CERTIFICATE AUTHORITY

WHAT IS A CERTIFICATE AUTHORITY (CA)?

▪ A **certificate authority (CA),** also sometimes referred to as a **certification authority**, is a company or organization that acts to validate the identities of entities (such as websites, email addresses, companies, or individual persons) and bind them to cryptographic keys through the issuance of electronic documents known as **digital certificates.**

▪ **SSL/TLS certificate** – is a cryptographic file installed in your web browser.

A digital certificate serve two major purposes:

▪ **Authentication and trust**, by serving as a credential to validate the identity of the entity that it is issued to and usualy called the "chain of trust")

▪ **Encryption,** for secure communication over insecure networks such as the Internet. **Integrity** of documents **signed** with the certificate so that they cannot be altered by a third party in transit.

▪ A **Certificate Authority (CA)** is a trusted central administrative entity that can issue digital certificates to users and servers.

▪ The **trust in the CA** is the foundation of trust in the certificate as a valid credential. A **CA** uses its private key to create a digital signature on the certificate that it issues to validate the certificate's origin. Others can use the **CA certificate's** public key to verify the authenticity of the certificates that the CA issues and signs.

▪ A **CA** can be either a public commercial entity, such as **VeriSign**, or it can be a private entity that an organization operates for internal purposes. Several businesses provide commercial **Certificate Authority** services for Internet users. **Digital Certificate Manager (DCM)** allows you to manage certificates from both public CAs and private CAs.

WORKING WITH CERTIFICATES

▪ In an **AD domain network**, the most useful CA servers are of the Enterprise variety. **Enterprise CA servers** integrate with AD, making them visible to machines in the network and automatically trusted by computers that you join to your domain.

▪ There are differing opinions on the matter of best practices when setting up a series of **CA servers.** For example, there is a good test lab guide published by Microsoft, which walks you through setting up a stand-alone **Root CA, a Subordinate Enterprise CA**, and then taking the stand-alone root offline.

▪ An advantage of this is that certificates are issued from the subordinate, not directly from the root, and so if certificate keys are somehow compromised in the environment, the **Root CA** is completely offline and unavailable so that it cannot be compromised.

TRUSTED ROOT STATUS

▪ The term trusted root refers to a special designation that is given to a **Certificate Authority certificate**. This trusted root designation allows a browser or other application to authenticate and accept certificates that the Certificate Authority (CA) issues.

▪ When you download a **Certificate Authority's certificate** into your browser, the browser allows you to designate it as a trusted root. Other applications that support using certificates must also be configured **to trust a CA** before the application can authenticate and trust certificates that a specific CA issues.

You can view the SSL/TLS certificate details of any website that runs on HTTPS, and you'll find the following information:

Public key – details on which public key and the corresponding algorithm are used the certificate

Subject – Information about the domain the certificate was issued to or, for certain types of certificates, the legitimacy of the organization operating the website.

Issuing CA – The signature of the trusted issuing authority that reaffirms the security of any information shared with the website.

Validity period – The time period until which the specific certificate is valid.

How do SSL/TLS certificates work?

Steps when a user connects to a website over HTTPS:

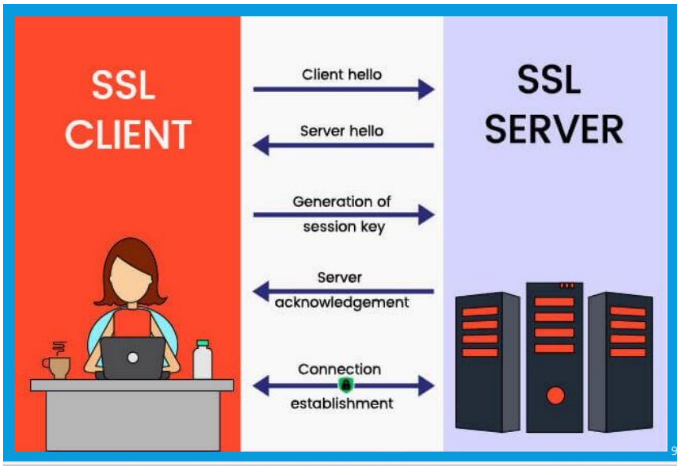
❑ The user machine's browser attempts a connection to the website's server machine secured with TLS and requests it to prove its identity.

