Cisco ISE and Certificates

How to Implement Cisco ISE and Server Side Certificates

Table of Contents

Certificate Usage

In a world of mobile devices, bring your own device IT models and networks without borders, certificates are fast becoming a common form of identification. The use of certificates does not need to invoke a fight or flight response in network administrators, and can be simplified quite a lot.

The most difficult concept for many to understand is the concept of a public certificate vs. a private certificate. Certificates are part of Public-Key cryptography or asymmetric encryption. Asymmetric means that the two communicating devices will each encrypt and decrypt the data with different encryption keys. The term "key" may sometimes be thrown around and interchanged with the term "certificate".

There will be two keys, a public and a private key.

1. **Public Key:** The public key is contained in the public certificate, and may be given to anyone in the world with whom you will communicate. In most cases,

2. **Private Key:** The private key should rarely leave the end-system. They represent the identity of that particular system, and if they are exposed and used by another entity – that other entity is now impersonating your identity.

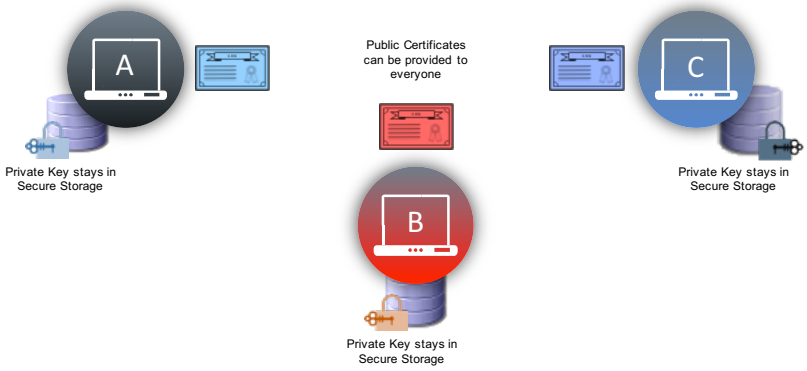

Figure 1. Public Certificates and Private Keys

Items that are encrypted using your public key may only be decrypted with your private key. Using Figure 1 as a reference, if endpoint C uses endpoint A's public key to encrypt some data, it can only be decrypted by endpoint A. Similarly, if B uses C's public key to encrypt data, that data may only be decrypted with C's private key.

So, what is a certificate?

A certificate is a signed document that represents an identity. When thinking of a certificate, try to relate it to a passport, a driver's license, or other personal identification card. That identification card is meant to represent you, and prove you are who you say you are. That certificate also contains the public key of that entity, so anyone with the public certificate will be able to encrypt data that only the certificate owner can decrypt.

This section will discuss how certificate-based authentications actually work. When presented with a certificate, an authentication server will perform the following checks (at a minimum):

1. Determine if a trusted authority has signed the Digital Certificate.

2. Examine both the start and end dates to determine if the certificate has expired.

3. Verify if the certificate has been revoked. (Could use OCSP or CRL check)

4. Validate that the client has provided proof of possession.

Let's examine the above 4 items one at a time:

Determine if a Trusted Authority has Signed the Digital Certificate

The signing of the certificate really has two parts:

The first part is that the certificate must have been signed correctly (following the correct format, etc). If it is not, it will be discarded immediately.

The second part is the public certificate of the signing Certificate Authority (CA) must be in the list of trusted certificates (the trusted certificates store), and it must be trusted for purposes of authentication. When using Cisco ISE, a copy of the signing CA's public certificate must be stored at **Administration > System > Certificates > Certificate Store** and it will need to have the "Trust for client authentication" use-case.

Where Are Certificates Used with ISE?

Certificates are employed often in a network implementing Secure Access. The certificates are used to identify the Identity Services Engine (ISE) to an endpoint as well as to secure the communication between that endpoint and the ISE node. The certificate is used for all HTTPS communication as well as the Extensible Authentication Protocol (EAP) communication.

HTTPS communication using the ISE certificate:

Every web portal with ISE version 1.1.0 and newer is secured using HTTPS (SSL encrypted HTTP communication). Examples include, but are not limited to:

- Administrative Portal
- Sponsor Portal & Guest Portals
- BYOD and Client Provisioning Portal (CPP)
- MyDevices Portal
- Mobile Device Management (MDM) Portal

Figure 2 describes the SSL encrypted process when communicating to the Admin portal.

Figure 2. HTTPS to Admin Interface

EAP Communication:

Certificates are used with nearly every possible EAP method. The main examples include:

- EAP-TLS
- PEAP
- EAP-FAST

With tunneled EAP methods such as PEAP and FAST, Transport Layer Security (TLS) is used to secure the credential exchange. Much like going to an HTTPS web site, the client establishes the connection to the server, which presents its certificate to the client. If the client trusts the certificate, the TLS tunnel is formed. The client's credentials are not sent to the server until after this tunnel is established, thereby ensuring a secure exchange. In a Secure Access deployment, the client is a supplicant, and the server is an ISE Policy Services node. Figure 3 describes an example using PEAP.

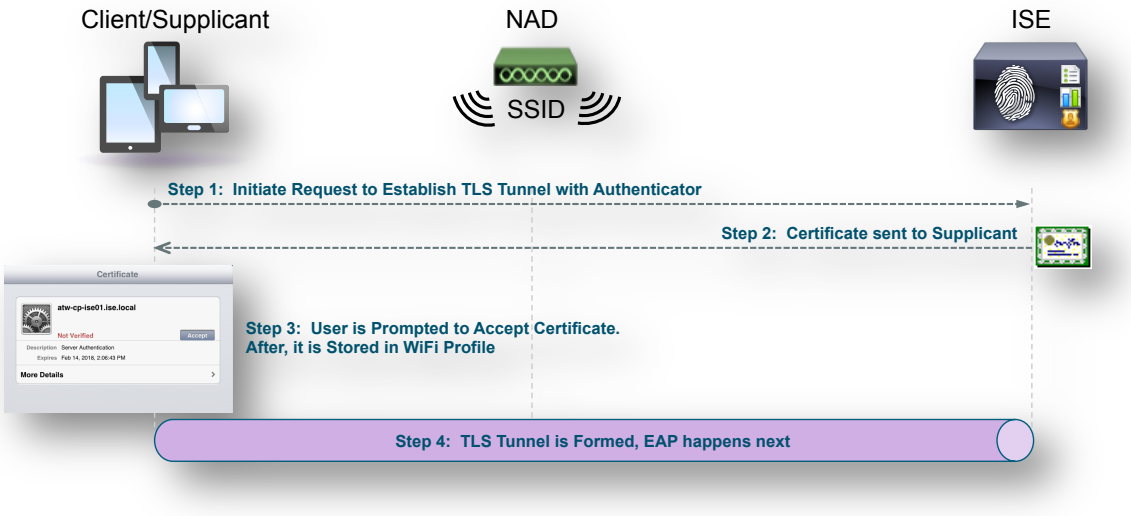

Figure 3. PEAP Example

As seen in Figures 2 and 3, regardless of where the certificates are being used, the basic functionality is the same. The client must trust the server, and the keys from within the certificates are used to encrypt and decrypt the communication. The concept of Trust is a very important one. Whenever working with certificate based authentications, a fundamental question to always ask yourself is: "Does the client trust the server" and "does the server trust the client".

Note: In many cases, the server is not required to trust the client certificate. Clients may also be configured to not require the server certificate to be trusted. All of the options are configuration choices.

Certificate Trust

Certificates are similar to signed documents. When a client communicates to a secure server, the server will send it's public certificate down to the client to be verified. The client will examine the certificate to determine if it should be trusted. The client is examining the certificate signer, and other attributes of the certificate, such as the subject.

The certificate signer:

When establishing a secure connection, the client will validate that the signer of the certificate is a trusted authority. If the client does not trust the certificate signer, a warning with be displayed, such as the one displayed in Figure 4. In Figure 4, the client (Firefox browser) is establishing a secure connection to an ISE admin portal. The certificate used to secure that portal is a self-signed certificate (signed only by ISE itself and not by a known authority), and therefore the browser does not trust that signer.

