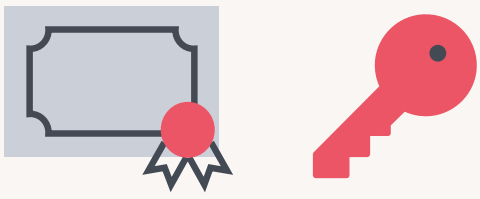
# Managing Certificates with Jamf

Certificates play a vital role in securing, authenticating and maintaining the stability of your Apple fleet. When used correctly, they will increase visibility while cutting down security risks.

## The basics of certificate-based communications

Certificates can appear confusing or overwhelming. This is often because they have been misused or misunderstood, and rolling out a successful cert-based project can be confusing and overwhelming without help.

Also, certificates are often misused in a customer environment. For instance, security tells you that there's a dozen or so certs that need to be installed on your Mac fleet. So you use Composer, back them up, and put them into your provisioning workflow.



**But what are those certificates for?**

We don't know. But they're installed in the keychain now. Are they trusted? Is the chain complete? Maybe. Maybe not.

Let's demystify certificate creation and deployment!

## What are Certificates?

A certificate is just a text file. That's it. There's a ton more behind the scenes as far as the signing, cryptography, and private key infrastructure (PKI) that you can delve into, but that's the basic file.

### How do you make a certificate?

To make a cert, we need a certificate request, or CSR. Then, we reach out to a certificate server that talks to the Certificate Authority (CA) and it will add its part to the conversation. Sometimes we add the root cert as well. Then, we have a digitally signed certificate.

If you open a certificate in a text editor, you will see only unreadable hex code, because the certificate is encrypted. On a Mac, you can still inspect the contents of a certificate using Spotlight by hitting the space bar, or by opening it in keychain access to see what's inside.

### What data does a certificate contain?

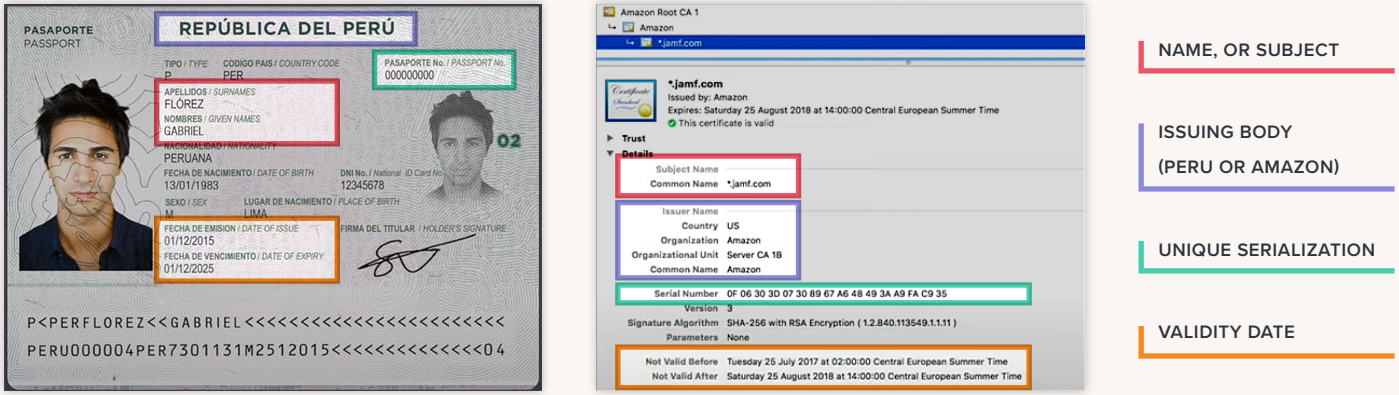
Identification data. A cert is essentially an id card. A certificate is a signed and trusted source of identification, sort of like a driver's license or passport. Like a passport, the validity of trust for the document is based on the issuing authority.

