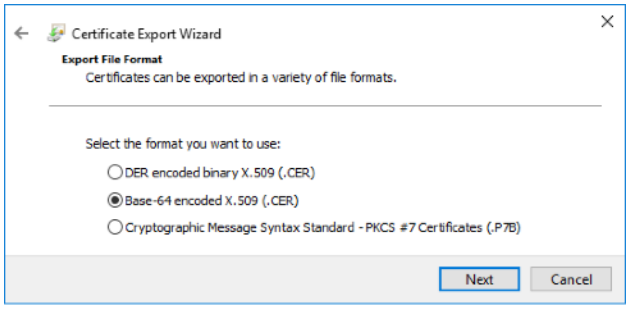
# Configuring IIS to use an alternate Client Certificate

If you have another identity file (.pfx or .p12) that you want to use to authenticate Jamf Pro to ISS, 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...".

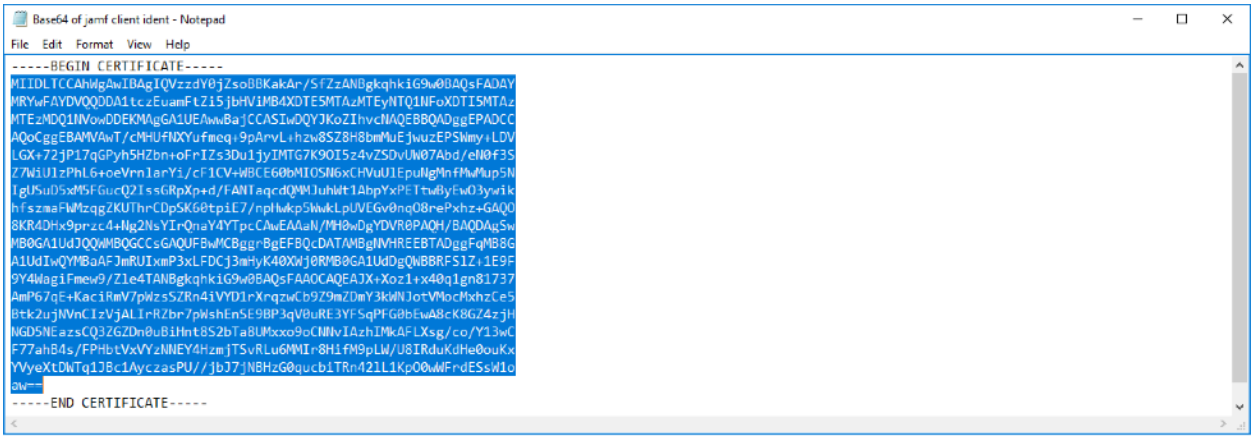



The wizard will step you through the export process. Do not export the private key. Select the Base-64 export format.




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

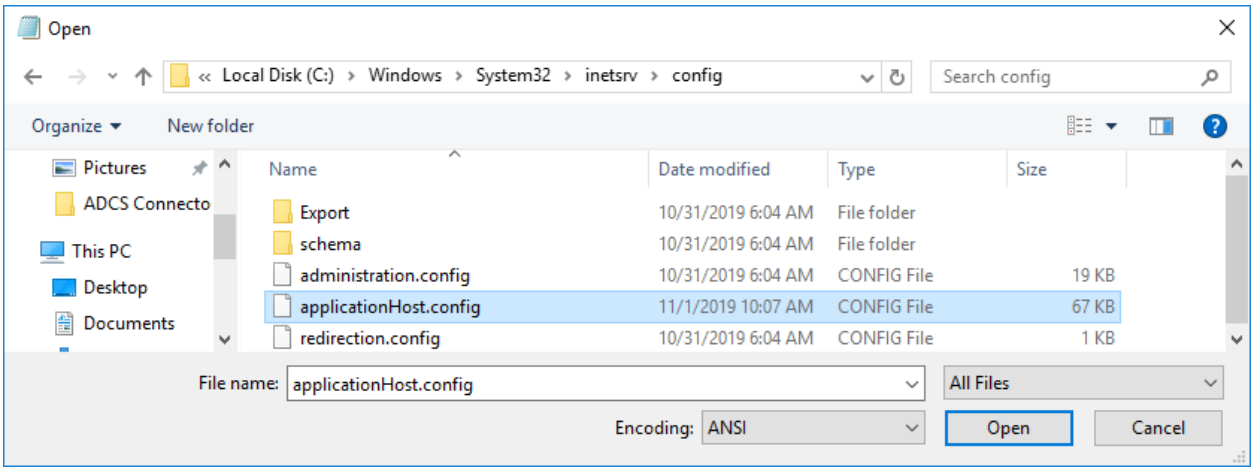


The instructions and screen shots for navigating to IIS's client certificate authentication configuration were demonstrated in "To review client certificate authentication settings..." in the "What you should see after running the ADCS Connector Installer" Appendix section. You can use the configuration 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.