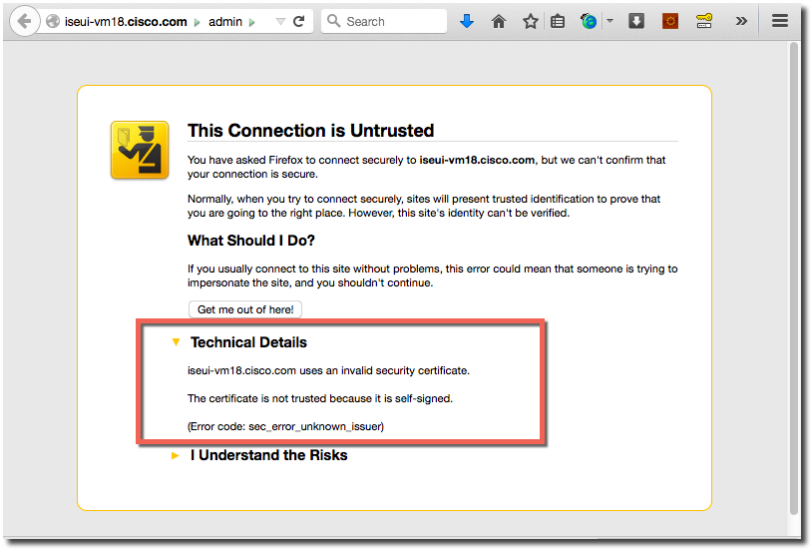

Figure 4. Untrusted Signer

The certificate subject:

The certificate is created with a specific subject. That subject defines the entity it was created to protect. For example, if you examine the certificate that is used with https://www.cisco.com you will see a representation similar what is displayed in Figure 5. The certificate used has a subject stating that this certificate was issued for securing www.cisco.com (the CN value of the certificate's subject).

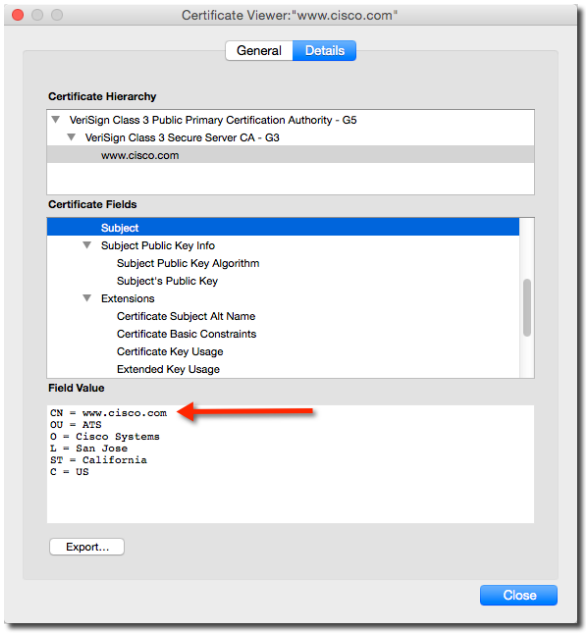

Figure 5. Certificate from https://www.cisco.com

In addition to a subject, a certificate may also have a subject alternative name (SAN). That extension to the certificate is designed to allow a single certificate be used to protect more than one fully qualified domain name (FQDN). Another way to look at that same statement is that many different FQDNs may point to the exact same secure site.

Using the certificate from www.cisco.com as the example, there are values listed in the SAN field, such as: www1.cisco.com, www2.cisco.com and others.

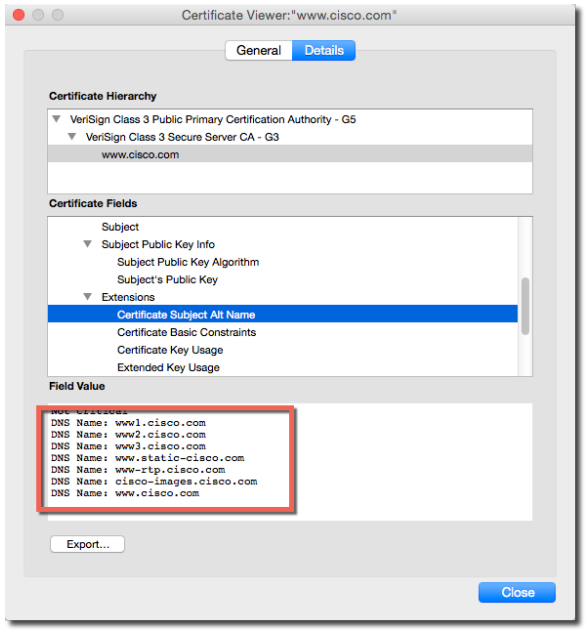

Figure 6. Subject Alternative Names

What Certificate Values Should be Used with an ISE Deployment?

With an ISE deployment, the administrator has a few choices related to using a single certificate for all identities or a mixture of different certificates, no more than one for each identity.

That is a confusing statement. Let's explain it a bit more. Each ISE node could have many different identities.

- **Admin Identity:** An ISE node has to identify itself to the other ISE nodes in an ISE cube (also called an ISE deployment). The Policy Administrative Node (PAN) must have secure bi-directional communication to all the Policy Service Nodes (PSNs) and the Monitoring and Troubleshooting Node (MnT) for policy synchronization and management communication. This same identity is used when an administrator connects to the administrative portal of an ISE node.

  The identity that must be protected for the Admin use-case is the FQDN of the ISE node itself. Therefore the subject or subject alternative name must match the FQDN of the ISE node.

- **EAP Identity:** An ISE node has to identify itself to the EAP (dot1x) clients that are connecting to the network. This is securing the layer-2 EAP communication, and therefore the name of the identity does not have to be DNS resolvable, and does not have to match the name of the ISE node itself.

  The identity that must be protected could be the FQDN of the ISE node itself, or another value such as "*aaa.security.demo.net*" or "*psn.ise.security.demo.net*"

- **Sponsor Portal Identity:** When using sponsored Guest services in an ISE deployment, a sponsor portal is used. That sponsor portal must have a friendly name (an HTTP host header) that is used to uniquely identify the portal itself, such as "*sponsor.security.demo.net*".

  The identity that must be protected is the friendly name. Therefore the certificate used on the sponsor portal must have a subject or subject alternative name that matches the friendly name.

- **MyDevices Portal:** When ISE is configured for Bring Your Own Device (BYOD), there is a MyDevices portal that end-users log into to manage their registered devices. Just like the Sponsor portal, the MyDevices portal requires a friendly name (an HTTP host header) that is used to uniquely identify the portal itself, such as "mydevices.security.demo.net".

  The identity that must be protected is the friendly name. Therefore the certificate used on the MyDevices portal must have a subject or subject alternative name that matches the friendly name.

- **pxGrid Identity:** When ISE is configured to be a pxGrid controller, it requires a certificate with both server and client extended key usages (EKU's). The name does not have to be DNS resolvable, so the subject and SAN fields are not necessarily important. However, a wildcard value may **not** be used with a pxGrid certificate at all (wildcards are covered in detail later in this document).

- **Guest Portal Identity:** There can be one or many portals used for guest access and centralized web authentication (CWA). As with all ISE portals, each one will need to identify itself and protect the communication to and from the portal with a certificate.

  In many guest cases, including CWA, there is an automatic redirection that is occurring to the FQDN of the ISE PSN itself.

  The identity that must be protected is the PSN(s) that are hosting the guest portals, and any friendly names that may be used. Therefore the certificate used for any portal must have a subject or subject alternative name that matches all the names it will be protecting.

Figure 7 illustrates an example of different certificates and what they may secure:

- Both PSN's admin certificates are being used to secure not only its communication with the PAN, but also the Guest portal for CWA.
- The sponsor portal is being protected with a certificate with sponsor.securitydemo.net in the certificate subject. The same certificate is being used for the sponsor portal on both PSN1 and PSN2.
- PSN1 and PSN2 are using their own EAP certificate for the securing of EAP communications.
- The PAN is using its admin certificate to protect both administrative communication to the PSNs as well as to secure the administrative GUI.

Note: This is just one example to try & solidify your understanding of where certificates are being used.