If I had a forklift license issued in Canada, that doesn't mean I'm authorized to operate a forklift in Germany. Similarly, a Sam's Club card doesn't allow me to shop at Costco.

Something like a passport only works as a valid ID everywhere because the issuing bodies have all agreed to trust each other. If Peru says you're a citizen, Canada accepts that because it is a trusted document.

If, for instance, someone produced a galactic passport and it claimed this person had the rank of Jedi Master, no one would believe the contents of that document because it didn't come from a trusted source.

Let's take a look at a certificate next to a passport and see exactly how much they have in common.




Certificates aren't totally incomprehensible!

## Why use certificates in computing?

For security's sake. The use of trusted certificates allows for encrypted communication, which can prevent information from being intercepted while in transit.

When you visit a website using https and you get a green checkmark or lock icon in the address bar, you're communicating with that server using a certificate. The shared connection with that site is secure.

### Certificates as Credentials

Certs can be used as an alternative to user credentials. When a client presents a certificate, the server inspects the cert and decides if it's going to trust that client based on the contents and the certificate authority that made that cert.

**Where can we use this form of secure identification?**

With 802.1x WiFi, typically referred to as cert-based WiFi. This is much better than a traditional WPA password, because the certs used cannot be shared.

Typically your network authenticator will be using Extensible Authentication Protocol (EAP) or Radius (another authentication protocol), depending on the encryption choices.

Jamf has the advantage of validating the client that is connecting to the network. The client device can present user information inside the certificate that proves exactly who is using a given device on your network.

This is very useful for your friends in InfoSec.


**Using certificates to connect to VPNs**

Another common use for certificates is connecting to a Virtual Private Network (VPN). Similar to WiFi, the username and password might not be enough to establish trust. We want to be able to validate the device is also trusted. Access is denied if the user credentials are disabled, or if the certificate has been revoked.

**Authenticating wired networks with certificates**

You can ensure your company data is secure with certificates.

802.1x authentication is not limited to Wi-Fi. Though less common, it can also be used on a wired network. It's another way of preventing just anyone from gaining access to a network connection and gaining access to private data.

**Using certificates with encrypted email**

When certificates are signed and trusted, an email message cannot be read without the correct certificates installed. This technology also ensures that the message hasn't been modified in transit after it was signed and sent.

# Managing certificates with Jamf Pro

## Deploying Certificates

Deploying certificates with an Apple MDM is straightforward: the MDM vendor, such as Jamf, maintains the certificate life cycle. For security, they have various opportunities to revoke the users' access if they:

- Fall out of compliance
- Leave the company

## How certificates play a role with Apple and MDM

- **Push notifications (APNS)**
  Apple's entire push notification system relies on a chain of trusted certificates for communication. None of it would work without certs.
- **Supervision identities**
  In the case of Apple Configurator, Jamf creates a certificate-based supervision identity that can be shared between multiple provisioning Macs used to supervise and enroll iOS devices. This same supervision identity can be added to Jamf Pro for the devices that it supervises.
- **Developer signing**
  As a developer, you get involved in dozens of types of certificates from app signing development distribution and even Apple Pay certs. You're probably using some of these certificates already in Jamf.

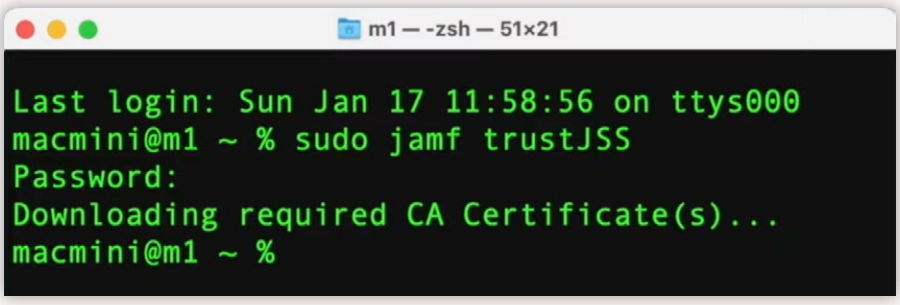
**Built-in certificate authority**

Jamf Pro has its own Certificate Authority (CA) built in. This self-signed CA produces a root cert that must be trusted on the device before it can trust the MDM profile.

