Managing Certificates with Jamf

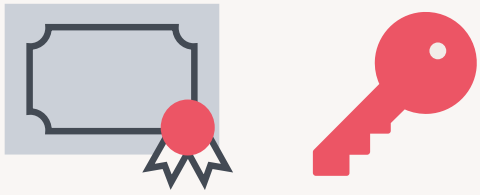
Certificates play a vital role in authenticating, securing and maintaining the stability of your Apple fleet. When used correctly, certificates can confirm user identity while minimizing security risks.

The struggle with certs

Certificates, or certs, can be overwhelming and appear confusing at first glance. This is often because they are misunderstood and misused, so rolling out a successful cert-based project can be a struggle without help from other teams.

An example of incorrect certificate usage is installing the certs into the keychain via packages and scripts. Also, sometimes deployed certs are never even used or are missing the complete chain of trust and don't actually increase security.

Certificates should be used to validate identity, secure the device, and simplify the user experience.



What is the right way to deploy certificates?

When using MDM, configuration profiles are the modern and secure method for deploying and revoking certificates. Additionally, Jamf can manage the certificate renewal and re-distribution to the client devices before it expires.

Let's demystify certificate creation and deployment!

What is a certificate?

A certificate is a secure text file. That's it. Of course, there is a lot of technology behind the scenes regarding the cryptography, signing, and the private key infrastructure (PKI) used in generating a certificate. But, simply put, certificates are text files. The text contained inside the certificate is what is important as well as what authority issued the file.

How do you make a certificate?

To make a cert, we first need a certificate signing request or CSR. It contains information that the Certificate Authority (CA) will use to create your certificate. It also contains the public key that will be included in your certificate and is signed with the corresponding private key. Sometimes we add the root cert to complete the trust chain. Then, we have a digitally-signed certificate.

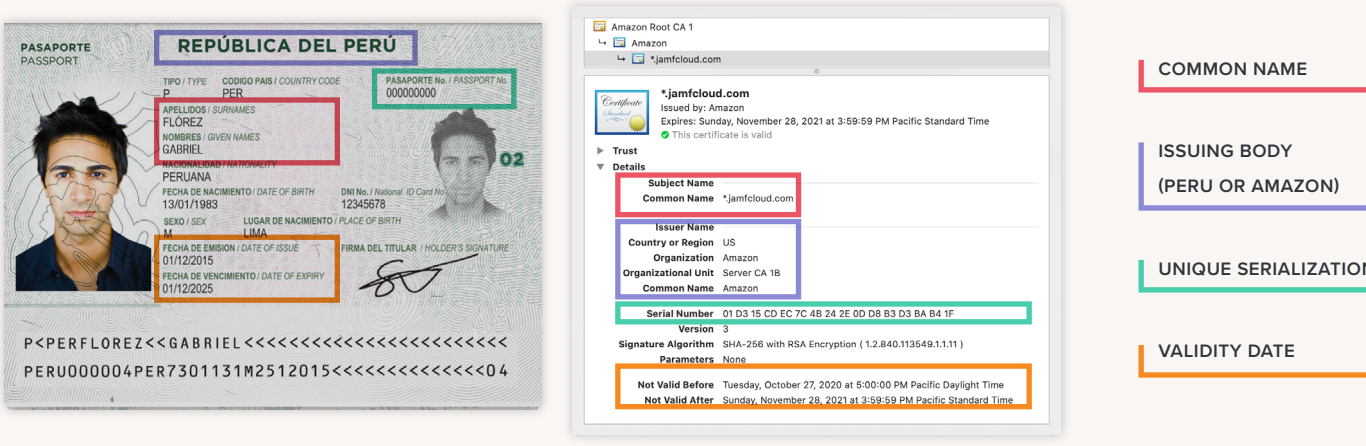
If you open a signed certificate in a text editor, you will see only unreadable hex code, because the certificate is encrypted. On a Mac, you can still inspect the contents of a certificate using Spotlight (by hitting the space bar), or by opening it in Keychain Access to read the details inside.

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, similar to a driver's license or passport. Like a passport, the validity of trust for the document is based on the issuing authority.

If you had a forklift license issued in Canada, that doesn't mean you are authorized to operate a forklift in Germany. 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 it because the passport is a trusted document.