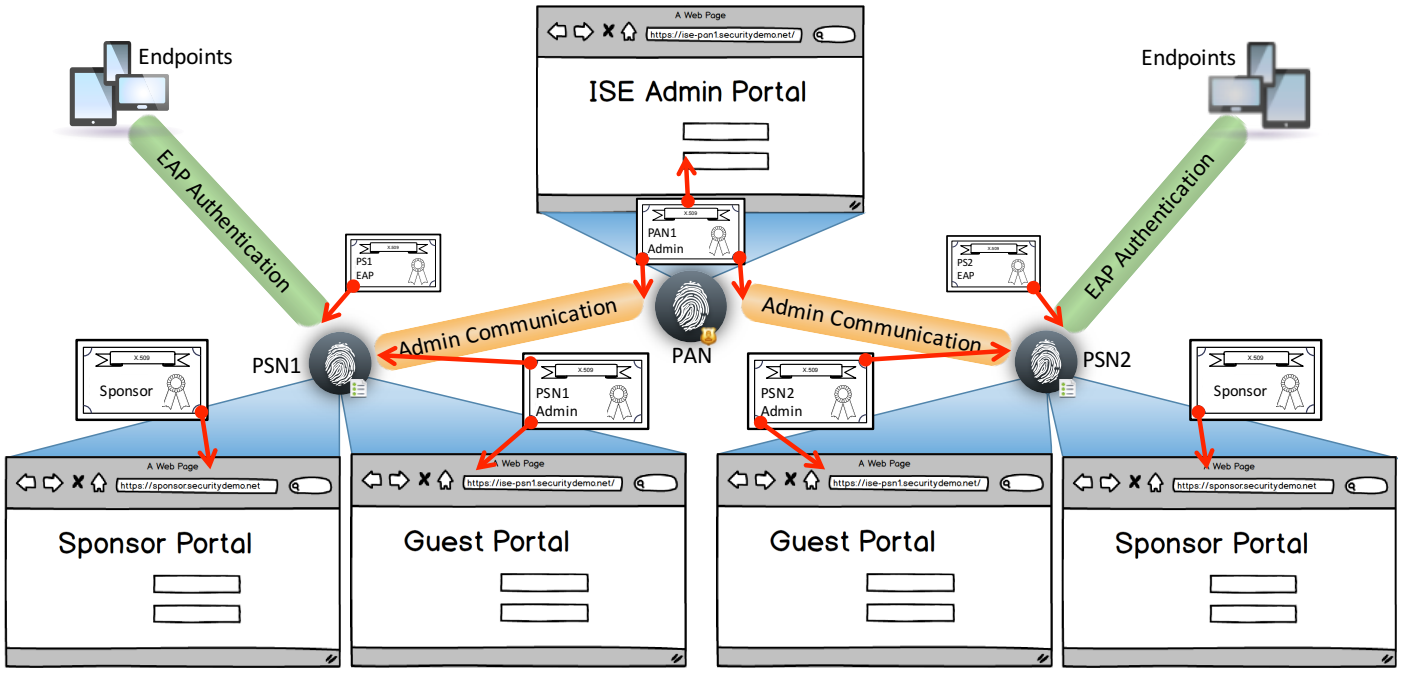

Figure 7. Sampling of Certificates and What They Secure

The ISE administrator's choice for how to break up these communications is very flexible. They can use one certificate per ISE node to secure all the different identities, they can use individual certificates for each service, they could use a single certificate and copy that certificate to each node in the deployment, or any combination of the above. The following sections will detail and illustrate three common ways of building ISE certificates.

Example 1: Single Certificate per Node. Used for all Services

This method uses a single certificate per each ISE node (PAN, MnT, PSN). That certificate will be used for the admin, EAP, pxGrid functions as well as securing all portals. To accomplish that each certificate should be configured for:

- Certificate Subject CN will contain the ISE node's FQDN
- No Subject Alternative Name is needed for the PAN or MNT if they are dedicated nodes
- For the PSNs, the Certificate Subject Alternative Name will contain:
    - ISE node's FQDN
    - Friendly name for the Sponsor Portal
    - Friendly name for the MyDevices Portal
- If pxGrid will be used, both server & client authentication extended key usages (EKUs) are required.

Figure 8 shows an example.

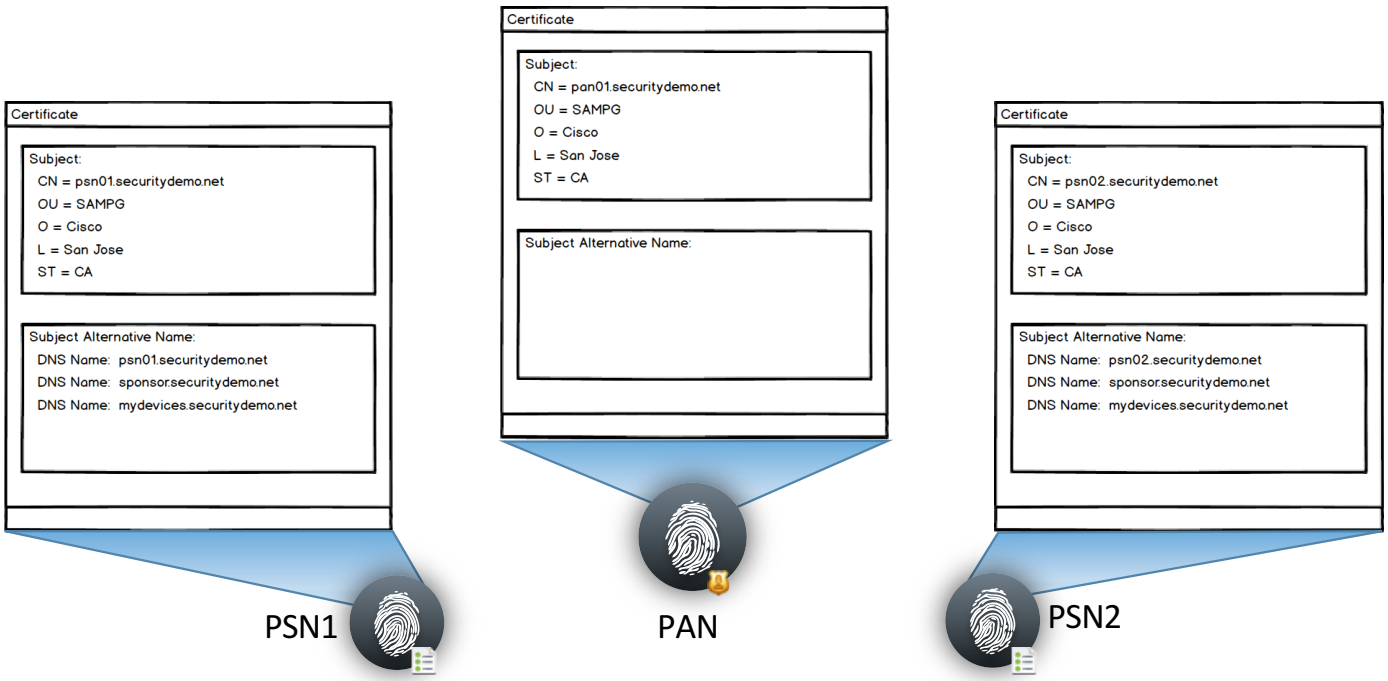

Figure 8. Model 1: Each Node has it's own Certificate

There are benefits and drawbacks to this model, as outlined in Table 1.

Table 1. Pro's and Con's of a Single Certificate per Node that is being Used for all Services

Pro's	Con's
Follows the security best practices of a unique certificate per node.	Many endpoints must accept the certificate for each PSNs EAP communication
Use of SAN's is fairly easy in ISE 1.2+	Are not using a different certificate on each portal.
Keeps certificate management rather easy	The same certificate is used for admin that guests and employee users will be exposed to.

Example 2: Multiple Certificates per Node. One for Each Service

This method uses a single certificate per function, per node. To accomplish that each certificate should be configured for:

Admin Certificate (All ISE nodes get one):

- Certificate Subject CN will contain the ISE node's FQDN
- No Subject Alternative Name is needed
- May be signed by internal (non-public) CA
- May be self-signed, although not recommended

pxGrid Certificate (All ISE nodes get one if you are using pxGrid):

- Certificate Subject CN will contain the ISE node's FQDN
- No Subject Alternative Name is needed
- *Should* be signed by public CA
- May be signed by internal (non-public) CA
- May be self-signed, although not recommended
- Must have both client and server authentication EKU's

EAP Certificate (All PSN's get one):

- Certificate Subject CN will contain the ISE node's FQDN
- No Subject Alternative Name is needed
- Should be signed by public CA

Portal Certificate(s) (One or more per PSN):

- Certificate Subject CN will contain the ISE node's FQDN
- Should be signed by public CA
- The Certificate Subject Alternative Name will contain:
    - ISE node's FQDN
    - Friendly name for each portal protected by the certificate
        - i.e: The FQDN for the MyDevices Portal
        - i.e: The FQDN for the Sponsor Portal

Figure 9 shows an example.