## How and where does Jamf Pro use certificates?

The short answer: almost everywhere. Certificates play a role in:

- Enrollment profiles
- Device management with the Jamf binary




The trustJSS Command

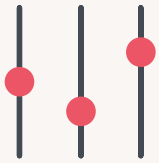
- Apache Tomcat SSL: For an externally facing server you'll need to install a publicly-trusted certificate.
- The built-in certificate authority is where you would configure Jamf to talk to an external CA server. SKAT proxy and ADCS Connector Settings are also configured here.
- Health Care Listener: Uses certificates to secure communication inside the hospital network.
- Single Sign-On
- LDAP-S over SSL
- The enrollment process
- Signing QuickAdd package: requires an app distribution certificate from Apple. This same cert is used by Composer to sign your other packages.
- Configuration Profiles: Config profiles created in Jamf Pro are signed automatically. This keeps them secure when deployed. If you download a config directly from the console, it's already signed. That's why you can't view the raw xml data with a text editor. If you need to edit a config profile created with Jamf, you'll need to unsign it first.
- App Provisioning Profile: A provisioning profile is a different sort of profile that also uses a cert. When working with custom iOS apps, your developer might need you to deploy the app with a provisioning profile. It's less common today, but Jamf does support it.
- Developer Certificate: you would get that cert at developer.apple.com, among other certs you might need. The more common method these days is to let Xcode create and embed the distribution certificates and provisioning profiles for you automatically. Once that's done, you'll have yourself an in-house app: a custom iOS app that can be deployed using Jamf to register test devices. If you need to deploy your custom app to hundreds of iOS devices or more, that will require an Enterprise developer signing certificate.
- Apple deployment portals: strictly speaking, these next few are actually tokens, a private key. Device enrollment and volume purchasing both get their certs from Apple using the token provided. (The more modern place to find that information is in Apple School Manager or Apple Business Manager.)
- GSX (Global Service Exchange)
- Cloud Distribution Point (JCDS)
- Jamf Push Proxy: if you send notifications to your devices through self-service, you'll need a push proxy certificate. They are automatically generated, so this one's easy to get set up.
- Patch Management and Customer Experience Metrics: while invisible to the Jamf admin, they do communicate using certificates and are sent securely to our servers.

So, yes. Jamf uses certificates nearly everywhere in our portfolio. But none of them are the certs that are being generated to install onto your devices.

**PRO TIP!**

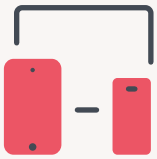
Jamf Cloud takes care of all of this web app work automatically. You need not ever worry about your TomCat settings again.

## Creating Certificates with Jamf



**Delivered via config profile**

To create certificates with Jamf, you do that in payload config profile. Jamf can also combine certs and Wi-Fi payloads for simplicity.



**Works with Mac or iOS**

This method for creating certs works for Mac OS, iOS and even tvOS. You can also include multiple certificates in a single payload if needed.

**It's all about the context**

When talking about user certificates and device certificates, it's important to know the context we're using. You'll hear talk of user certs and device certs or machine certs. For the most part, a user cert contains the user information in the subject, and a machine cert contains information in the subject about the device specifically.

What's important to know is if you choose to deploy the cert at the device level, you're installing that certificate into the system keychain and it's available to all current and new users of that device.

