High Level View of Certificates and Authorities in CUCM

Contents

Introduction

The purpose of this document is to understand the basics of certificates and certificate authorities. This document compliments other Cisco documents that refer to any encryption or authentication features in Cisco Unified Communications Manager (CUCM).

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

This document is not restricted to specific software and hardware versions.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Conventions

Refer to Cisco Technical Tips Conventions for more information on document conventions.

Purpose of Certificates

Certificates are used between end points to build a trust/authentication and encryption of data. This confirms that the endpoints communicate with the intended device and have the option to encrypt the data between the two endpoints.

Define Trust from a Certificate's Point of View

The most important part of certificates is the definition of which end points can be trusted by your end point. This document helps you know and define how your data is encrypted and shared with the intended website, phone, FTP server, and so on.

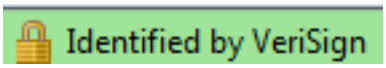
When your system trusts a certificate, this means that there is a pre-installed certificate(s) on your system which states it is 100 percent confident that it shares information with the correct end point. Otherwise, it terminates the communication between these end points.

A non-technical example of this is your driver's license. You use this license (server/service certificate) to prove that you are who you say you are; you obtained your license from your local Division of Motor Vehicles branch (intermediate certificate) who has been given permission by the Division of Motor Vehicles (DMV) of your State (Certificate Authority). When you need to show your license (server/service certificate) to an officer, the officer knows they can trust the DMV branch (intermediate certificate) and the Division of Motor Vehicles (certificate authority), and they can verify that this license was issued by them (Certificate Authority). Your identity is verified to the officer and now they trust that you are who you say you are. Otherwise, if you give a false license (server/service certificate) that was not signed by the DMV (intermediate certificate), then they will not trust who you say you are. The remainder of this document provides an in-depth, technical explanation of certificate hierarchy.

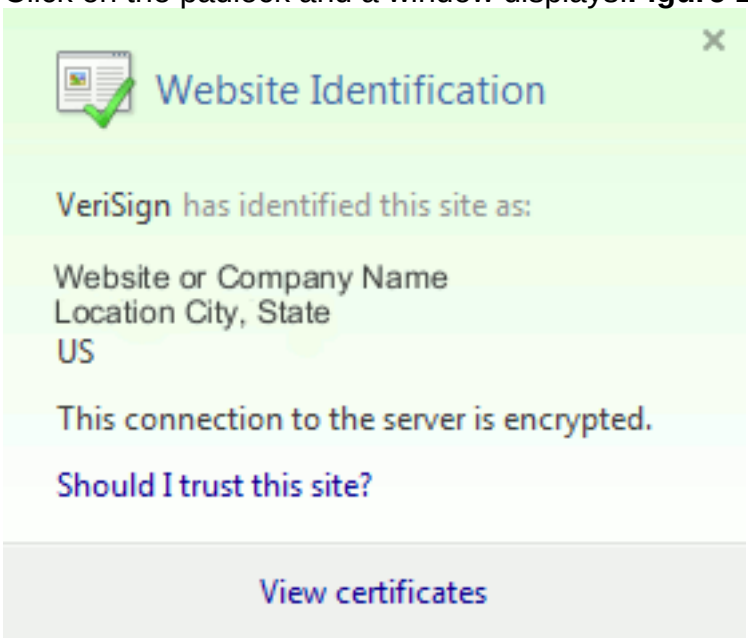
How Browsers Use Certificates

1. When you visit a website, enter the URL, such as http://www.cisco.com.
2. The DNS finds the IP address of the server that hosts that site.
3. The browser navigates to that site.