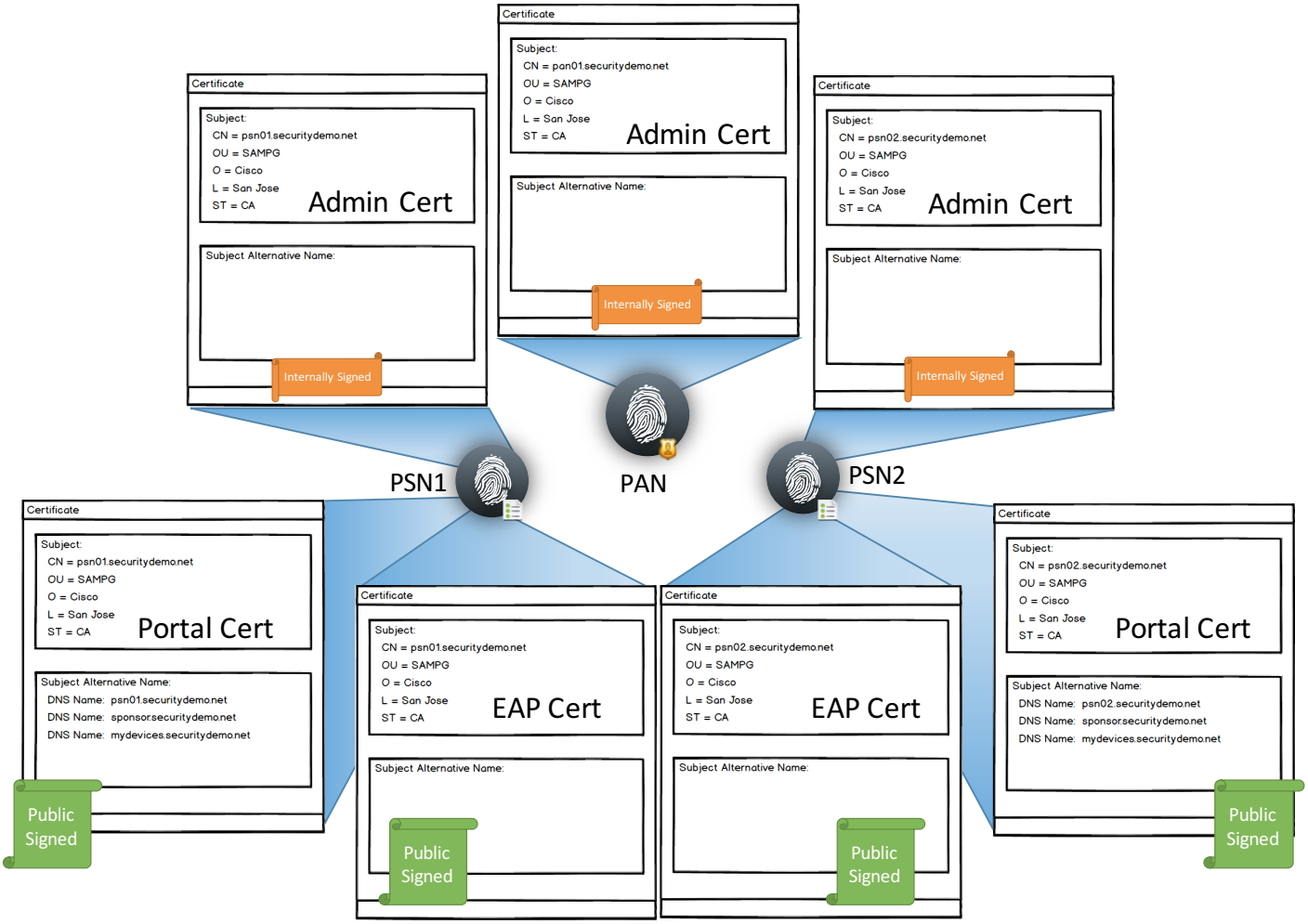

Figure 9. Model 2: Different Certs per Function

There are benefits and drawbacks to this model, as outlined in Table 2.

Table 2. Pro's and Con's of a Separate Certificate per Node and per Service

Pro's	Con's
Follows Security Best Practices of unique certificates per node, and goes further by providing unique certificates per node & per function.	More certificates to manage per ISE nodes than Method 1
	Can be expensive to manage that many signed certificates
	Many endpoint types must accept the certificate for each PSNs EAP communication

Example 3: Using the same certificate on all PSNs

This method uses the same private and public key-pair on all the ISE Nodes. This is often used for environments where there are a lot of BYOD-type devices. The main reason to follow this model is to ensure that the same exact certificate is used on all PSNs; so the BYOD devices will already trust any EAP server they must authenticate to.

To accomplish this method, generate one certificate signing request (CSR) on a single ISE node. After binding the signed certificate to the private key, export the resulting key pair & import it on all the other nodes. Configure the single certificate for:

- Certificate Subject CN will contain a single FQDN, such as ise.securitydemo.net
- The Certificate Subject Alternative Name will contain:
  - The subject CN, such as ise.securitydemo.net
  - Every ISE node's actual FQDN as SAN entries
  - Friendly name for the Sponsor Portal
  - Friendly name for the MyDevices Portal
- Certificate needs both client and server authentication EKUs

Figure 10 shows an example.

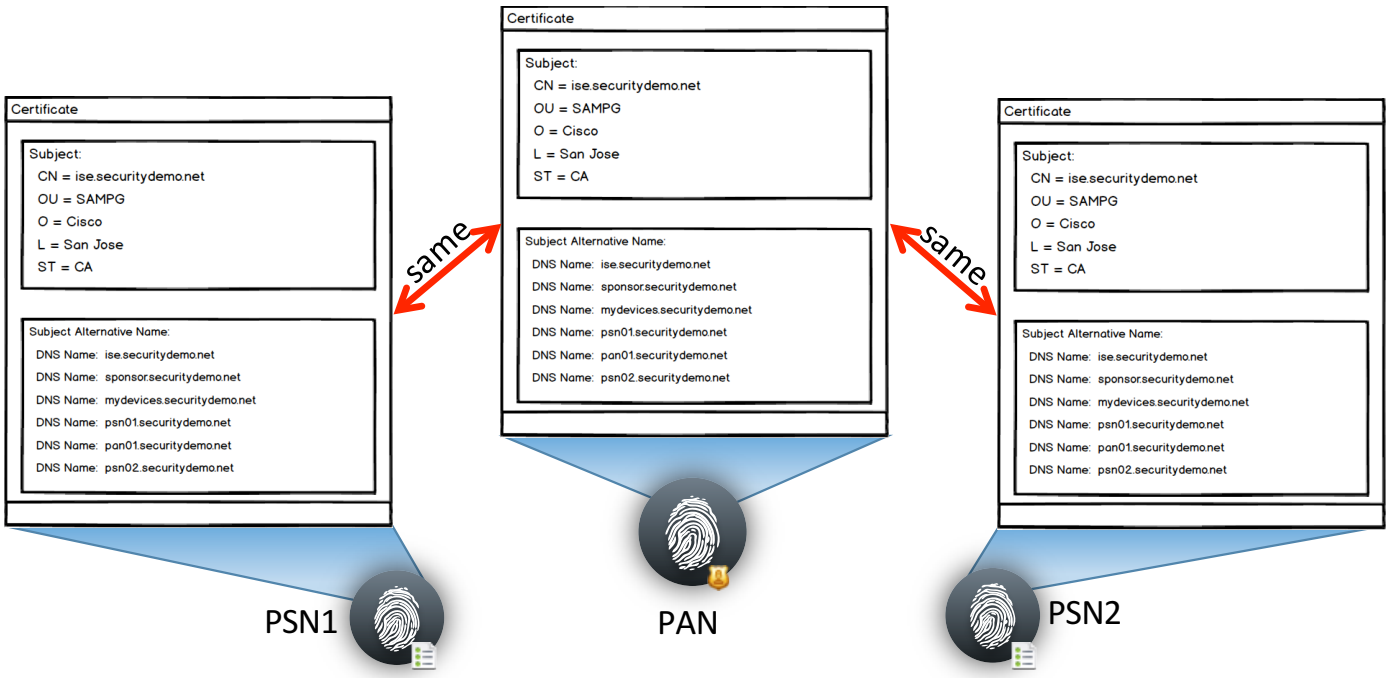

Figure 10. Model 3: Same Certificate on all Nodes

There are benefits and drawbacks to this model, as outlined in Table 3.

Table 3. Pro's and Con's of a Separate Certificate per Node and per Service

Pro's	Con's
BYOD type endpoints will not have to manually accept each cert for EAP authentications	Security Best-Practices of unique certificates per identity are broken. Each ISE node appears identical
Management is easy	Using a single certificate for admin and end-user facing

	functions
Certificate costs are kept down	If a new PSN node is ever added in the future, the certificate on every node will need to be updated to include the new FQDN

This concludes the examples. If it is not obvious, the ISE administrator is not limited to only these three examples, and is free to mix & match to fit whatever model is deemed most appropriate for the actual environment.

The next sections will discuss another model altogether, the use of wildcard and what we are calling "wildSAN" certificates.

Wildcard Certificates

What is a Wildcard Certificate?

A wildcard certificate is one that uses a wildcard notation (an asterisk and period before the domain name) and allows the certificate to be shared across multiple hosts in an organization. An example CN value for a wildcard certificate's Subject Name would look like the following: *.securitydemo.net

If you configure a Wildcard Certificate to use *.securitydemo.net, that same certificate may be used to secure any host whose DNS name ends in ".securitydemo.net", such as:

- aaa.securitydemo.net
- psn.securitydemo.net
- mydevices.securitydemo.net
- sponsor.securitydemo.net

A wildcard is only valid in the host field of the fully qualified domain name (FQDN). In other words, *.securitydemo.net would **not** match ise.aaa.securitydemo.net, because the wildcard value was not in the host portion of the FQDN.