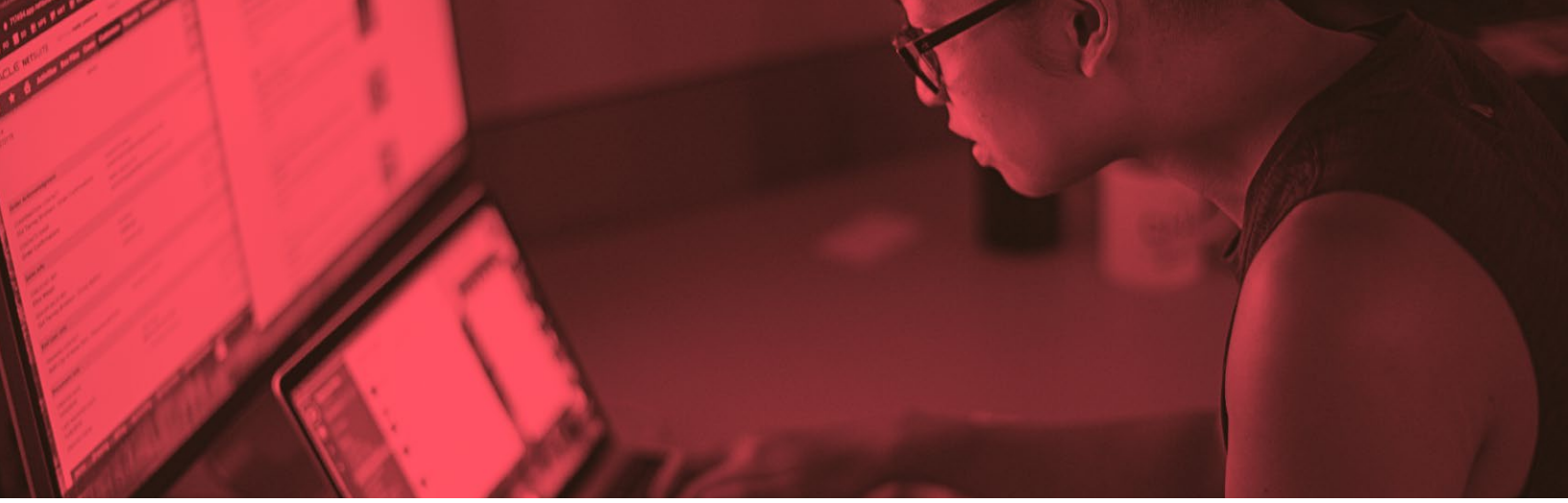
If you deploy it into the user level, it's installed directly in that user's keychain and won't be available to any other user on the system.

## Jamf Simple Certificate Enrollment Protocol Proxy (SCEP) Proxy and ADCS Connector

If you're going to be deploying VPN certs or rolling out 802.1x Wi-Fi, you'll need to use one of these to do it. They are related but different product offerings. They both exist as alternatives to binding your Mac to Active Directory to obtain the certificate, the benefit being that both solutions can work to produce certificates for Mac, iOS and tvOS devices.

**ADCSC or SCEP Proxy? Which do you choose?**

The correct choice between the two depends on so many details there is no single right answer. Don't think of it as SCEP vs. ADCS. You may even use a combination of both. We at Jamf are eager and willing to have those conversations with you to decide the best path of success for your cert-based projects. Please reach out.