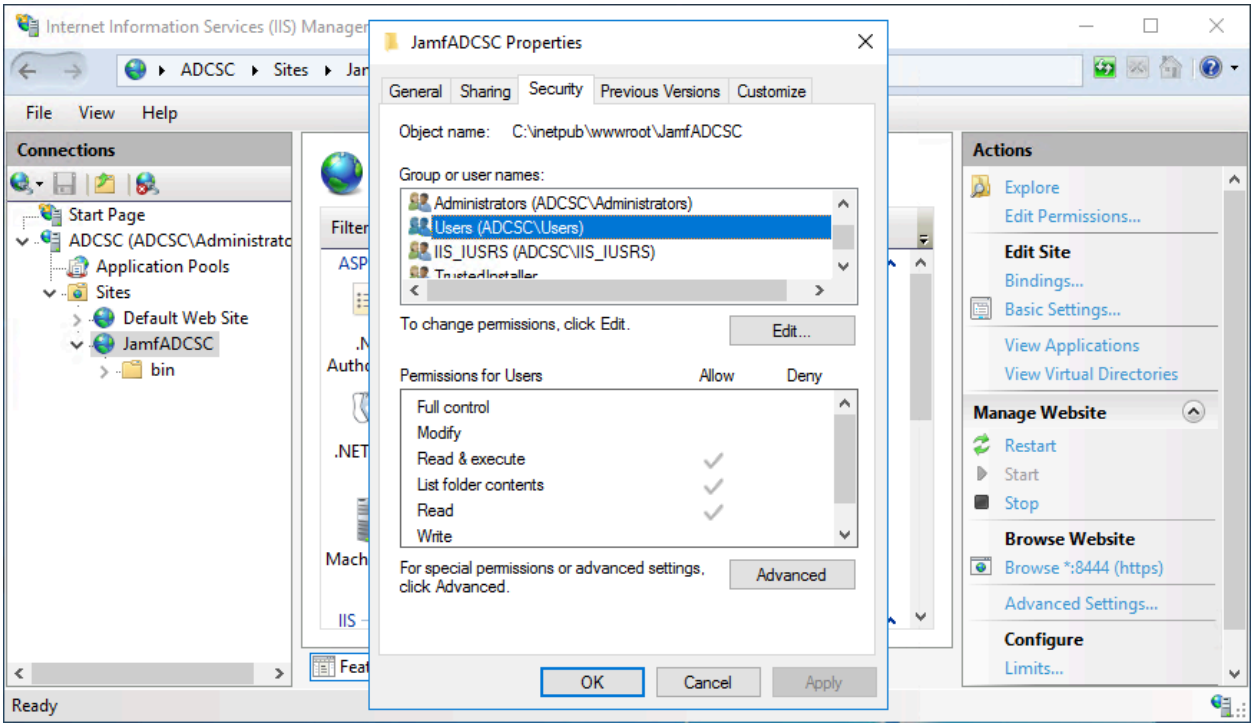
Run NotePad (or another text editor) as Administrator, change the file filter to "All Files" and navigating to the IIS applicationHost.config file.




Replace the existing base-64 key with the contents you coped from the .der you exported using the certificates snap-in and remove any carriage returns so the key is one contiguous string.