Figure 11 shows an example of using a wildcard certificate to secure a web site (specifically, the web interface of an ISE node). Notice in figure 11 that the URL entered into the browser address bar is "atw-lab-ise01.woland.com", but the certificate's common name is "*.woland.com".

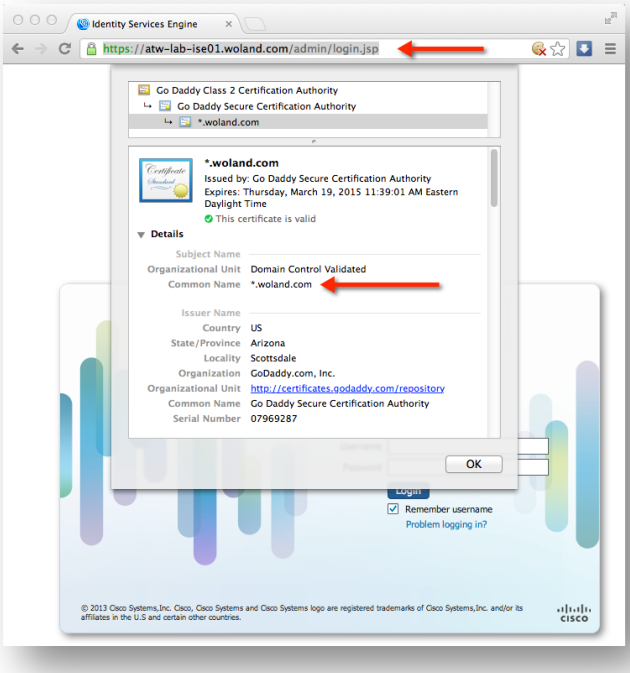

Figure 11. Example Wildcard Certificate

Note: Wildcard certificates secure communications in the same manner as a regular certificate, and requests are processed using the same validation methods.

Why use Wildcard Certificates?

There are a number of reasons to implement wildcard certificates with an ISE deployment. Ultimately, those who choose to use them, do so to ensure the end-user experience is as seamless as possible, especially given the vast difference and variety of endpoints.

Benefits of Wildcard Certificates

Some examples of benefits of wildcard certificate usage are:

1) Cost savings. Certificates signed by a 3rd-party Certificate Authority can be costly, especially as the number of servers increase. Wildcard certificates may be used on all nodes of the ISE Deployment (also referred to as the "ISE Cube").

2) Operational efficiency. Wildcard certificates allow all Policy Service Node (PSN) EAP and web services to share the same certificate. In addition to significant cost savings, certificate administration is also simplified through a "create once, apply to many" model.

3) Reduced authentication errors. Wildcard certificates address issues as seen with Apple iOS devices where the client stores trusted certificates within the profile, and does not follow the iOS Keychain where the signing root is trusted. When an iOS client first communicates to a PSN it will not explicitly trust the PSN certificate, even though a trusted Certificate Authority has signed the certificate.

   Using a wildcard certificate, the certificate will be the same across all PSNs, so the user will only need to accept the certificate once and successive authentications to different PSNs should proceed without error or prompting.

4) Simplified supplicant configuration. For example, Microsoft Windows supplicant with PEAP-MSCHAPv2 and server cert trust enabled often requires specification of each server certificate to trust, or user may be prompted to trust each PSN certificate when client connects using a different PSN. With wildcard certificates, a single server certificate can be trusted rather than individual certificates from each PSN.

Ultimately, wildcard certificates result in an improved user experience. Less prompting and more seamless connectivity will translate into happier users and increased productivity.

Drawbacks to Wildcard Certificates

There can be a number of benefits using wildcard certificates, but there are also a number of security considerations related to wildcard certificates including:

1) Loss of auditability and non-repudiation

2) Increased exposure of the private key

3) Not as common or as well understood by admins

Although considered less secure than assigning a unique server certificate per ISE node, cost and other operational factors often outweigh the security risk and necessitate the need to offer this as an option to our customers in their ISE deployments. Note, that other security devices like the ASA also support wildcard certificates.

You must always be careful when deploying wildcard certificates. For example if you create a certificate with *.securitydemo.net and an attacker is able to recover the private key, that attacker can spoof any server in the securitydemo.net domain. Therefore it is considered a best practice to partition the domain space to avoid this type of compromise.

To address this possible issue and to limit the scope of use, wildcard certificates may also be used to secure a specific subdomain of your organization. Just add an asterisk (*) in the subdomain area of the common name where you want to

specify the wildcard. For example, if you configure a wildcard certificate for *.ise.securitydemo.net, that same certificate may be used to secure any host who's dns name ends in ".ise.securitydemo.net", such as:

- psn.ise.securitydemo.net
- mydevices.ise.securitydemo.net
- sponsor.ise.securitydemo.net

Wildcard Certificate Compatibility

Wildcard certificates are most commonly constructed with the wildcard listed as the canonical name (CN) of the subject field of the certificate itself, such as what is shown in Figure 11. ISE version 1.2 and newer support this manner of construction, however not all endpoint supplicants will support the wildcard in the subject of the certificate.

All Microsoft native supplicants tested (including Windows Mobile) do not support wildcards in the subject of the certificate. The use of another supplicant, such as Cisco's AnyConnect Network Access Manager (NAM), will allow the use of wildcard characters in the subject field. Another option is to use special wildcard certificates like DigiCert's Wildcard Plus that is designed to work on incompatible devices by including specific sub-domains in the Subject Alternative Name of the certificate.

For more information on Microsoft's support of wildcard certificates, see here: http://technet.microsoft.com/en-US/cc730460

Making Wildcards Work: the "WildSAN" Method

Although the limitation with Microsoft supplicants may appear to be a deterrent to using wildcard certificates, there are alternative ways to construct the certificate that allow it to work with all devices tested with Secure Access, including the Microsoft native supplicants.

Instead of constructing the subject to include the wildcard values, you may put those wildcard values into the Subject Alterative Name (SAN) field instead. The SAN field maintains an extension designed for the checking of the domain name, dNSName. See RFCs 6125 and 2128 for more detail, and a small excerpt from RFC 6125 is displayed in figure 12. This method is often referred to as the "WildSAN" method.

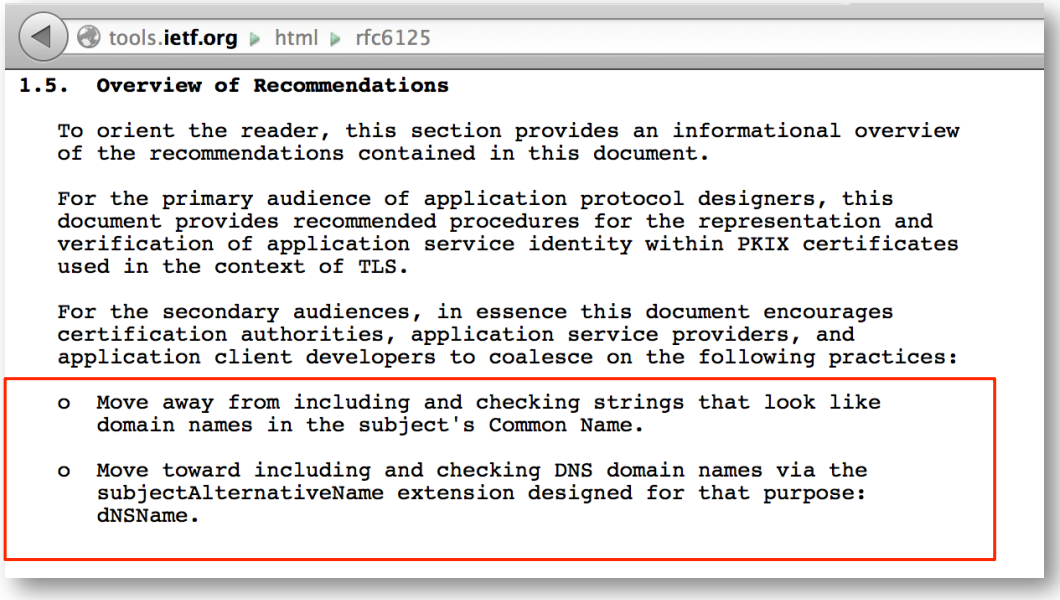

Figure 12. Excerpt from RFC 6125

ISE Support for Wildcard Certificates

ISE added support for wildcard certificates in version 1.2. Prior to version 1.2, ISE will perform verification of any certificates enabled for HTTPS to ensure the CN field matches the host Fully Qualified Domain Name (FQDN) exactly. If the fields did not match, the certificate could not be used for HTTPS.