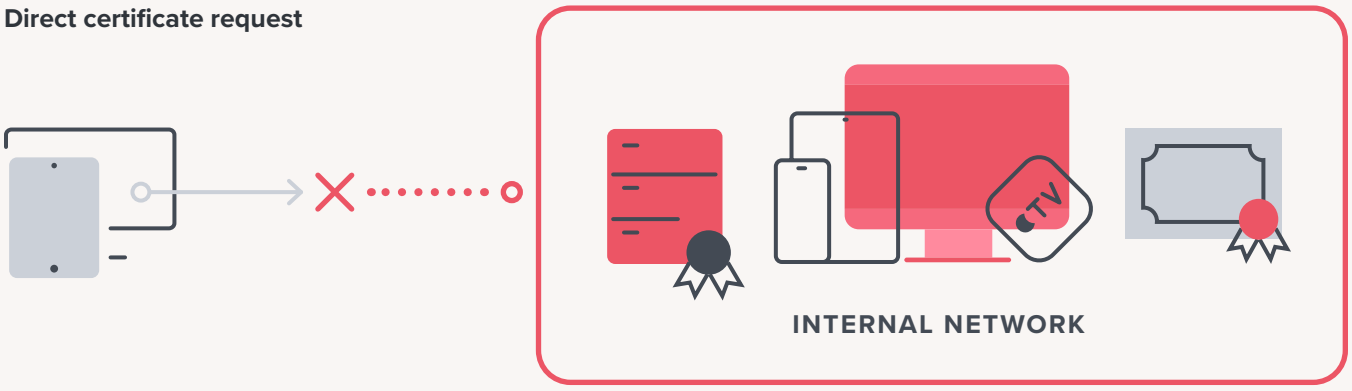


## Deploying Certificates

There are a few ways to deploy certificates:

- Manually, by going to a web portal and entering the info. This is the more cumbersome way.
- Through third-party applications like Nomad or Jamf Connect. These tools can take already-provided information from the user and then make the certificate request on their behalf. Some challenges: this still requires the user to enter some data, and the requirement that the request must be made from within the network or via VPN can cause problems. Additionally, these apps are only available on MacOS.
- Direct certificate request: Jamf Pro can create the request and the device can communicate directly with the server. This means it can all be automated. Communication is between the device and certificate server, which means the device must be on same network as certificate server.

**Direct certificate request**




If you have devices outside the network and they try to contact the certificate server for that certificate, it's very unlikely that the security team will allow this to go through.