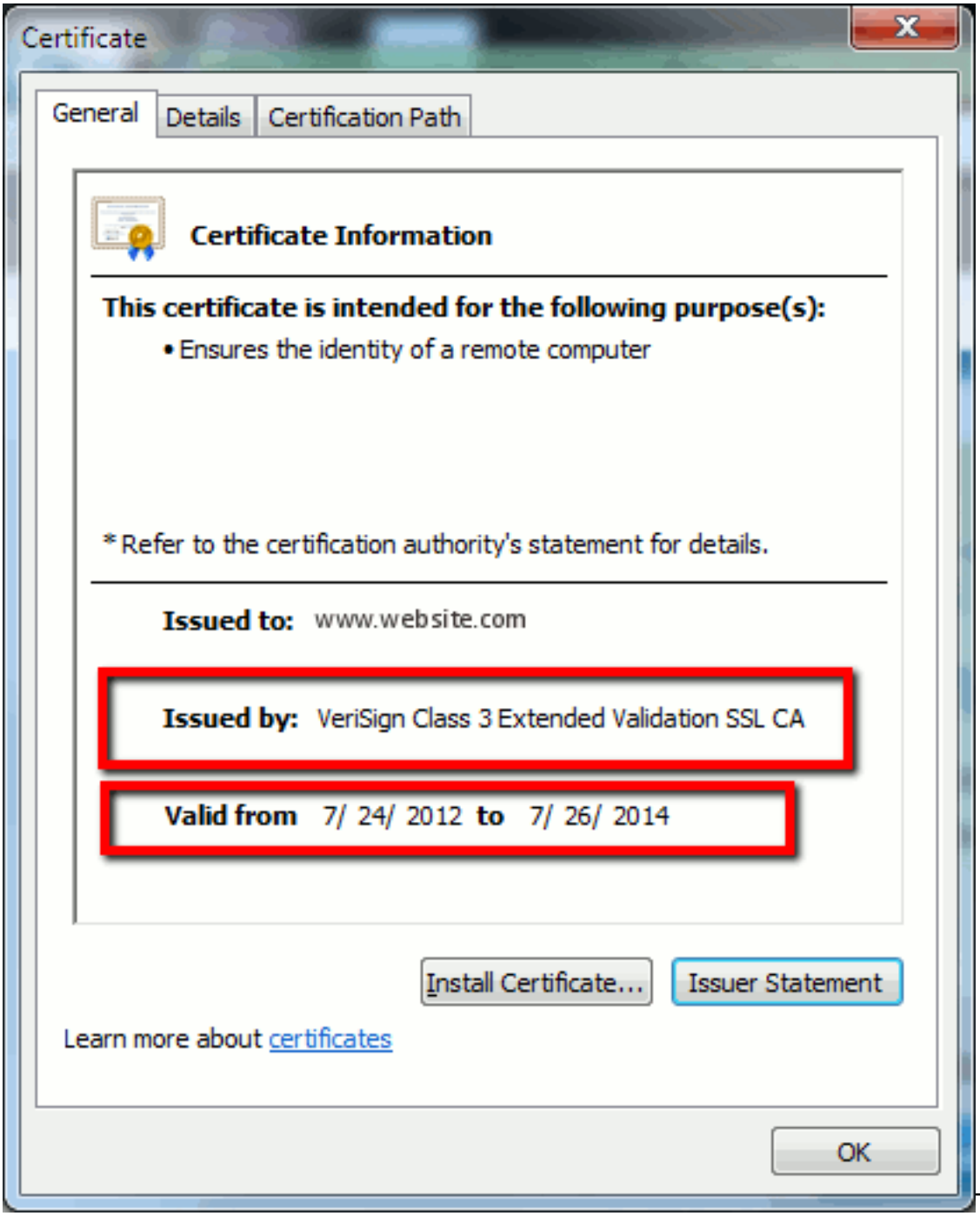
Without certificates, it is impossible to know if a rogue DNS server was used, or if you were routed to another server. Certificates ensure that you are properly and securely routed to the intended website, such as your bank website, where the personal or sensitive information you enter is secure.

All browsers have different icons they use, but normally, you see a padlock in the address bar like this: 

1. Click on the padlock and a window displays:**Figure 1: Website Identification**



2. Click on **View Certificates** to see the site's certificate as shown in this example:**Figure 2: Certificate Information, General Tab**



The highlighted information is important.**Issued by** is the Company or Certificate Authority (CA) that your system already trusts.**Valid from/to** is the date range that this certificate is usable. (Sometimes you see a certificate where you know you trust the CA, but you see the certificate is invalid. Always check the date so you know whether or not it has expired.)**TIP:** A best practice is to create a reminder in your calendar to renew the certificate before it expires. This prevents future issues.

The Differences Between PEM versus DER Certificates

PEM is ASCII; DER is binary. Figure 3 shows the PEM Certificate format.