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

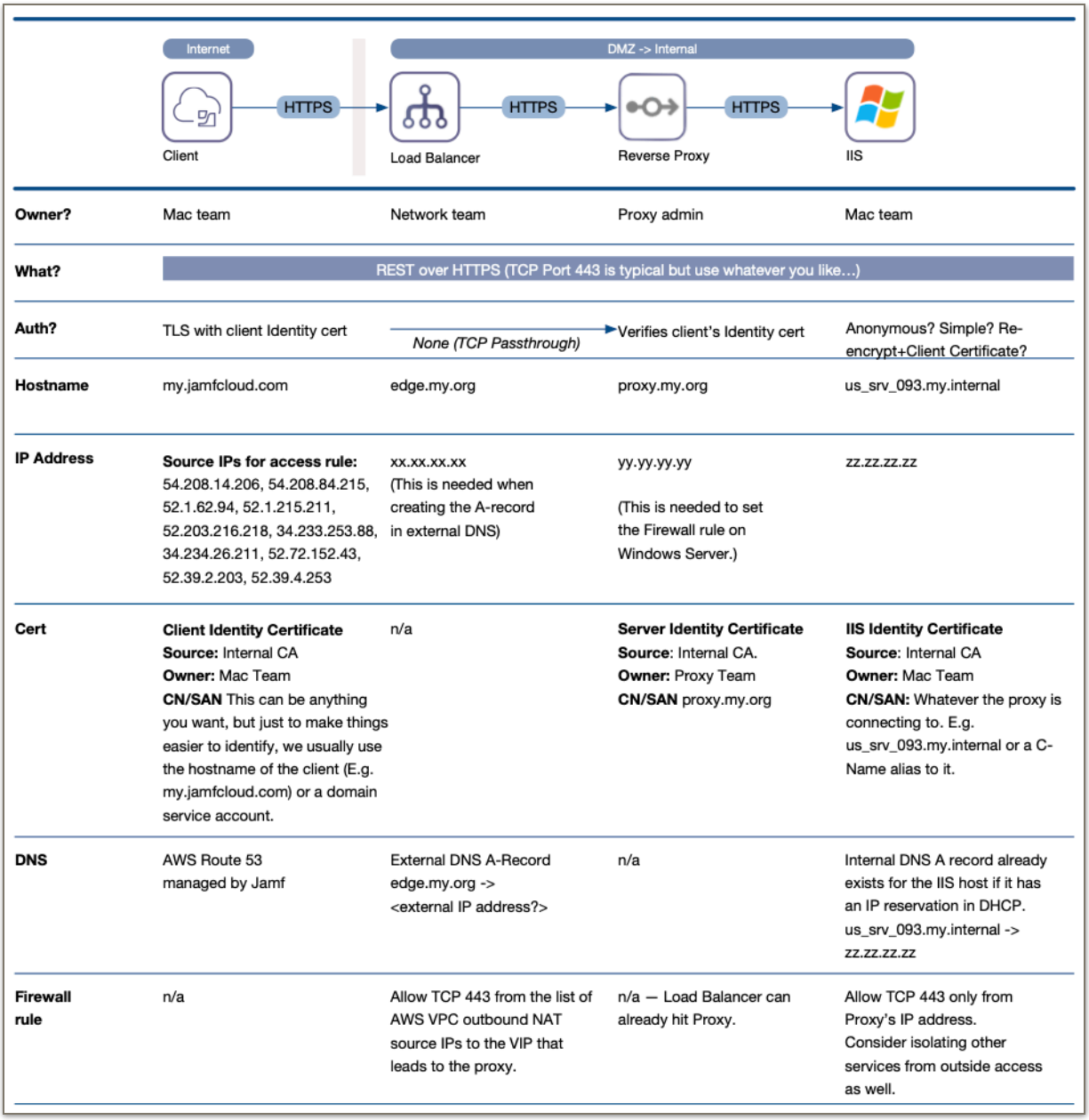
The Connector will be implemented in many different IT environments, each with different network layouts and practices for service deployment. Some of these will mandate high-availability, reverse proxy, or web application firewall configurations. The understanding needed to implement any of these is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

The implications are:

- If a proxy, load balancer, or web application firewall is used to terminate TLS, the server certificate must have a name or SAN that matches the host name 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 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:




| | | | | |
|---|---|---|---|---|
| **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 (This is needed when creating the A-record in external DNS) | yy.yy.yy.yy (This is needed to set the Firewall rule on Windows Server.) | 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 DCOM 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](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 (ADCS) 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

Once you've installed the ADCS Connector, you might create a Jamf Pro profile with a certificates payload and scope it to a single device. You could set it to be installed via Self Service so you can decide when it should install, noting the time so you can check logs if necessary. Most of the time, the certificate profile comes down without a hitch. But if there is a configuration error, the install will spin for several minutes and the profile will show as pending in the device's management tab in the Jamf Pro. Console.