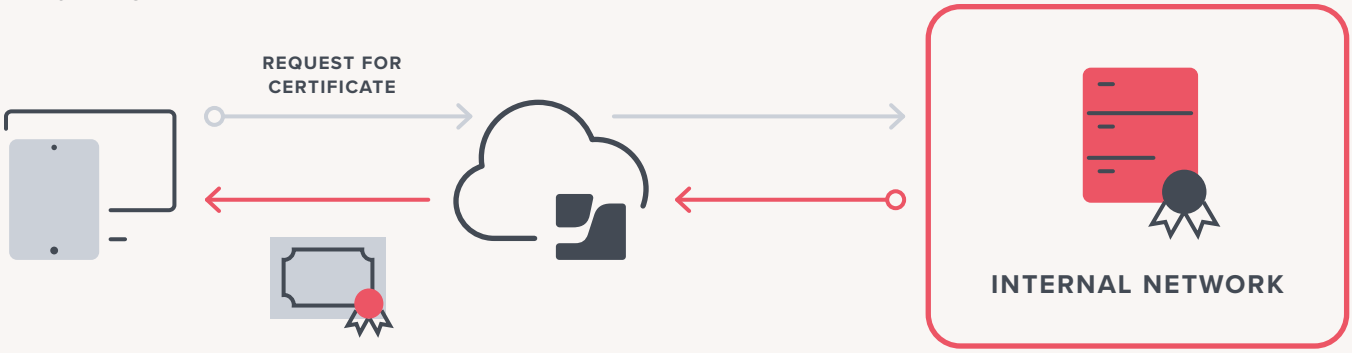
## Jamf Certificate Proxy

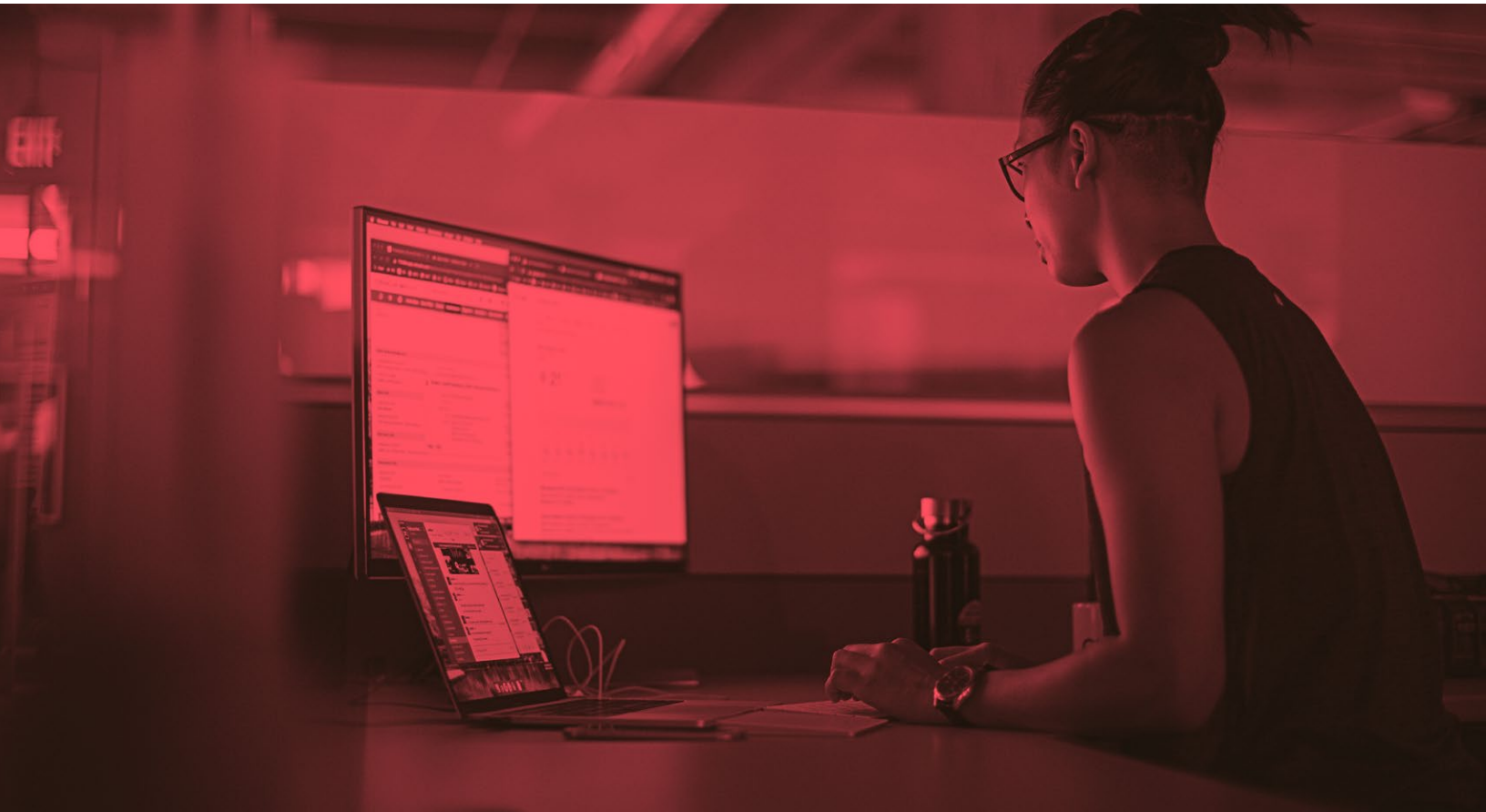
This is where Jamf Pro can help without compromising security, and is the most common method for requesting certs.

In this method, Jamf Pro acts as a proxy between the device and the certificate server using SCEP or ACS Connector. This provides the benefits of the previous methods with an added benefit that the communication flow changes. The device only needs to be able to contact the Jamf Pro server. That means the device can be on any network and still get the required certs.