Figure 3: PEM Certificate Example

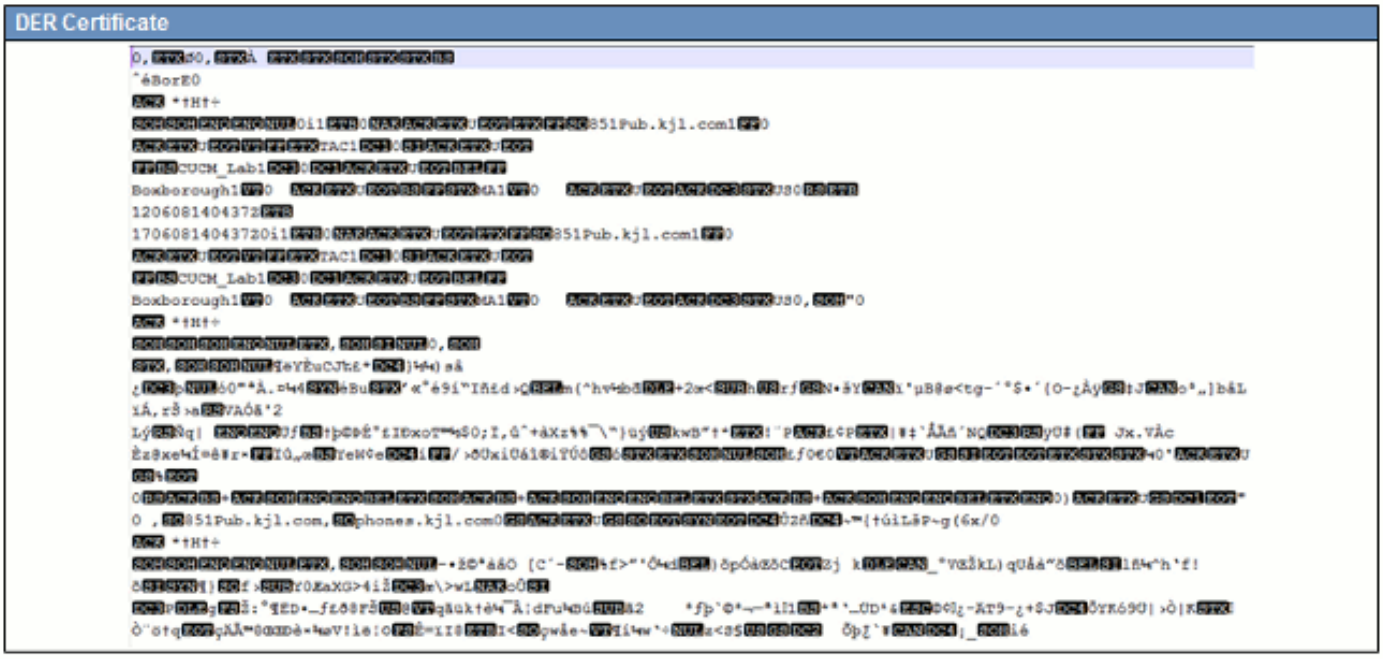
PEM Certificate

```
-----BEGIN CERTIFICATE-----
MIID2DCCAsCgAwIBAgIIDY2I6UJvckUwDQYJKoZIhvcNAQEFBQAwaTEXMBUGA1UE
AwwOODUxUHViLmtqbC5jb20xDDAKBgNVBAsMA1RBQzERMA8GA1UECgwIQ1VDTV9M
YWIxEzARBgNVBAcMCkJveGJvcm91Z2gxCzAJBgNVBAgMAk1BMQswCQYDVQQGEwJV
UzAeFw0xMjA2MDgxNDA0MzdaFw0xNzA2MDgxNDA0MzdaMGkxFzAVBgNVBAMMDjg1
MVB1Yi5ramwuY29tMQwwCgYDVQQLDANUQUMxETAPBgNVBAoMCENVQ01fTGFiMRMw
EQYDVQQHDApCb3hib3JvdWdoMQswCQYDVQQIDAJNQTELMAkGA1UEBhMCVVMwggEi
MA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC261nIdUNKiaMqFH29vClz4iC/
E/4A8zAiqsAupLw0FpDpQnUCkquw6Tntk0nxo2SbUQdtjyheaHa9YphkECsynDwa
aIEfcoMdTpWaWRjvJ7VCQPg8dGettLoklbSNe08tv8D/HYdKGG+zhF1i4kzvwYJy
ipthHlZB0+MnMglM/R7RcZ18oAUF3IMIhv6p3sm6o51J0HhvVJm9JDA7zyz7iCvg
WHolJa9ck338/R9rd0KUhioDIahQBqOiUAN8pYdgxcPxtE5REx7/3CMoDCBKeC5W
wGMJyHpAeGW8zaTqpXLXDM/7hJwIWWVXomUU7Qwvm/DceGnc4e6uaZ/a9B3zAgMB
AAGjgYMwgYAwCwYDVR0PBAQDAgK8MCcGA1UdJQQgMB4GCCsGAQUFBwMBBggrBgEF
BQcDAgYIKwYBBQUHAwUwKQYDVR0RBCIwIIIODUxUHViLmtqbC5jb22CDnBob251
cy5ramwuY29tMB0GA1UdDgQWBBTbWvEUfpl7hvrsTJpQfmcoNpB4LzANBgkqhkiG
9w0BAQUFAAOCAQEArZWeqarg4tagW0O0rQElZj6UJ9S8ZAcp9XDT4Iz1QwRaaiBr
EBhfulaMjmtMKXFV5eCU9QcPbPG8XmiRZiEg9Q8Wtn0OZpuPGlkwxmFYRz40aY4T
5lw+d0wVb9sPChNQEGccjjqwtstElYWDo/A4RoqdH0ALceP8a4bovK/CpmRGdb5C
+hqP4zIJs4P+YKmzrJeq7H8xCCqqkYXcRLkmG6mif78txFQ5lr8rJEoU1VlL8znc
fJvSfEsCfwnSqPaGcQTnxMOZOIyM0OjXvvhWIezrpk8cyj3vSTgXSTwO53flZX4L
tu28d5H3AHo8U6cfHRIJ1f6Yv2ClGBShXwFp6Q==
-----END CERTIFICATE-----
```

Figure 4 shows the DER Certificate.