To troubleshoot, we'll need to trace through the certificate deployment process to figure out where the problem lies.

1) A device falls into scope for installation of a certificate profile that uses the ADCS Connector
2) Jamf Pro detects the need for a new certificate. It generates a signing request and sends it to the Connector.
3) The Connector sends the request to Microsoft ADCS.
4) ADCS returns a request ID. The Connector returns that to Jamf Pro.
5) After a short pause, Jamf Pro asks the connector to retrieve the certificate for the current request ID.
6) The Connector retrieves the certificate from ADCS and returns it to Jamf Pro.
7) Jamf Pro adds the certificate payload to a profile and tells the device to check in to retrieve and install it.

### Troubleshooting Step 1: "A device falls into scope for installation of a certificate profile"

Normally profile installs have to wait for an Apple Push Notification cycle. We can take all that out of the equation by putting the profile in Self Service. If you see the profile in Self Service on the intended device, you've scoped it correctly.

### Troubleshooting Step 2: "Jamf Pro sends a signing request to the Connector"

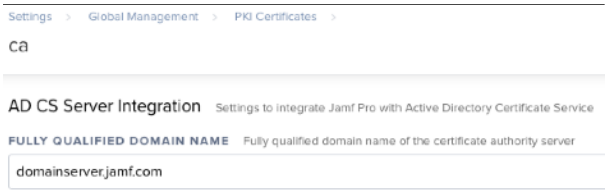
Most problems with this step have to do with a network problem, like not having a needed firewall rule or using the wrong SSL certificate. We can track that down in the logs. The Jamf Pro application logs will show if there was a connection or SSL error when it tried to open a connection to the Connector. The IIS logs on the connector will let us know if the connection ever arrived on the far end. Your firewall admin can check the firewall logs to see if the traffic arrived from Jamf Pro but got blocked.

See"*Troubleshooting Topic: Viewing the IIS Access Logs*" below if you're not familiar with the IIS logs.

The following are some messages you may see in the Jamf Pro logs and some possible causes.

### Viewing the Jamf Software Server Logs

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

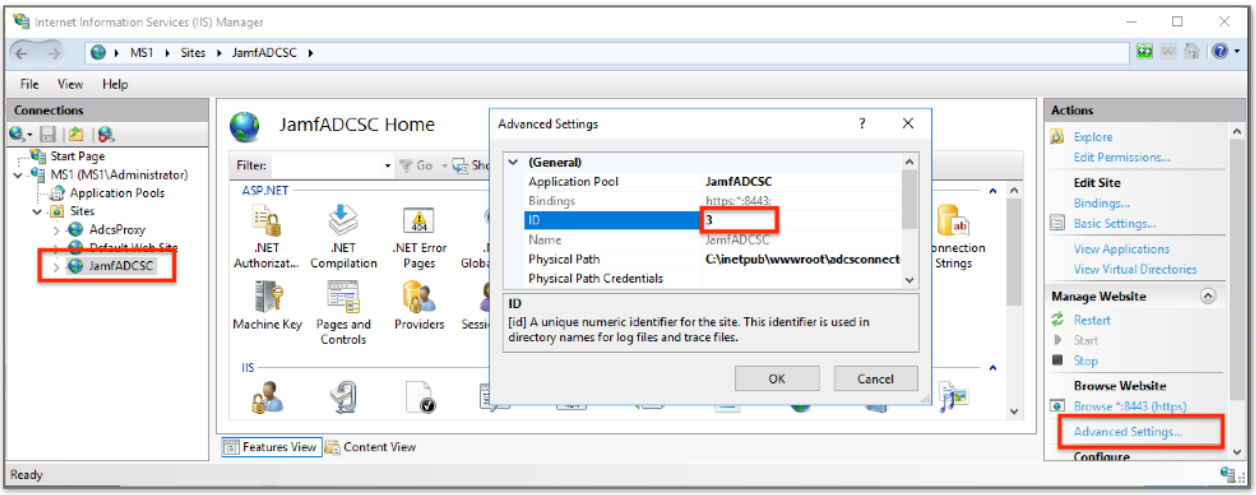
| Log Error | Solution |
|---|---|
| | You entered the wrong host name for the ADCS Connector. Remember that this is the external FQDN to which Jamf Pro will connect, not the actual host name of the computer running the connector software on your internal/DMZ network. Do not use an IP address. The hostname must have an entry in public DNS that leads to your Connector's external IP address. |
| | |
| `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)` | The wrong hostname has been entered for the AD Domain Controller or the server is unreachable from the Connector host:<br>  |
| `Caused by: com.jamfsoftware.pki.adcs.exception.AdcsConnectorCertificateNotIssuedException: CR_DISP_DENIED: Request denied` | There are three main suspects.<br>1. ADCS Connector server hasn't been given permissions on the template or CA.<br>2. Template key size is larger than 2048 bits<br>3. Your template's "Name" and "Display Name" are different and you entered the display name into Jamf Pro. Use the actual template name |