This restriction exists because prior to version 1.2, ISE would use that CN value to replace the variable in the url-redirect AV pair string. For all Centralized Web Authentication (CWA), on-boarding, posture redirection and more, the CN value would be used.

Beginning in ISE version 1.2, the behavior has been modified to use the hostname and domain-name as it is defined in the underlying operating system (ADE-OS) configuration of ISE, instead of relying on the CN field of the certificate. The following CLI output example is showing the hostname and domain-name of an ISE node.

```
atw-tme-ise134/admin# show run | i hostname
hostname atw-tme-ise134
atw-tme-ise134/admin# show run | i domain
ip domain-name securitydemo.net
```

Constructing the WildSAN Certificate

Since we know we must insert the wildcard value into the Subject Alternative Name (SAN) field of the certificate as a workaround for Microsoft native supplicants, we are left with two main ways to construct the certificate:

Option 1: Leave the canonical name (CN) field of the subject blank and insert the wildcard into the SAN field.

While this appears to work perfectly well with most private Certificate Authorities, such as the Microsoft Active Directory CA, the majority of Public authorities will not allow the creation of a certificate without the CN value.

Figure 13 shows an example of a valid wildcard certificate without the CN field.

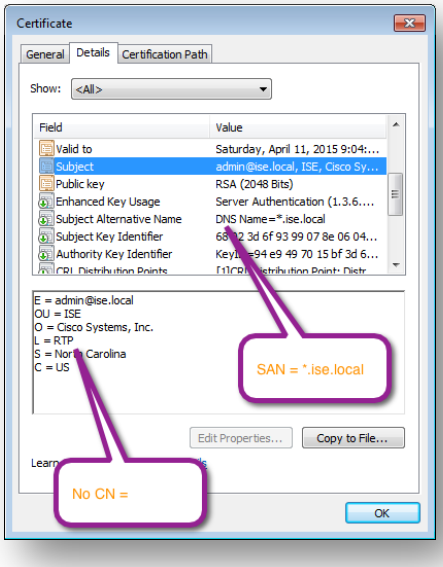

Figure 13. Example of Option 1

Option 2 (Cisco Preferred Best Practice): Use a generic hostname for the CN field of the subject, and insert both the same generic hostname and the wildcard value into the SAN Field.

This method has been successful with the majority of tested public Certificate Authorities, such as Comodo.com and SSL.com. With these public CA's the type of certificate to request is the "Multi-Domain Unified Communications Certificate" (UCC).

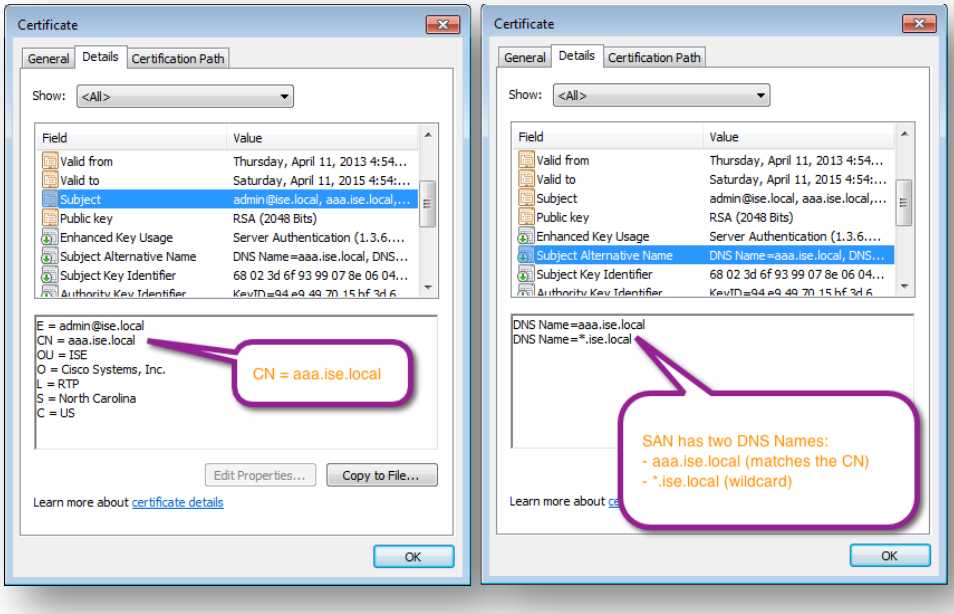

Figure 14. Example of Option 2

Note: With both option 1 and 2, the resulting wildcard certificate only needs to used on the ISE nodes running Policy Services, it is not required to be used on the Policy Administration Nodes (PAN) or Monitoring and Troubleshooting (MnT) nodes.

Implementing WildSAN Certificates

Now that we have reviewed wildcard certificates and their usage with ISE, we will walk through the creation of a wildcard certificate following the best practice of using a generic hostname for the CN field of the subject, and insert both the same generic hostname and the wildcard value into the SAN Field.

Generate the WildSAN Certificate

There are a few ways to import a wildcard or wildSAN certificate into ISE version 1.3. This procedure will follow what we expect to be the most common approach, which is to create the Certificate Signing Request (CSR) within the ISE administrative interface and submit that CSR to the signing Certificate Authority (CA). The resulting signed public key will be bound to the CSR on ISE.

The final private and public key-pair will be exported from the first ISE node, and imported on the other nodes in the deployment.

Create the Certificate Signing Request (CSR)

From the first ISE node, navigate to the certificates section of the administrative GUI. For dedicated Policy Services Nodes, the path will be "**Administration > System > Certificates > Certificate Signing Requests**".

Step 1 Click **Generate Certificate Signing Request (CSR)**
Step 2 Under usage, click the "**Certificate(s) will be used for**" drop-down and select **EAP Authentication**
Step 3 Select the "**Allow Wildcard Certificates**" check box
Step 4 For the Certificate Subject, replace the $FQDN$ variable with a generic FQDN for the ISE PSNs.
Step 5 Select at least two DNS Names under the Subject Alternative Name (SAN) section

One of the DNS Names must match the CN value from Step 4.

The other DNS Name should be the wildcard value.

Figure 15 displays a sample CSR for a WildSAN certificate.

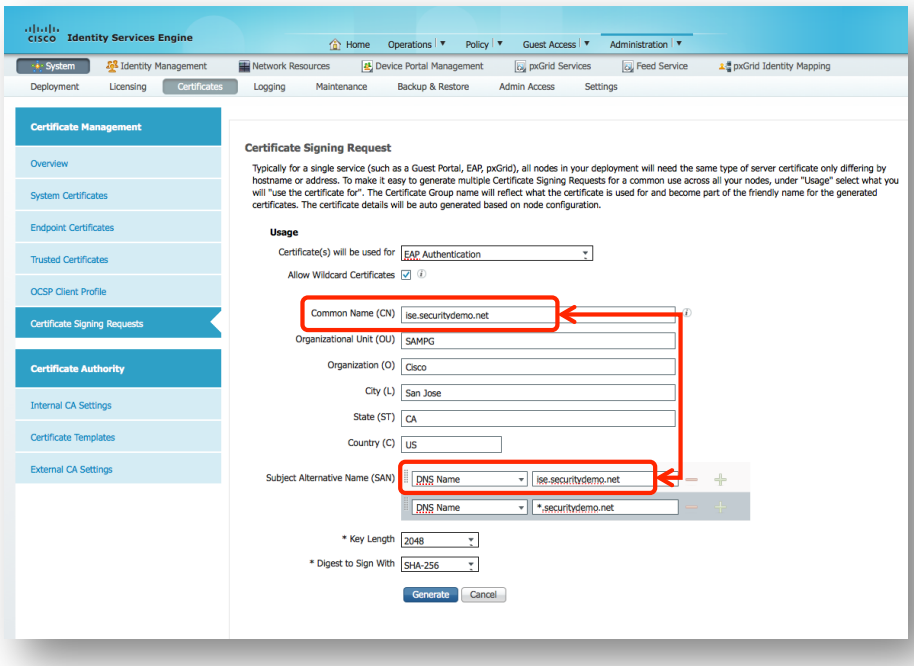

Figure 15. Example WildSAN Certificate Signing Request

Step 6 Click **Generate**

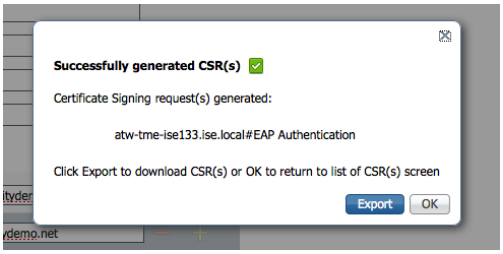

Figure 16. Successful CSR Generation

Step 7 Click **Export**, save the resulting .pem file to a location on your local system where it is retrieved easily.