Figure 4: DER Certificate Example

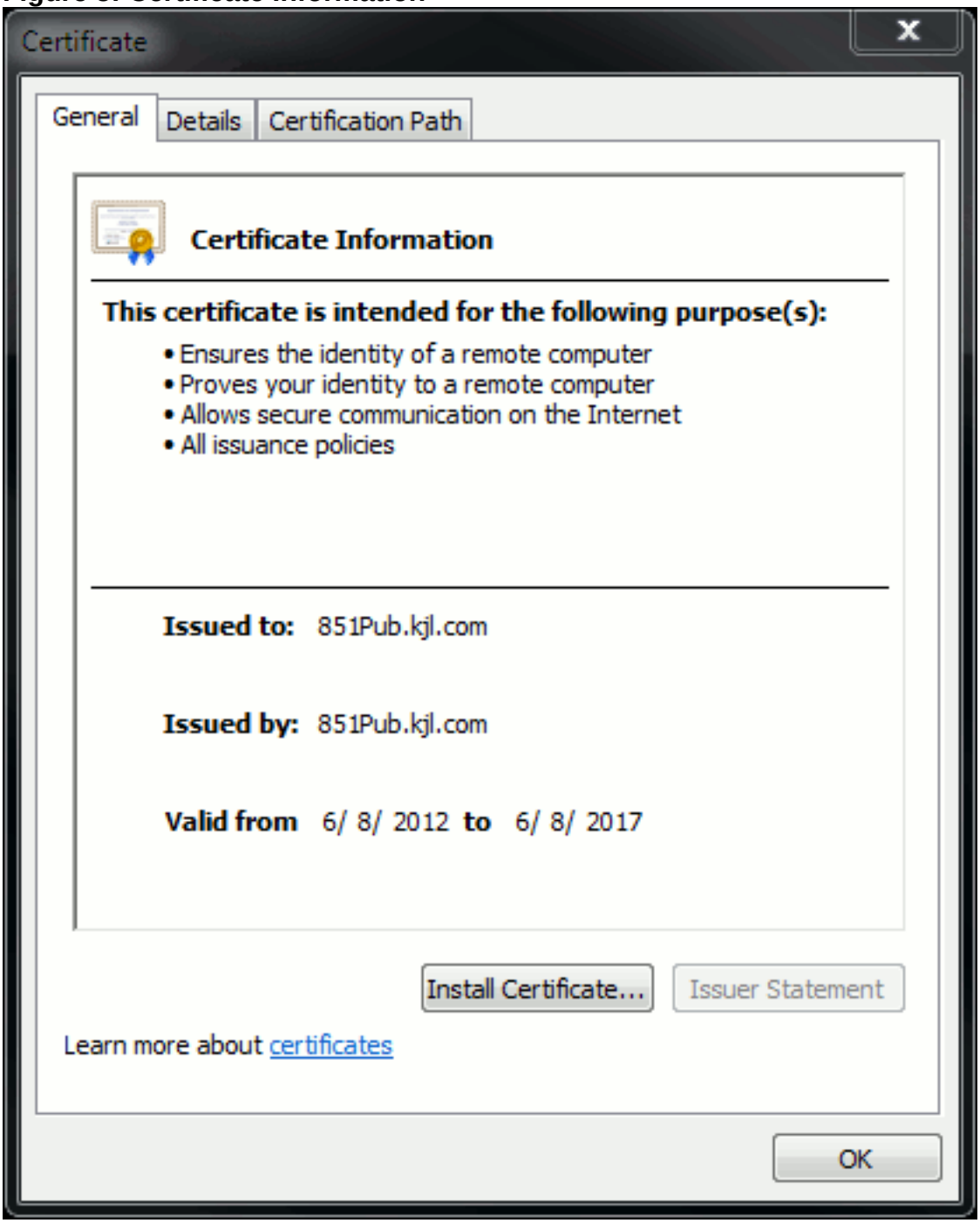
DER Certificate



Most CA companies like VeriSign or Thawt use PEM format to send the certificates to customers, because it is email-friendly. The customer should copy the entire string and include **-----BEGIN CERTIFICATE-----** and **-----END CERTIFICATE-----**, paste it into a text file, and save it with the extension .PEM or .CER.

Windows can read DER and CER formats with its own Certificate Management Applet and shows the certificate as shown in Figure 5.