| Log Error | Solution |
| --- | --- |
| `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];` | In this case, the admin has entered an IP address for the ADCS Connector. Use of IP address subjects has been depreciated by standards bodies and is not supported by Jamf Pro.<br><br>You may also see this error even if you used a host name. In this case, 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. This would be the case if you are using a reverse proxy or external VIP DNS entry that is different than the hostname of the Jamf ADCS Connector server but you used the host name of the Connector Server when installing the software and that became the subject of the generated cert. |

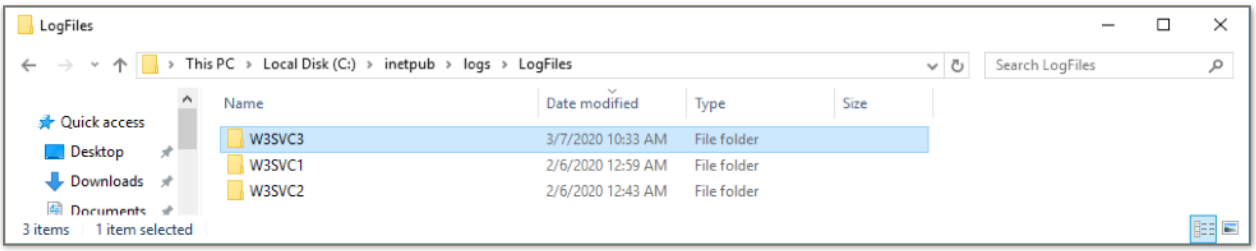
miscoNormally profile installs have to wait for an Apple Push Notification cycle. We can take all that out of the equation by putting the profile in Self Service. If you see the profile in Self Service on the intended device, you've scoped it correctly.

### Troubleshooting Topic: 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. If you see connections, you know your Jamf Pro traffic is getting to IIS. Otherwise, you have a networking/firewall issue.

### Example IIS Log Entries — The 403.16 Error

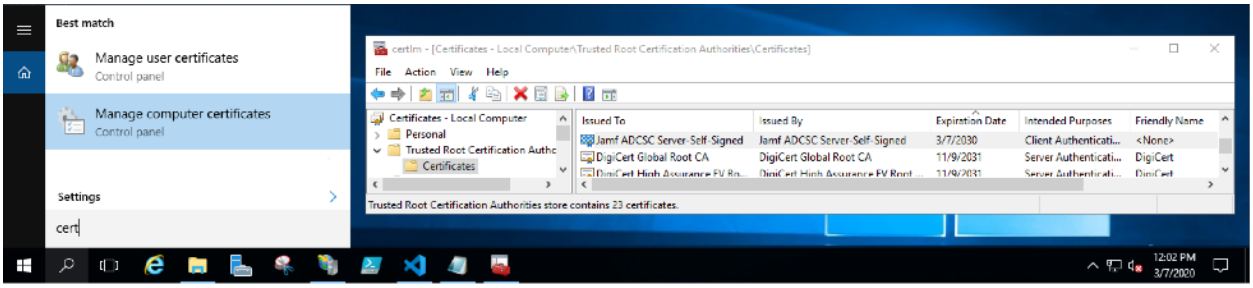
In the example below, we see that at 2020-03-07 15:33:11, a client with IP address 192.168.1.57 attempted to connect to the /api/v1/certificate/request endpoint on port 8443. An HTTP status of 403.16 was returned.

```
#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
```