The important data is the issued and expiration date, subject name (SN), subject alternative name (SAN), serial number, and issuer name. The chain of trust is crucial when validating certs. The most important considerations are which CA generated the cert, whether the CA is publicly trusted, and whether the intermediate certs referenced are present in the trust chain.



Compare a passport next to a certificate to see exactly how much they have in common!

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 whilst in transit. When visiting a website using HTTPS your browser will get a green checkmark or lock icon in the address bar, which means you are communicating with that server using a certificate. This is an indication that the shared connection with that site is secure.

Certificates as credentials

Certs can be used as an alternative to a username and password. When a client presents a certificate, the server inspects the contents to confirm specific criteria. If it passes, then the client gains access to the service or resources. We can disable the user's access by removing the cert when they fall out of compliance or leave your organization.

Where can we use this form of secure identification?

Certificate based Wi-Fi authentication

With 802.1x Wi-Fi the client presents a cert instead of a Wi-Fi password. This cert can present machine data as well as user identity information. This is much better than a traditional WPA password because the certificates are unique to each device and cannot be shared.

Jamf has the advantage of combining Wi-Fi authentication, certificates, and SCEP proxy instructions into a single profile that allows a user to join to the network automatically. The end-user does not need to be prompted for anything.

Using certificates to connect to VPNs

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

Authenticating wired networks with certificates

You can ensure your company network and private data are 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 is another way of preventing just any random device from plugging into a network connection and gaining access to private data.

Using certificates with encrypted email

When emails are signed, the message contents cannot be read without the correct certificates installed on the recipient device. This technology also ensures that the message hasn't been modified in transit after it was signed and sent.

How and where does Jamf Pro use certificates?

The short answer: nearly everywhere.

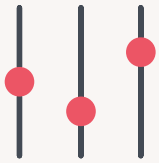
- Jamf Pro has a built-in certificate authority to author certs used in building the trust between the managed devices and the server.
- The enrollment profile can be signed to eliminate the CA "trust" step.
- Apache Tomcat SSL: For an externally facing 'on premises' Jamf Pro Server, you'll need to install your own publicly trusted certificate.
- Healthcare Listener: We use certificates to secure communication between Jamf Pro and the HCL.
- Single Sign-On: Jamf Pro can generate the certificate used to sign the messages sent to the Identity Provider.
- Directory Lookups: Jamf Pro can use SSL to securely communicate to Active Directory, LDAP or NDS.
- Package Signing: Requires an app distribution certificate from the Apple Developer portal. Jamf Pro will use this to sign the Quick Add package. This cert can also be used by Composer to sign your other software 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 Pro, you'll need to un-sign it first.
- App Provisioning Profile: This is a different type of profile that also uses a cert. Some iOS app developers might need you to deploy their app with a provisioning profile. It's less common today, but Jamf Pro does support it.
- Developer Certificate: The more common method these days is to let Xcode create and embed the distribution certificates and provisioning profiles automatically. Once that's done, you'll have yourself an in-house iOS app that can be deployed using Jamf Pro to registered 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 are actually tokens (like a private key). Device enrollment and volume purchasing both get their certs from Apple using the token provided. Previously called DEP and VPP, the current place to find the token is in Apple School Manager or Apple Business Manager.
- Global Service Exchange: We support Apple's option to query GSX for warranty information using a certificate.
- Cloud Distribution Point (JCDS): Client downloads of packages is done securely over HTTPS. Jamf Pro also supports certificate-based communication to other CDNs such as Akami, AWS S3 Buckets and Rackspace.
- Jamf Push Proxy: If you plan on sending notifications to your devices through self-service, you'll need a push proxy certificate. The Push Proxy cert is automatically generated by Jamf, 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 so information is sent securely to our servers.

"These aren't the certificates you are looking for..." While it's helpful to see some of the many places Jamf Pro uses certificates; none of the above certs are the ones being created and installed onto your devices!

PRO TIP!

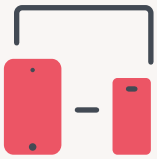
Jamf Cloud is already publicly trusted and eliminates the need to edit the Tomcat settings or configure a publicly trusted certificate.

Deploying certificates with Jamf



Delivered via config profile

To create and manage certificates with Jamf, just add them to a configuration profile. Jamf can also combine certs and Wi-Fi payloads for simplicity.