Figure 5: Certificate Information

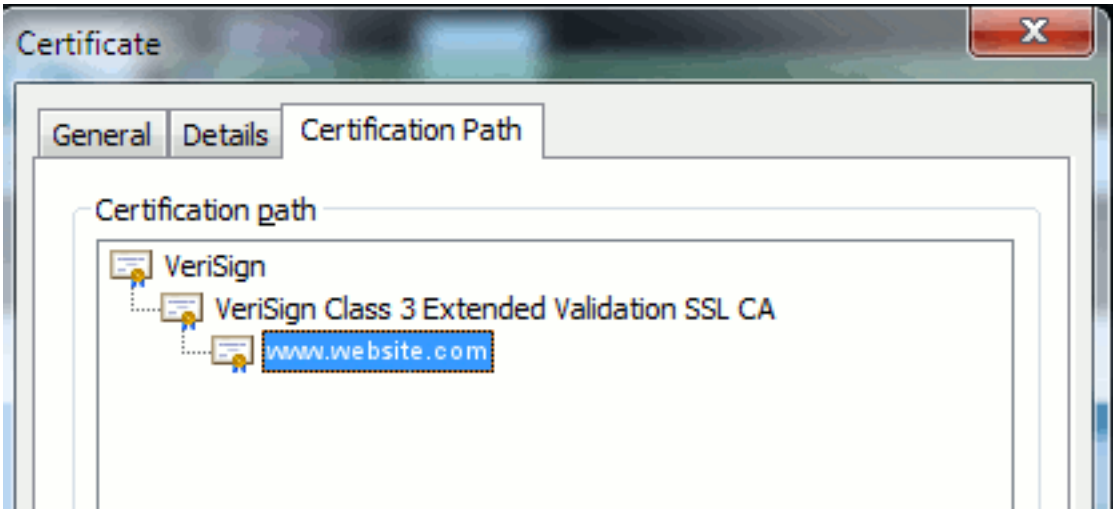


In some cases, a device requires a specific format (ASCII or binary). In order to change this, download the certificate from the CA in the needed format or use an SSL converter tool, such as https://www.sslshopper.com/ssl-converter.html.

Certificate Hierarchy

In order to trust a certificate from an end point, there must be a trust already established with a third party CA. For example, Figure 6 shows there is a hierarchy of three certificates.

Figure 6: Certificate Hierarchy



- **Verisign** is a CA.
- **Verisign Class 3 Extended Validation SSL CA** is an intermediate or signing server certificate (a server authorized by CA to issue certificates in its name).
- **www.website.com** is a server or service certificate.

Your end point needs to know that it can trust both the CA and intermediate certificates first before it knows that it can trust the server certificate presented by the SSL Handshake (details below). To better understand how this trust works, refer to the section in this document: **Define "Trust" from a Certificate's Point of View**.

Self-signed Certificates versus Third-party Certificates

The main differences between self-signed and third-party certificates are who signed the certificate, whether you trust them.

A self-signed certificate is a certificate signed by the server that presents it; therefore, the server/service certificate and the CA certificate are the same.

A third-party CA is a service provided by either a public CA (like Verisign, Entrust, Digicert) or a server (like Windows 2003, Linux, Unix, IOS) that controls the validity of the server/service certificate.

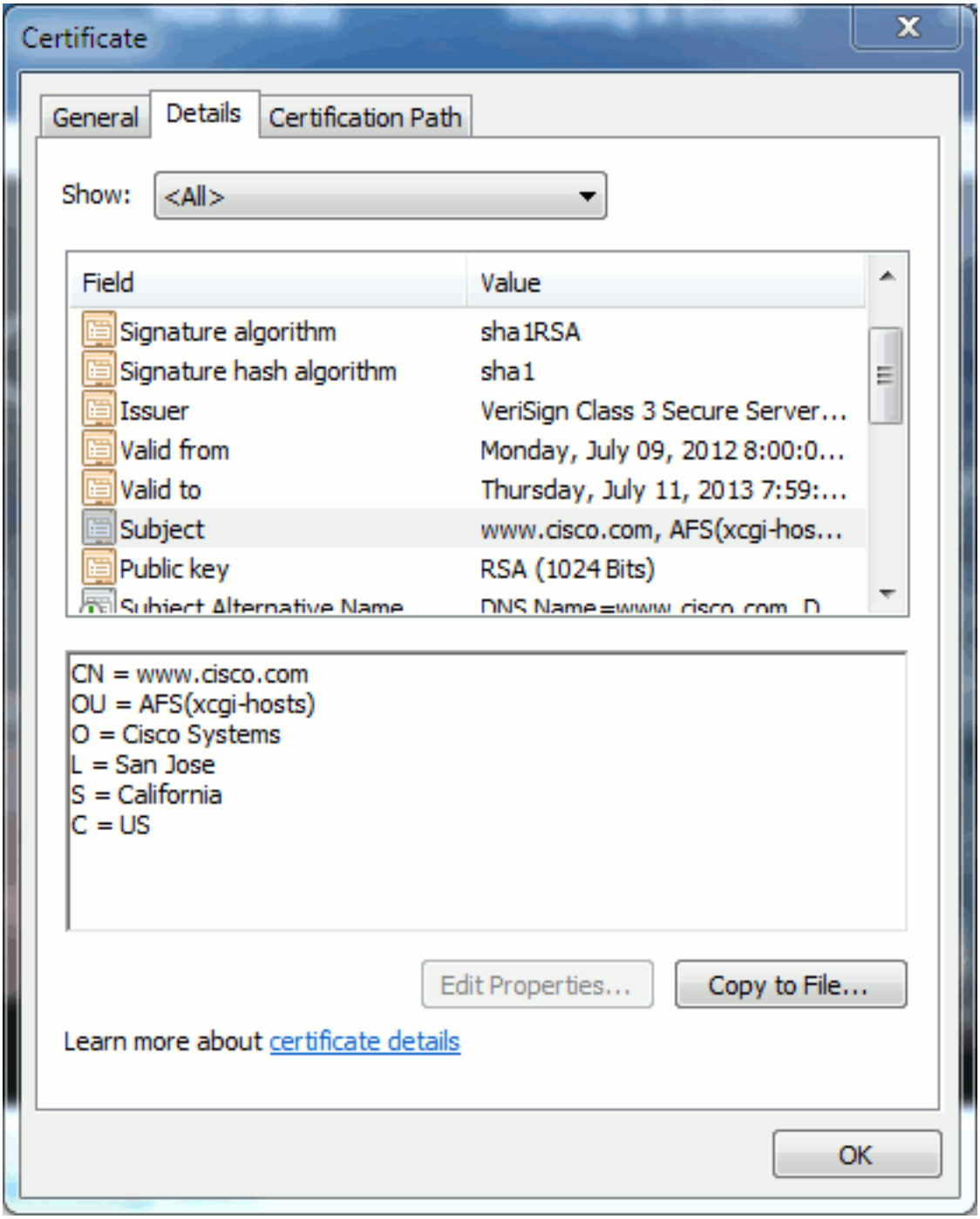
Each one can be a CA. Whether or not your system trusts that CA, is what matters the most.