❑ The web server receives the request and sends back a copy of its TLS certificate along with its public key.

❑ The browser receives the certificate and checks its legitimacy by comparing it with a predefined list of trusted CAs.

❑ The web server decrypts the message using its private key and sends an acknowledgement—which is encrypted using the session key—back to the browser to start the session.

❑ The TLS handshake process is now complete



THE CHAIN OF TRUST

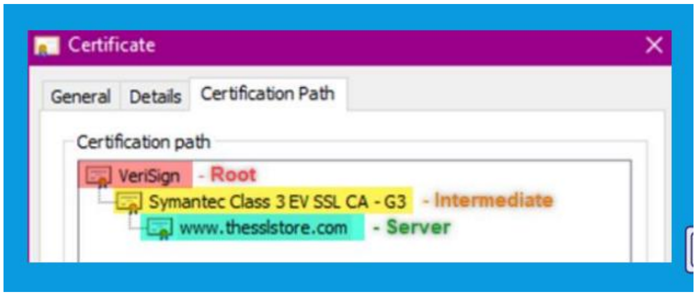
❑ Certificate chain of trust is an important concept in public key infrastructure (PKI) that helps trace an SSL/TLS certificate back to its root certificate, i.e., the issuing CA with which it was signed.

The 3 Significant components of the chain of trust:

1. **Root certificate** - is a digital certificate that belongs to the issuing Certificate Authority.

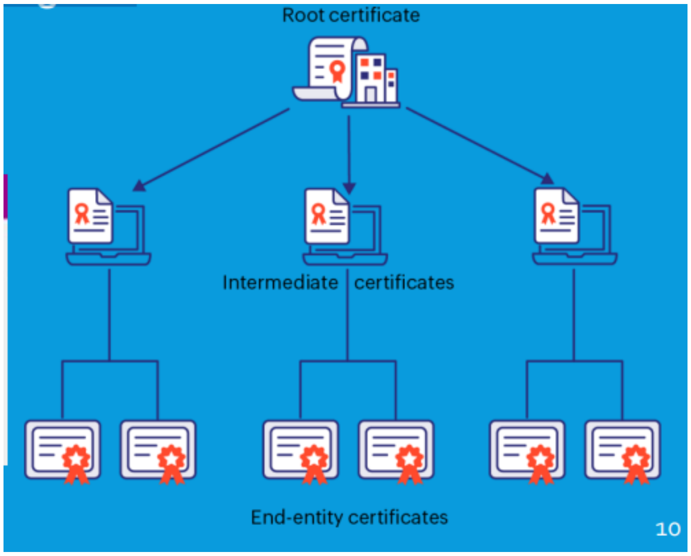
2. **Intermediate certificate** - they act as middle-men between the protected root certificates and the server certificates issued out to the public.

3. **Server or end-entity certificate** - the server certificate is the one issued to the specific domain the user is needing coverage for



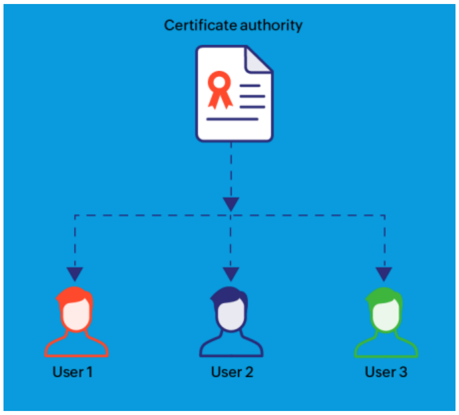
Certification path

VeriSign - Root
Symantec Class 3 EV SSL CA - G3 - Intermediate
www.thesslstore.com - Server