Supports all device types

These methods of creating certs work for Mac OS, iOS and even tvOS. You can also include multiple certificates in a single payload if needed.

Jamf Simple Certificate Enrollment Protocol Proxy (SCEP Proxy) and Active Directory Certificate Server Connector (AD CS Connector)

With users able to work from anywhere in the world, devices likely won't be able to reach your certificate server directly. Jamf can have a conversation with the cert server on behalf of the client to facilitate certificate creation. SCEP Proxy and AD CS Connector are similar and related, but separate, product offerings. They both exist as alternatives to binding your Mac to Active Directory to obtain the certificate. Another benefit is that both solutions can produce certificates for Mac, iOS and tvOS devices.

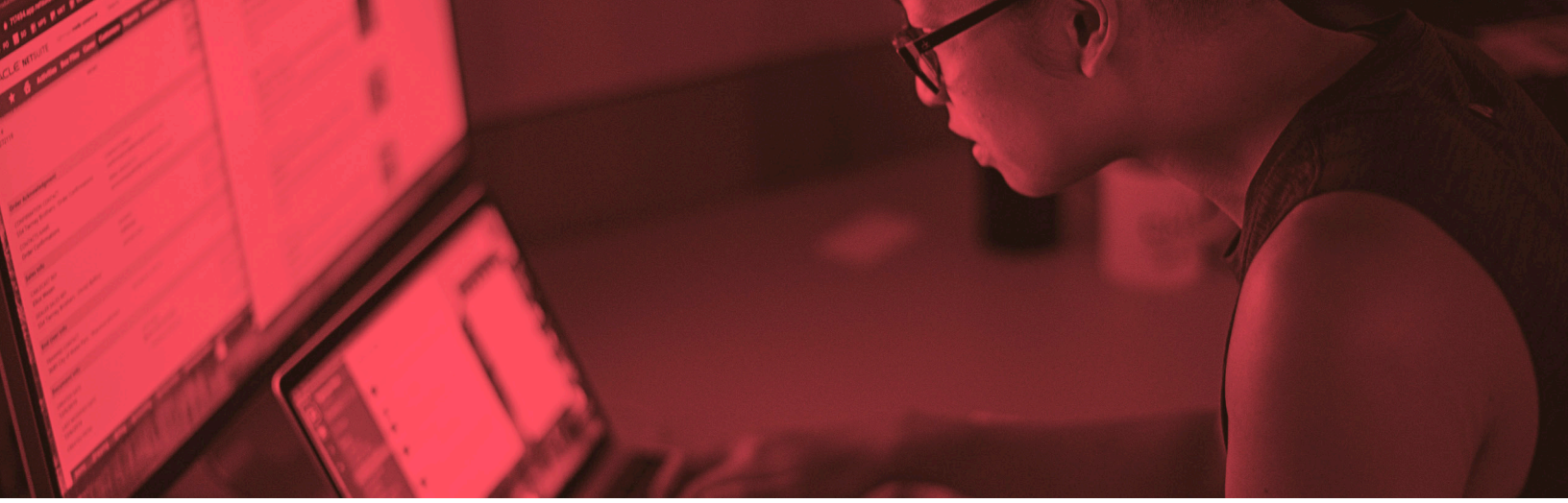
It's all about the context

When working with 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's information in the subject, and a machine cert contains information in the subject specifically about the device.

With Jamf Pro, it is important to understand that when you are deploying any type of certificate at the device level, you're installing that cert into the System Keychain and it's available to all current and new users of that device. If you deploy to the user level, the certificate is installed into that specific user's keychain and won't be available to any other user on the device.

AD CS Connector 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. AD CS. Your environment may require that you use both. We at Jamf are eager and willing to have this conversation with you to decide the best path of success for your cert-based projects. Please reach out.