Common Names and Subject Alternative Names

Common Names (CN) and Subject Alternative Names (SAN) are references to the IP address or Fully Qualified Domain Name (FQDN) of the address that is requested. For instance, if you enter https://www.cisco.com, then the CN or SAN must have www.cisco.com in the header.

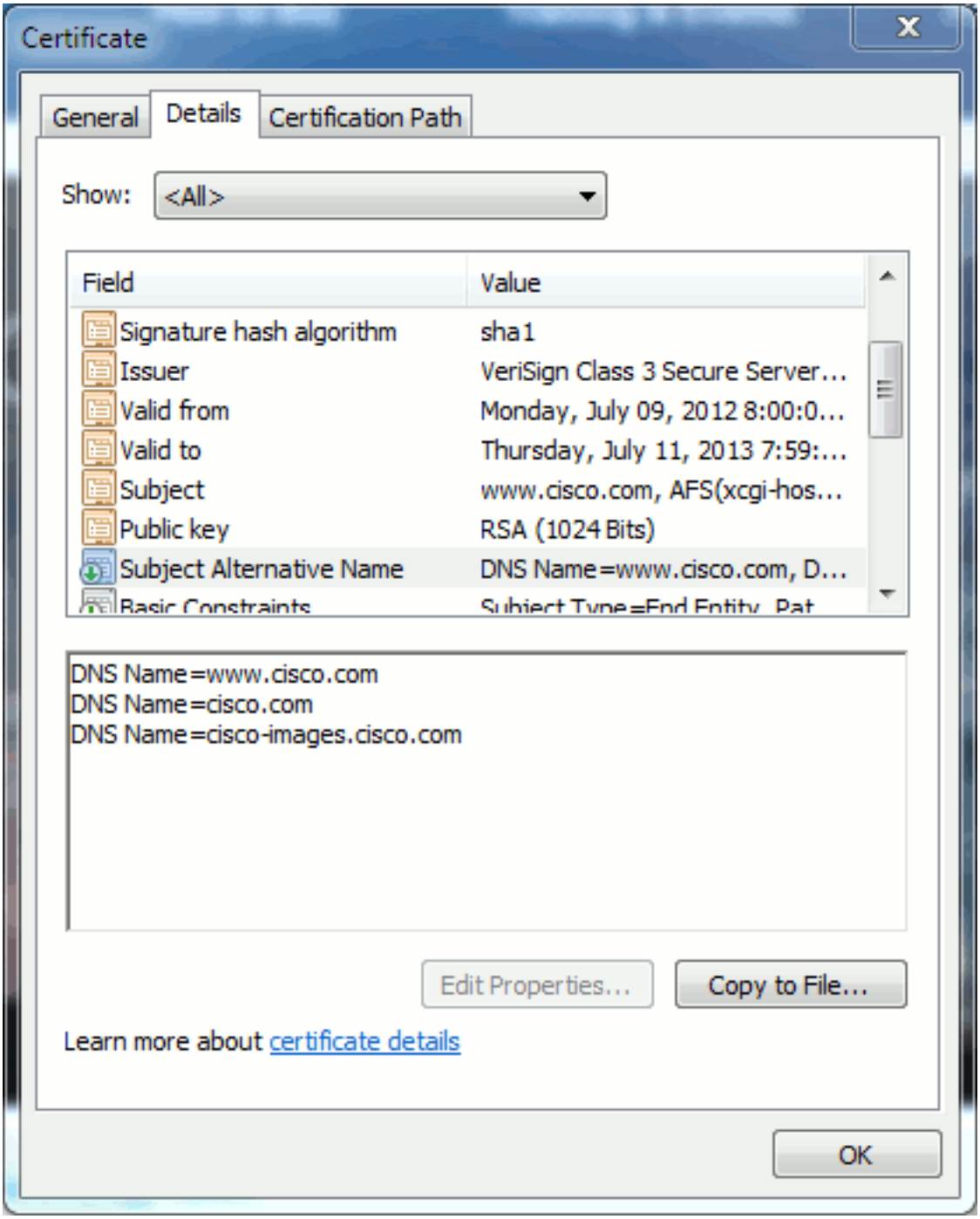
In the example shown in Figure 7, the certificate has the CN as www.cisco.com. The URL request for www.cisco.com from the browser checks the URL FQDN against the information the certificate presents. In this case, they match, and it shows the SSL handshake is successful. This website has been verified to be the correct website and communications are now encrypted between the desktop and the website.