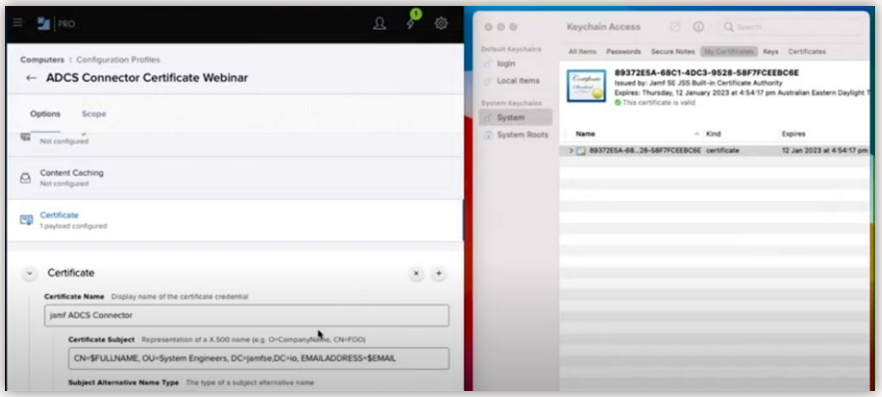
Here's what the communication is like:

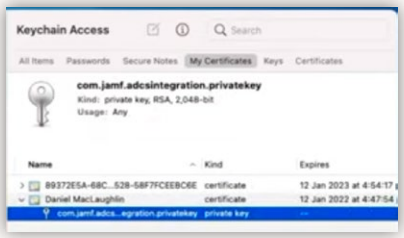
**Proxy using Jamf Pro**



And here's how it looks to the Apple admin using ADCS connector (which is similar to the look using SKEP), which has already been configured. This particular device has already been enrolled. And the process is happening on a guest network with no connection to the server with no Active Directory binding:




On the right we can see the keychain access application that is already showing one certificate: the device enrollment certificate.

You can see the contents of the request: full name and email address for the content variables. The admin will then go to the scope tab, add the device, and save.

From this point Jamf Pro will now be communicating with the Jamf server. It will request the certificate from the certificate server, it will deliver the certificate, and Jamf will deploy it to the device.

The certificate can now be used as an authentication method for Wi-Fi or VPN.

To make it a *trusted* certificate, you'll need to deploy the root certificate of the server to provide a trust chain.


## So you have the certificate. How do you use it?

An end user can manually select the certificate and use it as an authentication to the corporate wireless network.

When the user attempts to connect to an 802.1x enterprise network, they change the mode to EAP/TLS and then select a certificate from the keychain.

Similarly, with VPN the end-user will see similar options, providing that kind of VPN support certificates as a method of authentication. You would need to work within the respective teams within your organization for this.

This can all be automated by including the network and VPN settings payload within the configuration profile that contains the certificate payload.

## How to start the process of certificate creation in your organization

You will need:

- A supported version of the OS on the device
- A supported CA such as Microsoft certificate authority, digicert, Entrust Certificate Solutions or Venafi
- Support from other teams in your organization
  - Networking
  - Security
  - Certificate team

Reach out to a Jamf representative to learn how to help you succeed in deploying certificates within your organization. We can communicate with other teams in your organization to provide context in what is required from each department in order to have a successful deployment.

Have questions? Please reach out to us at info@jamf.com and we'd be more than happy to schedule some time with you to sit down and talk it through.

## Resources:

**From Jamf:**

JNUC Cert talk: jamf.it/jnuc-cert

Jamf as SCEP Proxy: jamf.it/scep

Jamf ADCS Connector: jamf.it/adcsc

Cert Deployment 101 webinar: jamf.it/cert-101

**From Apple:**

MDM Cert Guide: jamf.it/mdm-cert

Requirements for trusted certs: jamf.it/apple-cert-trust

Limits on trusted certs: jamf.it/apple-cert-limits