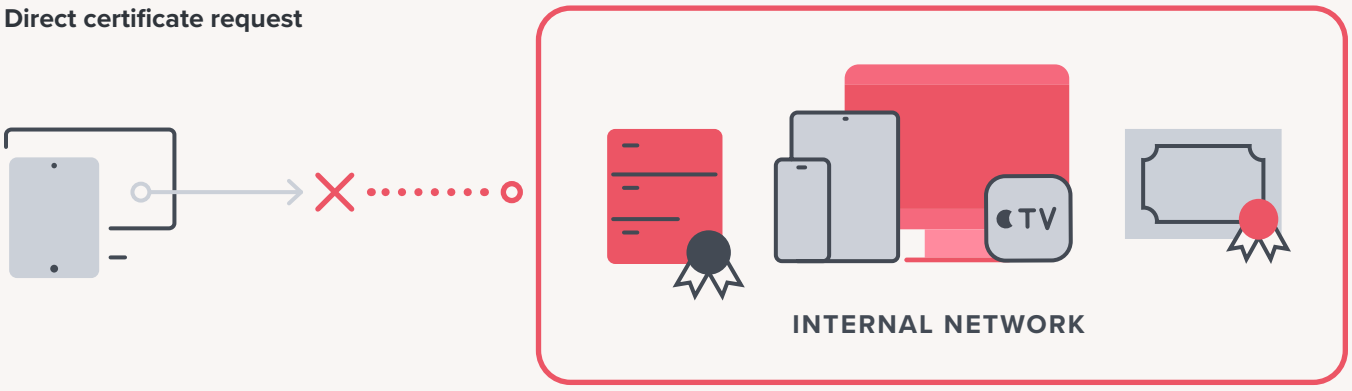


Creating Certificates

There are a few ways to create certificates:

- **Manually, by going to a web portal and entering the info.** This is the more cumbersome way that is rarely used outside of testing your environment.
- **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 on the domain network or via VPN can cause problems. Additionally, these apps are only available on macOS.
- **Direct certificate request.** Jamf Pro can create the request in a configuration profile payload but have the device reach out directly to the cert server. Communication is between the device and certificate server, which means the device must be on the same network as the certificate server or have the SCEP server exposed externally.

Direct certificate request



If your devices are outside the trusted internal network, none of the three above methods are likely to generate a certificate. Network Security usually won't allow it.