Submit the CSR to the Certificate Authority

Now that the CSR has been exported, it needs to be submitted to a CA for signing.

Step 1 Open the CSR (.pem file saved to your local system) in your favorite text editor and copy all the text inclusive of the "-----BEGIN CERTIFICATE REQUEST-----" through "-----END CERTIFICATE REQUEST-----". Figure 16 shows an example.

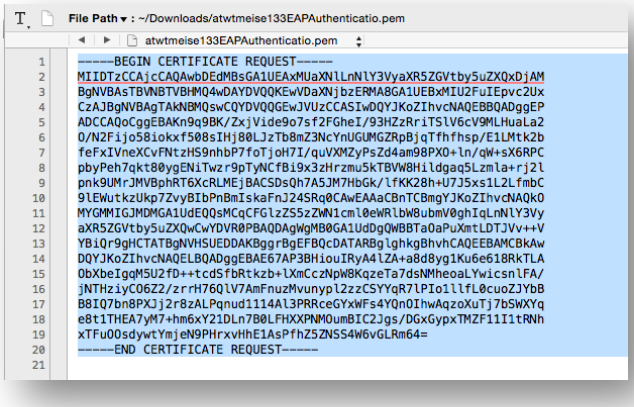

Figure 17. The Certificate Signing Request

Step 2 Paste the contents of the CSR into the certificate request on the chosen CA, such as seen in Figures 18 – 20.

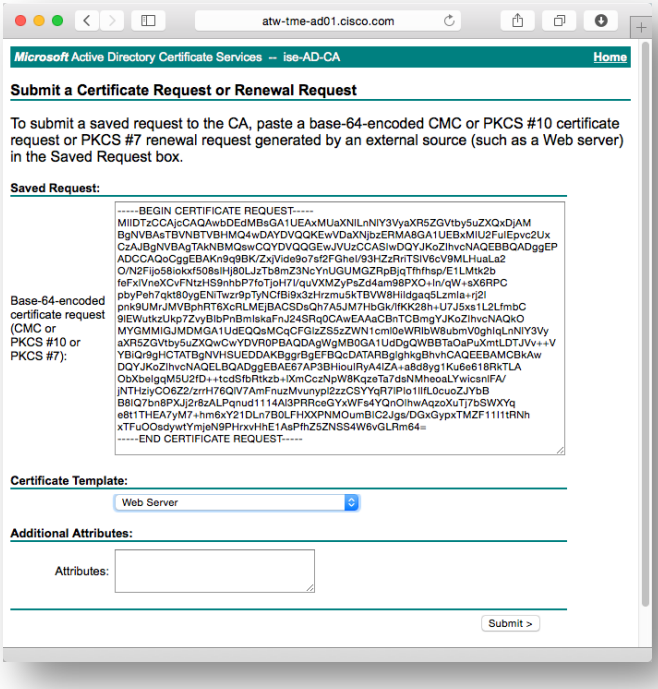

Figure 18. Paste the CSR into the Certificate Request Form – Active Directory CA

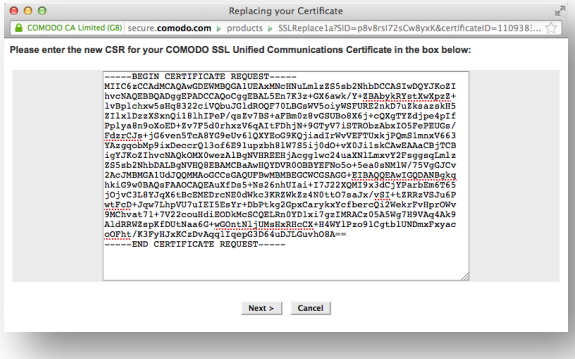

Figure 19. Paste the CSR into the Certificate Request Form – Public CA (Comodo.com)

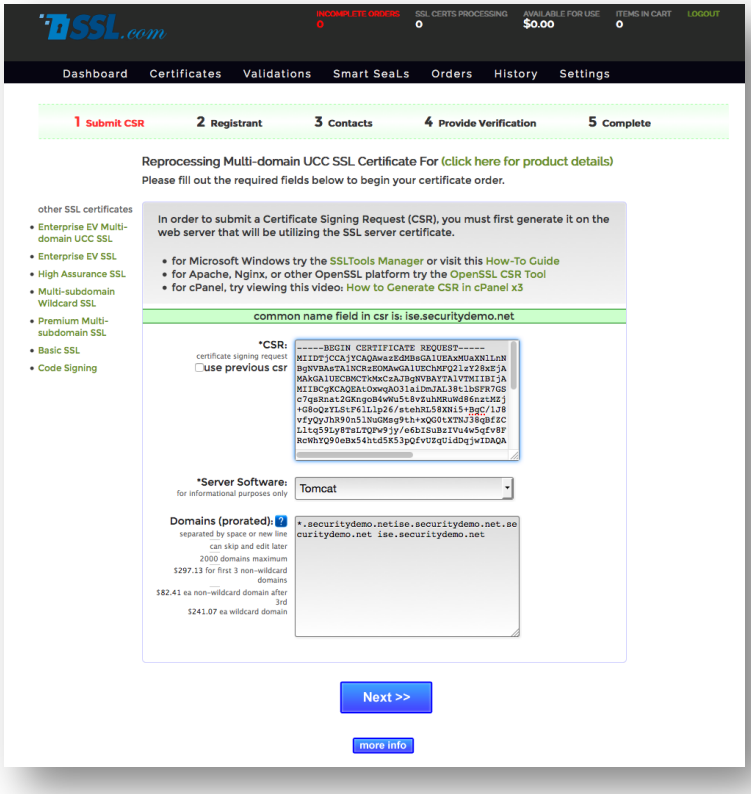

Figure 20. Paste the CSR into the Certificate Request Form – Public CA (SSL.com)

Step 3 Click **Next**, **Submit**, **Continue** or whichever button will allow you to proceed with the signing request.

A private certificate authority, such as the Active Directory CA, may immediately present you with a download page, as seen in Figure 21. Public CA's may require much more time to validate your permissions and then they will email you a link to download the signed certificate from your portal, or they may even email you with a zip file containing the signed certificate and the public certificates for the signing CA hierarchy, also seen in Figure 21.

Your job at this point is to download the signed certificate and the public certificates for all the CA's in the trust hierarchy. Then you will add those public certificates to the ISE trusted certificates store. Lastly, you will bind the Certificate signing request to the signed certificate that was sent to you.

There may be an option to accept a certificate chain, do not use a chain if possible. With the myriad of endpoint devices that connect to ISE, it is always best to use individual certificates with Base 64 encoding (PEM format).

Step 4 Download the resulting signed certificate from the portal or from the email. Always use Base 64 (PEM) encoding, as seen in Figure 21.

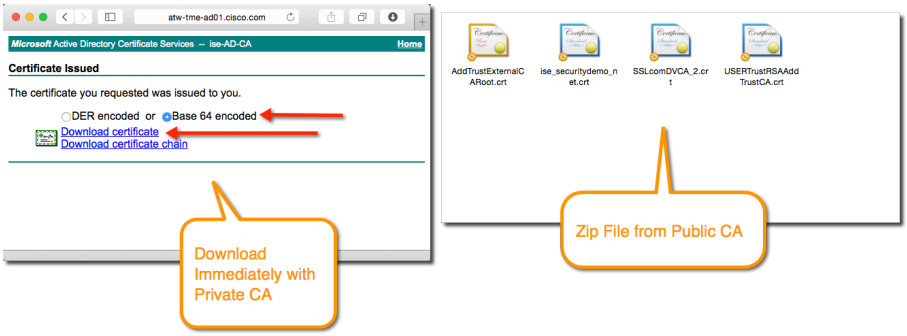

Figure 21. Download the Signed Certificate or Unpack the .Zip File

Step 5 Download the CA's public certificates. This can often be from the home page of the certificate authority as seen in Figure 22.