Figure 7: Website Verification



In the same certificate, there is a SAN header for three FQDN/DNS addresses:

Figure 8: SAN Header



This certificate can authenticate/verify www.cisco.com (also defined in the CN), cisco.com, and cisco-images.cisco.com. This means you can also type cisco.com, and this same certificate can be used to authenticate and encrypt this website.

CUCM can create SAN headers. Refer to Jason Burn's document, CUCM Uploading CCMAdmin Web GUI Certificates on the Support Community for more information on SAN headers.

Wild Card Certificates

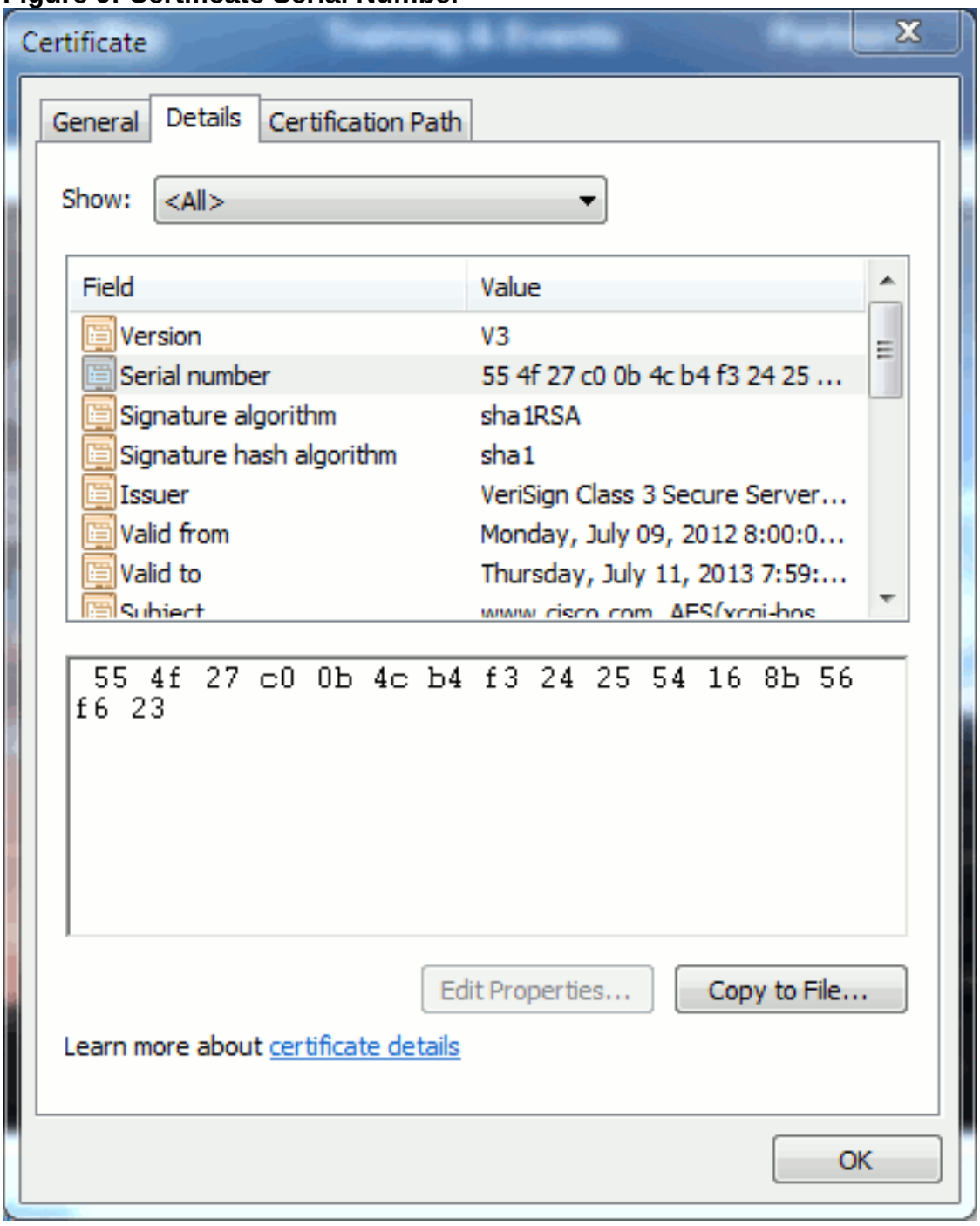
Wildcard certificates are certificates that use an asterisk (*) to represent any string in a section of a URL. For example, in order to have a certificate for www.cisco.com, ftp.cisco.com, ssh.cisco.com, and so on, an administrator would only need to create a certificate for *.cisco.com. In order to save money, the administrator only needs to buy a single certificate and does not need to purchase multiple certificates.