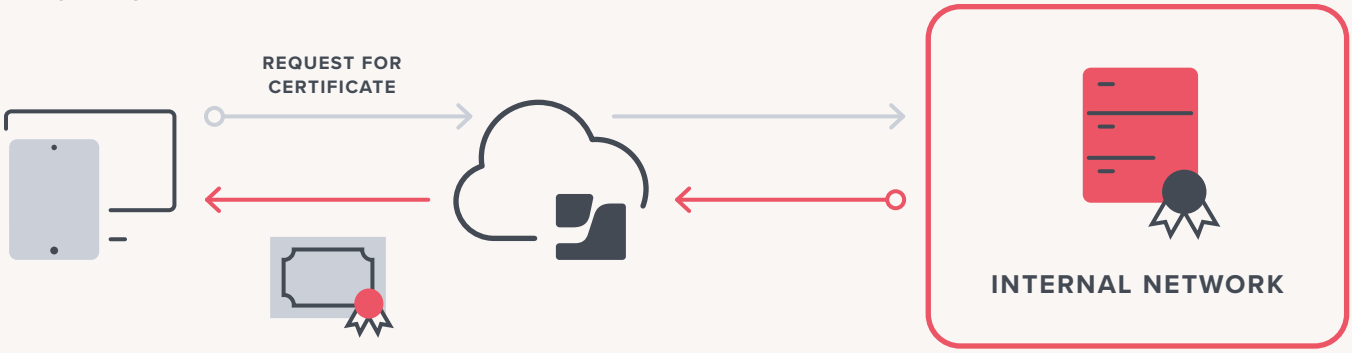
Jamf as a certificate proxy

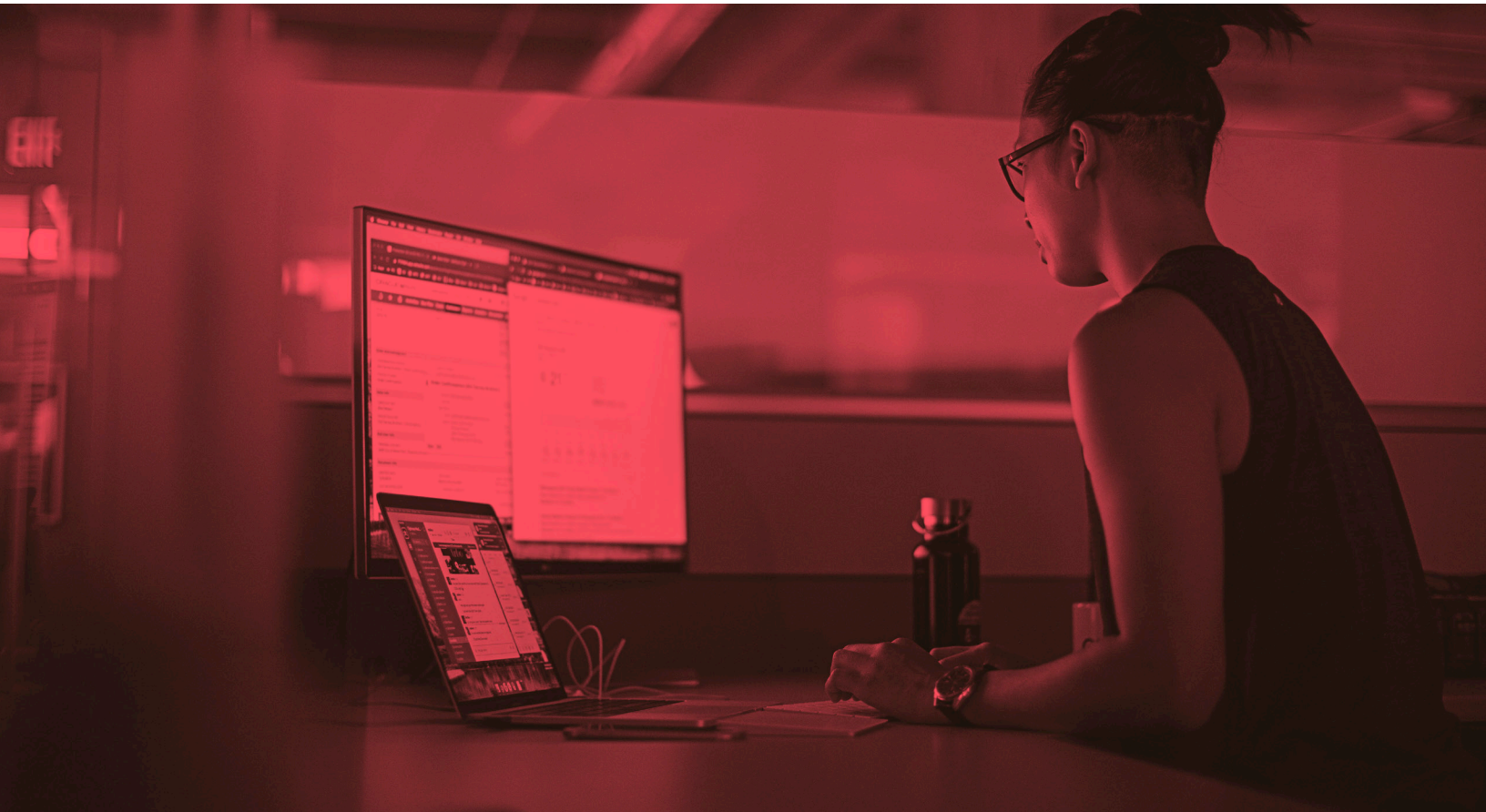
This is where Jamf Pro can help without compromising security. It's the most common method for requesting and creating certificates.

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

Here's what the communication is like:

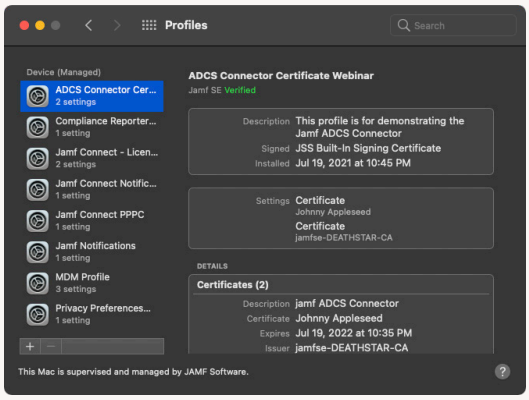
Proxy using Jamf Pro

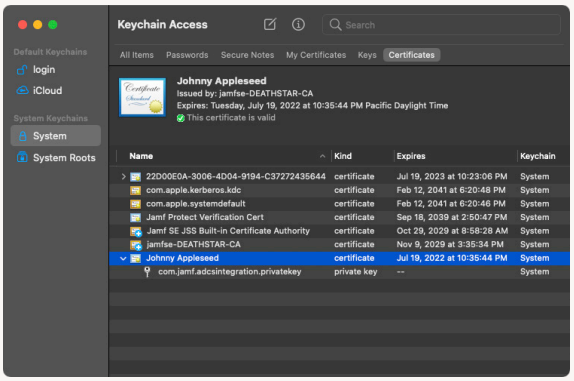


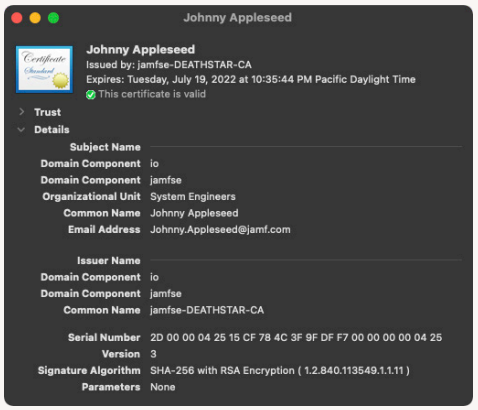


Where to find the deployed certificates

Once you have built out the configuration profile in Jamf Pro and scoped to the device. Let's take a peek at how it looks on a managed Mac. If you are on iOS, the certificates will be in Settings > General > Profiles.



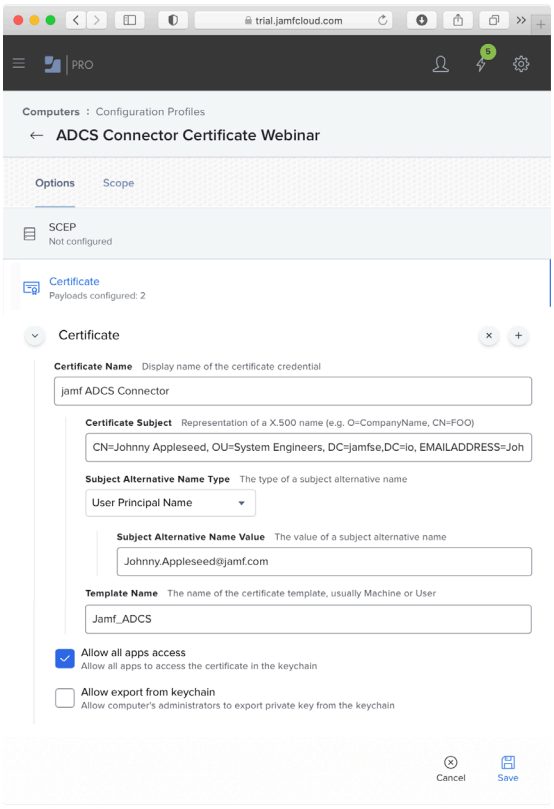




On the left, you can find the Configuration Profile that contains two certificates: the Root CA and the User Certificate that was generated with AD CS Connector. This is the complete chain, so the cert is automatically trusted and available to be used for WiFi, VPN, or other applications.

In the middle, you see what the two certificates look like in Keychain Access. The Root CA certificate has a gold icon with a blue plus indicating that it is trusted by all users. This happened because certificates installed via MDM with Jamf Pro are automatically trusted by the client device.

Double click the certificate to inspect it in more detail. Note that the User Certificate which has a blue icon is listed as "valid" with the green checkmark. This is because the Root CA is present. The two certificates did not need to be bundled together, but it can make things easier to manage.



So you have the certificate deployed... Now what?

For a completely Apple native solution, the Jamf administrator would configure a Configuration Profile that includes all the necessary settings to allow automatic connections to 802.1x WiFi or to VPN.

In other implementations, however, the end-user would manually select the Wi-Fi by the SSID and then be prompted to choose their identity from a drop-down. Authenticating to 802.1x Wi-Fi may require a username and password, depending on your network setup. Similarly, if your VPN client supports certificate authentication the user would be instructed to select their own user cert for authentication.

In the configuration above, the certificate will not be exportable by the user. This is a great way to increase the security of certificates.

How to start implementing a certificate deployment workflow in your organization

You will need:

- A managed Apple device enrolled in Jamf Pro
- 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 we can help you succeed in deploying certificates to your Apple devices. We will communicate with other teams in your organization to provide the context of 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 AD CS 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**