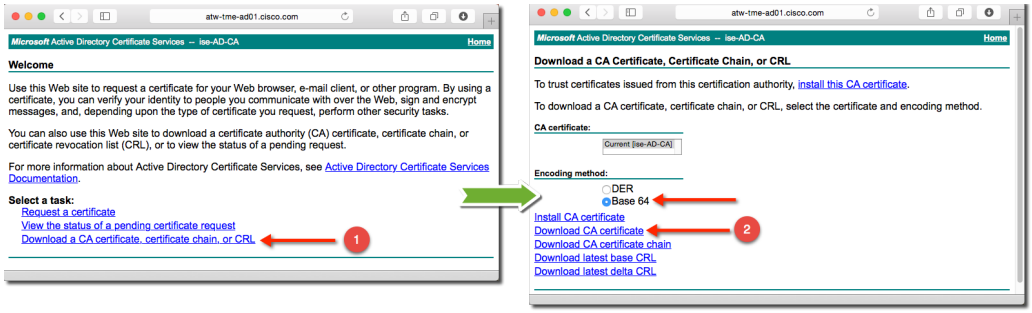

Figure 22. Download the CA Certificate

Import the CA Public Certificates to the Trusted Certificate Store

You will now install the CA public certificate(s) into ISE's Trusted Certificate store, as seen in Figure 23. This store maintains copies of the public certificate of any device that ISE "trusts".

Step 1 Within the ISE GUI, navigate to **Administration > System > Certificates**
Step 2 On the left-hand side, under Certificate Management: Click on **Trusted Certificates**.
Step 3 Click **Import**
Step 4 **Browse** for the public certificate file, as seen in Figure 23.
Step 5 Add a **Friendly Name** the display, as seen in Figure 23.
Step 6 Ensure the "Trust for authentication within ISE" is selected, as seen in Figure 23.
Step 7 (optional) If the CA will also issue endpoint certificates, then select "Trust for client authentication and syslog". If the CA is a public trusted root, then do not check the client authentication check-box.
Step 8 Click **Submit**.
Step 9 Repeat steps 8 through 13 for each certificate authority.

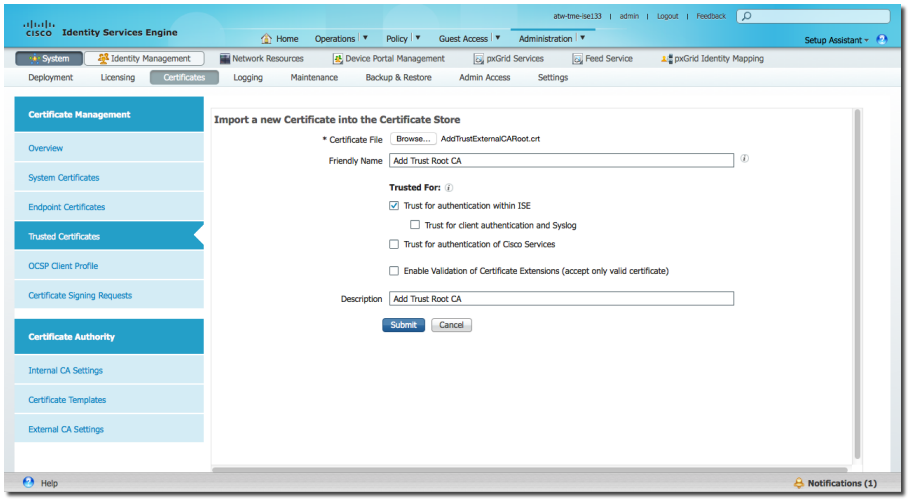

Figure 23. Importing a CA Public Certificate into the Trusted Certificates store

Bind the Newly Signed Certificate to the Signing Request

Now that ISE trusts the signing CA, it's time to bind the signed certificate to the certificate signing request within ISE.

From the ISE GUI:

Step 1 Navigate to **Administration > System > Certificates**
Step 2 Click on **Certificate Signing Requests**
Step 3 Select the certificate signing request as seen in Figure 24
Step 4 Click **Bind Certificate**
Step 5 Browse to the signed certificate
Step 6 If wildcards or the WildSAN method were used, click "**Allow Wildcard Certificates**" as seen in Figure 24
Step 7 Click **Submit**

The certificate is now bound to the EAP method. You will now go back into the signed certificate and add the other functions.

Step 8 Navigate to **Administration > System > Certificates > System Certificates**
Step 9 Select the signed certificate and click **Edit**
Step 10 Under the "Usage" section, select **Admin** and **Portal**.

Note: pxGrid cannot leverage certificates that use wildcard values. Do not select the pxGrid role if the certificate uses any wildcard values.

Step 11 When selecting **Portal**, the Certificate Group Tag drop down will appear. Version 1.3 of ISE does not allow certificate group tags to be re-assigned to a new certificate. Therefore, select **Add New…**
Step 12 Choose a name for your new certificate group tag
Step 13 Click **Save**

Reuse the Same Certificate Pair on other ISE Nodes

If you choose to do so, you could reuse the same certificate on the other ISE nodes. For more on reasons why, see the section titled "**Example 3: Using the same certificate on all PSNs**".

From the ISE GUI:

Step 1 Navigate to **Administration > System > Certificates > System Certificates**

Step 2 Select the new certificate
Step 3 Click **Export**
Step 4 Choose **Export Certificate and Private Key**
Step 5 Provide a password that will be used later when importing the certificate key-pair
Step 6 Click **Export**. Figure 24 shows the certificate being exported.

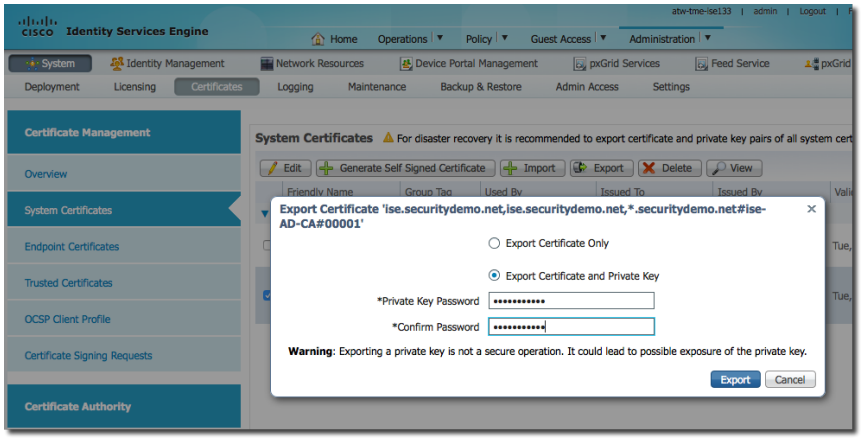

Figure 24. Export the Key-Pair

Step 7 The key-pair is exported as a zip file, save that zip file to a location that be accessed quickly.

Now that the key-pair has been saved, you will need to extract the zip file from step 7, so the two certificate files may be accessed individually.

On one of the remaining ISE nodes:

Step 8 Navigate to **Administration > System > Certificates > System Certificates**.
Step 9 Click **Import**.
Step 10 Select the correct node from the "**Select Node**" drop down.
Step 11 Click **Browse** for the Certificate File, and locate the certificate file from the zip file with the .pem extension (for example "isesecuritydemonet.pem")
Step 12 Click **Browse** for the Private Key File, and locate the private key file from the zip file with the .pvk extension (for example "isesecuritydemonet.pvk")
Step 13 Provide the password you created when exporting the certificate pair.
Step 14 Ensure the "**Allow Wildcard Certificates**" check box is enabled
Step 15 Choose the protocol for this certificate to be bound to: EAP, Admin, Portal (or all of them)
Step 16 Click **Submit**

Figure 25 shows the importing of the certificate. Repeat steps 8 through 15 for the remaining ISE nodes.

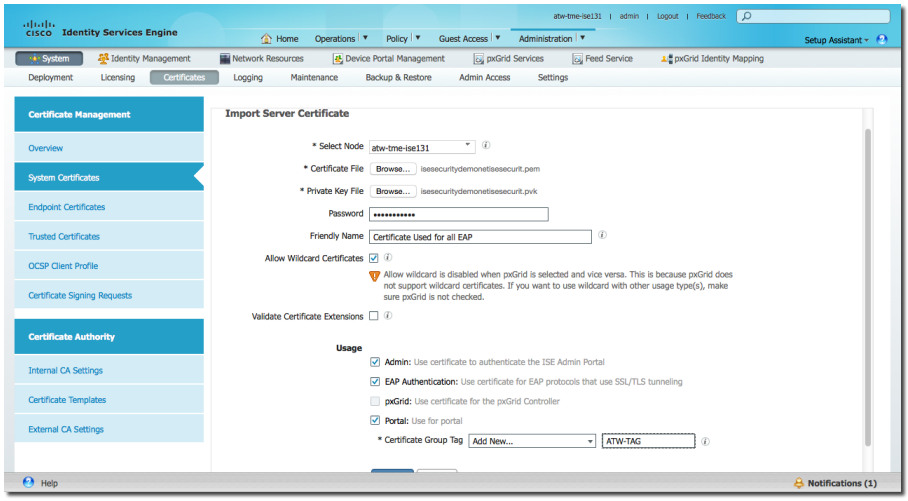

Figure 25. Importing the Certificate Pair