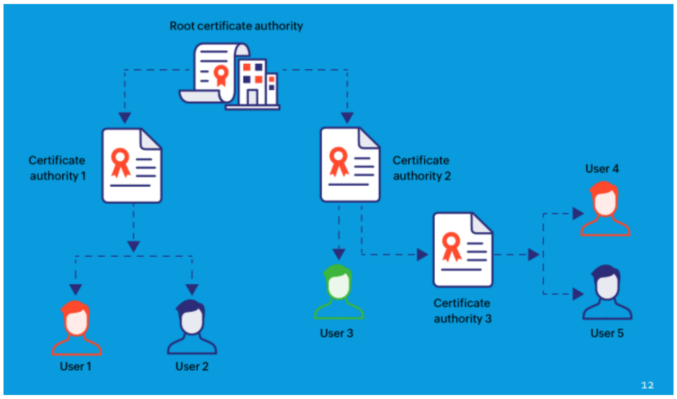


TYPES OF TRUST MODELS

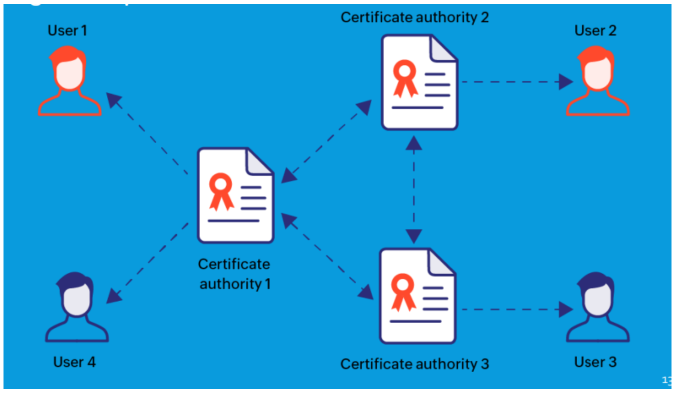
▪ **Single CA model** - In this model, all user certificates are directly issued by the CA present in the chain. The chain of trust begins with the CA and ends with the server certificate with no intermediate certificate involved.



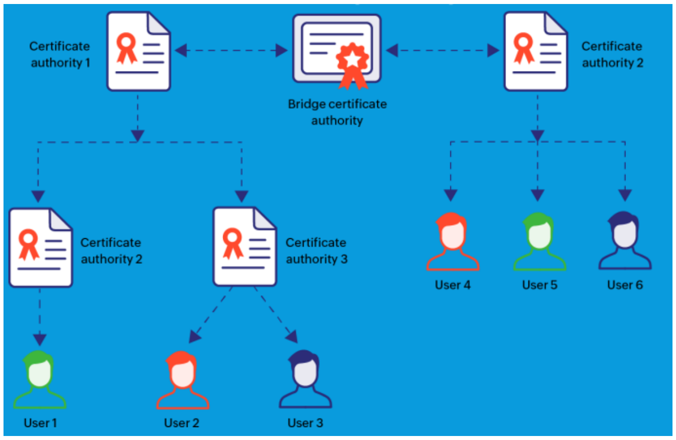
- **Hierarchical model** - also known as the tree model, is the most commonly adopted method for implementing PKI. In this configuration, the root CA, or the trust anchor, occupies the top most position of the trust chain and the trust relationships are unidirectional (from top to bottom)



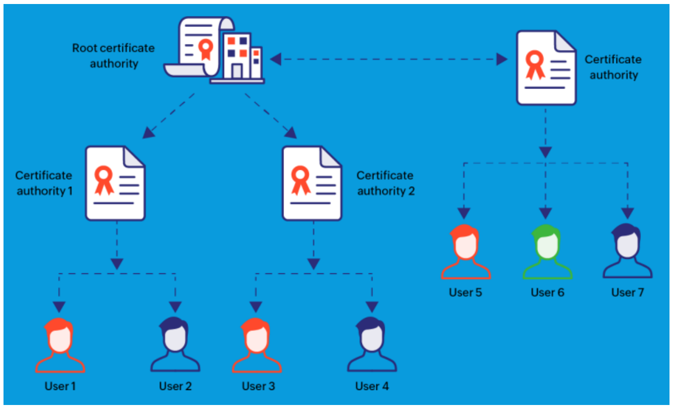
- **Mesh model** - the mesh trust model is also called cross-certificate architecture, where the trust anchor of an end server is its local CA. All the CAs function independently and the trust relationship is mutual as opposed to the hierarchical model that operates with a single trust point.



- **Bridge model** - this model is similar to the mesh model with bidirectional trust relationships but with a bridge CA acting as the hub for all the autonomous CAs involved in the architecture. In this model, the chain of trust is less complex than the mesh framework since it's traceable back to a single bridge CA



- **Hybrid model** - the highly dynamic nature of IT networks calls for organizations to adopt trust models that are agile and scalable



CERTIFICATE AUTHORITY POLICY DATA

When you create a local **Certificate Authority (CA)** with Digital Certificate Manager, you can specify the policy data for the local CA. The **policy data for a local CA** describes the signing privileges that it has. The policy data determines:

➢ Whether the local CA can issue and sign user certificates.