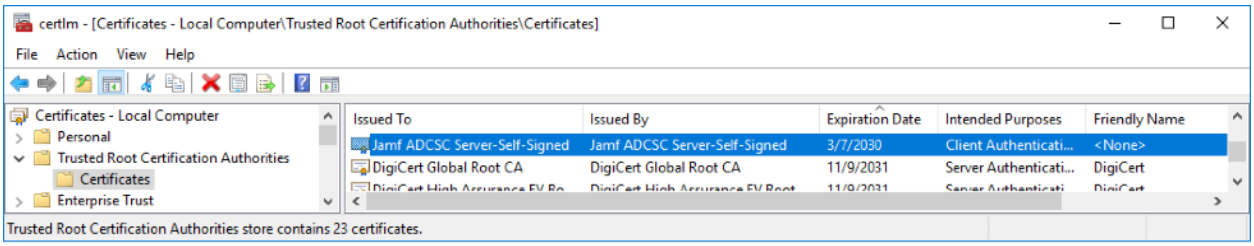
If we lookup that error, we'll find https://support.microsoft.com/en-us/help/942061/error-message-when-you-visit-a-web-site-that-is-hosted-on-iis-7-0-http.

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.



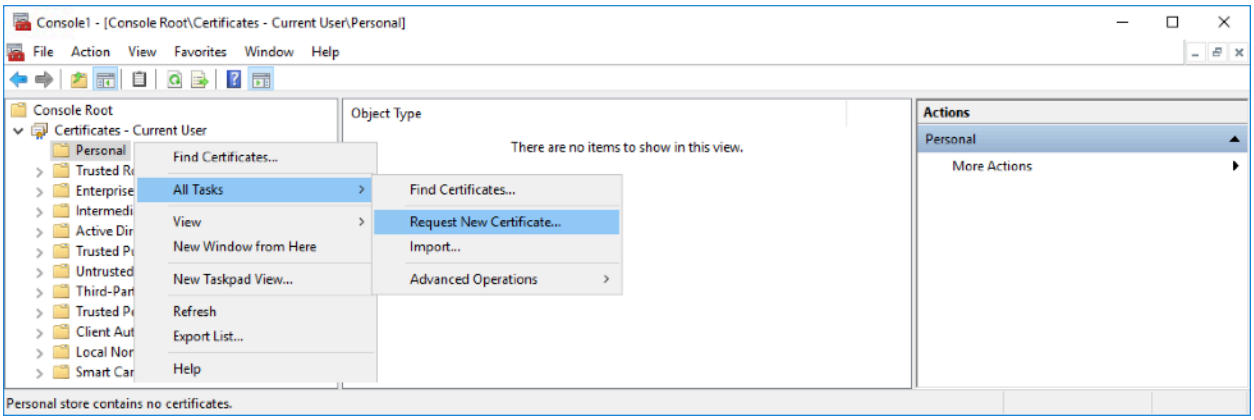

It's not uncommon for organizations to have a computer management restriction that blocks any new certs from being added to the trusted roots because if anything wrong gets in there you're a sitting duck for MITM attacks. We've seen several cases where the server where the Connector was being installed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




### 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

Testing ADCS Connector with Jamf Pro requires that put a device into scope for a certificate profile, wait for the interaction with Apple Push and the device, then wait for the communication with the Connector and from Connector to ADCS to complete. This is an important test but not very 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 Microsoft AD Certificate Services hostname (ms.jamf.com)...
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
Reply from 192.168.1.112: bytes=32 time<1ms TTL=128
Reply from 192.168.1.112: bytes=32 time<1ms TTL=128
Reply from 192.168.1.112: bytes=32 time<1ms TTL=128
Ping statistics for 192.168.1.112:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),