This feature is not currently supported by Cisco Unified Communications Manager (CUCM). However, you can keep track of this enhancement: CSCta14114: Request for support of wildcard certificate in CUCM and private key import.

Identify the Certificates

When certificates have the same information in them, you can see if it is the same certificate. All certificates have a unique serial number. You can use this to compare if the certificates are the same certificates, regenerated, or counterfeit. Figure 9 provides an example:

Figure 9: Certificate Serial Number



CSRs and Their Purpose

CSR stands for Certificate Signing Request. If you want to create a third-party certificate for a CUCM server, you need a CSR to present to the CA. This CSR looks a lot like a PEM (ASCII)

certificate.

Note: This is not a certificate and cannot be used as one.

CUCM creates CSRs automatically via web GUI: **Cisco Unified Operating System Administration** > **Security** > **Certificate Management** > **Generate CSR** > choose the service you want to create the certificate > then **Generate CSR**. Every time this option is used, a new private key and CSR is generated.

Note: A private key is a file that is unique to this server and service. This should never be given to anyone! If you provide a private key to someone, it compromises the security that the certificate provides. Also, do not regenerate a new CSR for the same service if you use the old CSR to create a certificate. CUCM deletes the old CSR and private key and replaces both of them, which makes the old CSR useless.

Refer to Jason Burn's documentation on the Support Community: CUCM Uploading CCMAdmin Web GUI Certificates for information on how to create CSRs.

Use of Certificates Between End Point and SSL/TLS Handshake Process

The handshake protocol is a series of sequenced messages that negotiate the security parameters of a data transfer session. Refer to the SSL/TLS in Detail, which documents the message sequence in the handshake protocol. These can be seen in a packet capture (PCAP). Details include the initial, subsequent, and final messages sent and received between the client and server.

How CUCM Uses Certificates

The Difference Between tomcat and tomcat-trust

When certificates are uploaded to CUCM, there are two options for each service via **Cisco Unified Operating System Administration** > **Security** > **Certificate Management** > **Find**.

The five services that allow you to **manage** certificates in CUCM are:

- tomcat
- ipsec
- callmanager
- capf
- tvs (in CUCM Release 8.0 and later)

Here are the services that allow you to **upload** certificates to CUCM:

- tomcat
- tomcat-trust
- ipsec
- ipsec-trust
- callmanager
- callmanager-trust

- capf
- capf-trust

These are the services available in CUCM Release 8.0 and later:

- tvs
- tvs-trust
- phone-trust
- phone-vpn-trust
- phone-sast-trust
- phone-ctl-trust

Refer to the CUCM Security Guides by Release for more details on these types of certificates. This section only explains the difference between a service certificate and a trust certificate.

For example, with **tomcat**, the **tomcat-trusts** upload the CA and intermediate certificates so that this CUCM node knows it can trust any certificate signed by the CA and intermediate server. The tomcat certificate is the certificate that is presented by the tomcat service on this server, if an end point makes an HTTP request to this server. In order to allow presentation of third-party certificates by tomcat, the CUCM node needs to know it can trust the CA and intermediate server. Therefore, it is a requirement to upload the CA and intermediate certificates before the tomcat (service) certificate is uploaded.

Refer to Jason Burn's CUCM Uploading CCMAdmin Web GUI Certificates on the Support Community for information that will help you understand how to upload certificates to CUCM.

Each service has its own service certificate and trust certificates. They do not work off each other. In other words, a CA and intermediate certificate uploaded as a tomcat-trust service cannot be used by the callmanager service.

Note: Certificates in CUCM are a per node basis. Therefore, if you need certificates uploaded to the publisher, and you need the subscribers to have the same certificates, you need to upload them to each individual server and node prior to CUCM Release 8.5. In CUCM Release 8.5 and later, there is a service that replicates uploaded certificates to the rest of the nodes in the cluster.

Note: Each node has a different CN. Therefore, a CSR must be created by each node in order for the service to present their own certificates.

If you have additional specific questions on any of the CUCM security features, refer to the security documentation.

Conclusion

This document assists and builds a high level of knowledge on certificates. This subject can matter can become more in-depth, but this document familiarizes you enough to work with certificates. If you have questions on any CUCM security features, refer to the CUCM Security Guides by Release for more information.

Related Information

- **Cisco Unified Communications Manager (CallManager) Maintenance and Security**

Guides

- Cisco Unified Communications Manager (CallManager)
- Cisco Unified Communications Manager Express
- Cisco Support Community: CUCM Uploading CCMAdmin Web GUI Certificates
- Bug CSCta14114: Request for support of wildcard certificate in CUCM and private key import
- Cisco Emergency Responder (CER) Explained
- Technical Support & Documentation - Cisco Systems