➢How long certificates that the local CA issues are valid

Self-signed SSL/TLS certificates and their risks

Digital certificates needn't necessarily be obtained from public CAs. Users can generate self-signed certificates on their own local hosts with their own private keys instead of requesting them from a public or private CA. RISK:

❑ **Dangerous public browsing behavior**: Self-signed certificates still produce security warnings.

❑ **Spoofing attacks**: Compromised self-signed certificates can cause attackers to spoof the identity of the victim and expand their access across the organization.

❑ **Inability to revoke**: Unlike CA certificates, self-signed certificates can't be revoked.

Steps to acquire CA-signed SSL/TLS certificates

CA-signed certificates, or public certificates, are digital certificates signed and issued by trusted CAs. It's highly recommended that enterprises use CA-signed certificates for public facing sites since they will be automatically verified and trusted by client operating systems.

STEPS:

❑ **Choosing the CA:** Selecting the right type of CA partner that meets corporate security policies is the first step towards acquiring and managing public CA certificates. The list of things to take into consideration when evaluating CAs include:

❑ Security reputation

❑ Pricing

❑ Customer service

❑ Technical agility

❑ **Choosing the certificate type**: After choosing the CA, the next step is to assess and find which SSL/TLS certificate type meets your requirements. The three major types:

❑ **Domain validated (DV SSL) certificates** - the least stringent validation.

❑ **Organization validated (OV SSL) certificates** - the CA verifies the requesting entity's legal ownership of the domain and also vets the organizational information to some extent, giving some enhanced visibility on the party processing your information.

❑ **Extended validation (EV SSL) certificates** - the longest and costliest, and certificate issuance is done by adhering to the strict protocol of the EV SSL Certificate Guidelines as formally ratified by the CA/B Forum

❑ **CSR generation and certificate request**: is to generate a CSR and send it to the CA. The CSR is an encoded file containing the public key and some details about the certificate and the organization, and it's the accepted standard of requesting a certificate from a public CA.

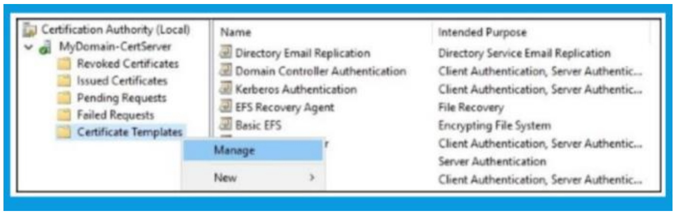
❑ **Validation and provisioning** - once the CA receives a CSR, it uses the enclosed public key to decrypt the signature and gather the specified information. The CA then verifies your identity, right to the domain specified, and runs a background check on your organization (depending on the type of certificate requested).

❑ **Monitoring and renewal** - Once the certificates are deployed to required servers within the network, they need to be constantly monitored for their usage

CREATING A CERTIFICATE TEMPLATE TO PREPARE FOR ISSUING MACHINE CERTIFICATES TO YOUR CLIENTS

▪ There are some built-in certificate templates that preinstall when you add the CA role to your server. Some companies utilize these built-in templates for issuing certificates, but it is a better practice to create **your own templates**.

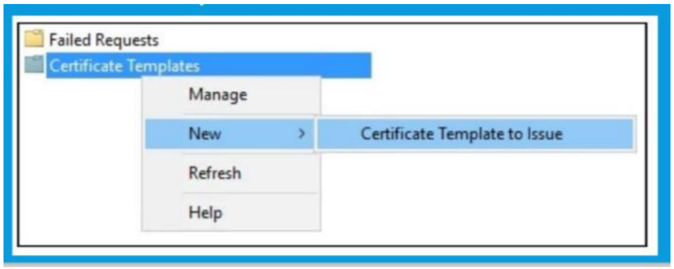
▪ There are a few criteria we need to meet in these certificates, and the built-in Computer template comes close to checking all the options that we need. So we will take that template, copy it, and modify it to meet our requirements.