Running certutil will let you know if the needed network ports are open and if your
current username has permissions on the CA. (In this example, ms.jamf.com is the name
of the MS ADCS server, and "CA" is the name of the 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
```

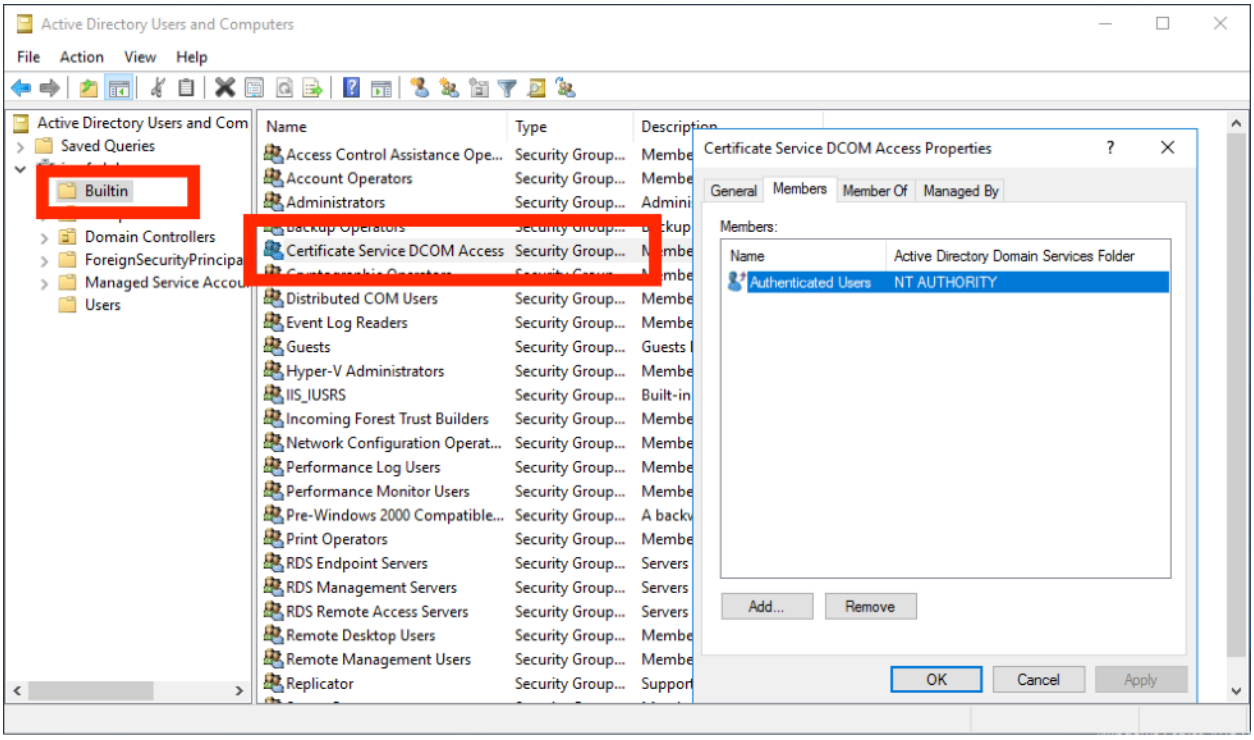
```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

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 temporarily 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. The subject of the client certificate used by Jamf Pro when authenticating to IIS does not have to match the host name of your Jamf Pro instances so there's no issue there.

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

Could you have 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 myclientcert.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 or Windows Server HTTPS App Proxy?

That would be a great approach since it reverses the traffic flow and we like to avoid inbound firewall rules. However The Jamf ADCS Connector uses client certificate-based authentication, which as of this writing is not supported by Azure App proxy as stated here:

> ***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*

So if you used Azure AD App Proxy, nothing is checking the authenticity of the authentication certificate. TLS is being terminated in the cloud and the client certificate will not pass through to IIS.

Organizations that prefer to use Azure App Proxy should consider using the SCEP Proxy method for their certificate deployment. The following Microsoft references explain the process.

*Ref: "Integrate with Azure Active Directory Application Proxy on a Network Device Enrollment Service (NDES) server" https://docs.microsoft.com/en-us/azure/active-directory/app-proxy/active-directory-app-proxy-protect-ndes*

*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*

In addition to the Azure approach, Microsoft Windows Server also offers an App Proxy feature. But the same issue arises -- here too, the App Proxy terminates TLS, breaking the flow of the client authentication certificate. You can, however, put a reverse proxy in front of the Web App Proxy feature built into MS Windows Server and configure the reverse proxy to verify Jamf Pro's client authentication certificate and then pass the traffic on to the MS App Proxy layer. But since TLS have been terminated at the MS App Proxy layer and does not pass-through to IIS, you would need to turn off authentication on IIS and rely on strict firewall rules to make sure nothing can touch the web server unless it flowed through the reverse proxy and theclient authentication certificate has been verified. The security of this configuration might be acceptable to some organizations but will not be sufficient for others.

Some reverse proxies are capable of terminating TLS with a client certificate authentication and then introduce another authentication method as they emit the traffic that MS Windows App proxy will pass-through to IIS, like Simple auth, for example. Beware! This is a complex setup and likely to lead to significant frustration unless your network administrator is totally familiar with all the components involved.