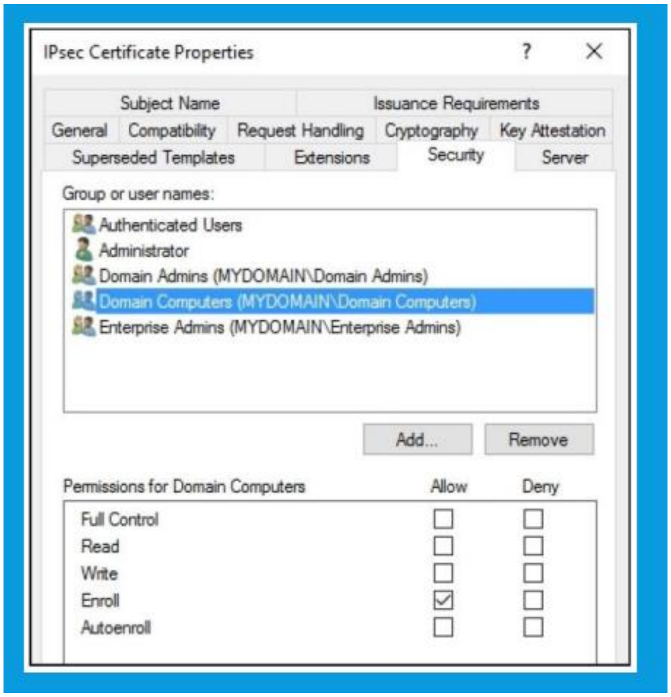


Name	Intended Purpose
Directory Email Replication	Directory Service Email Replication
Domain Controller Authentication	Client Authentication, Server Authentic...
Kerberos Authentication	Client Authentication, Server Authentic...
EFS Recovery Agent	File Recovery
Basic EFS	Encrypting File System
	Client Authentication, Server Authentic...
	Server Authentication
	Client Authentication, Server Authentic...

PUBLISHING A CERTIFICATE TEMPLATE TO ALLOW ENROLLMENT

❑ Having created a new certificate template does not necessarily mean that you are ready to start issuing certificates based on that template. We also need to publish our new template so that the CA server knows that it is ready to publish out to computers and users. There is also a security section of the template properties, where you need to define who or what has access to request certificates based on that template.




IPsec Certificate Properties

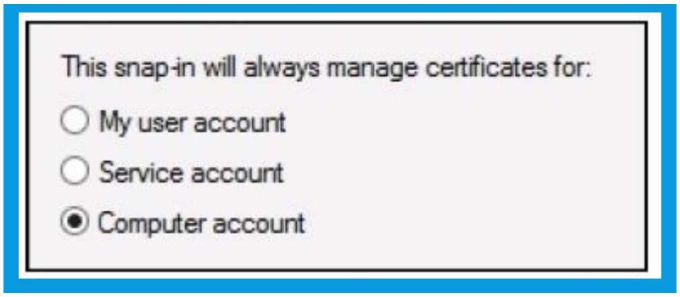
Group or user names:

Authenticated Users
Administrator
Domain Admins (MYDOMAIN\Domain Admins)
Domain Computers (MYDOMAIN\Domain Computers)
Enterprise Admins (MYDOMAIN\Enterprise Admins)

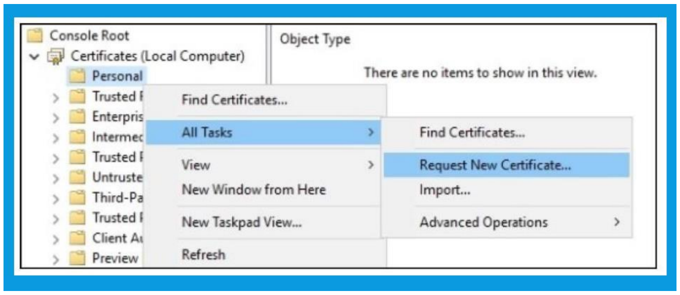
Permissions for Domain Computers	Allow	Deny
Full Control	□	□
Read	□	□
Write	□	□
Enroll	☑	□
Autoenroll	□	□

USING MMC TO REQUEST A NEW CERTIFICATE

▪ The most common way that I see administrators interface with the certificates on their systems is through the MMC snap-in tool. **MMC** is short for **Microsoft Management Console**, and by using MMC, you can administer just about anything in the operating system.


This snap-in will always manage certificates for:

○ My user account
○ Service account
● Computer account



There are no items to show in this view.

Prepared by: Kenth Davis Guardiario