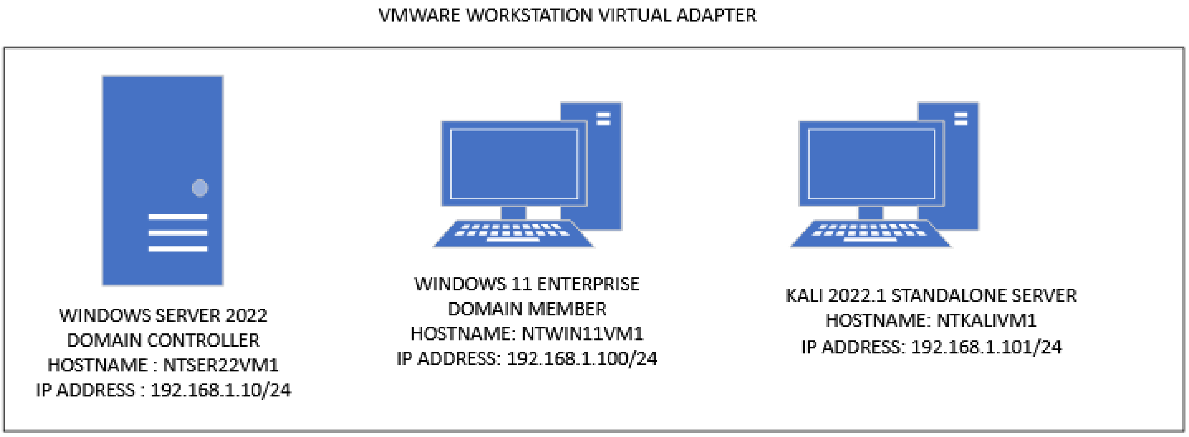
Exercise 1 - PKI Fundamentals

The Public Key Infrastructure (PKI) is a hierarchical framework of Certificate Authorities (CAs) that can issue and sign digital certificates. For networked elements such as web servers, users, and a variety of other resources, these certificates are utilized for validation, data protection, and encryption. Certificates can be used for a variety of things, including disk encryption and email encryption. You can generate an appropriate form of certificate for use based on your needs.

In this exercise, PKI Concepts Types of Certificates Certificate Formats

Topology


VMWARE WORKSTATION VIRTUAL ADAPTER
WINDOWS SERVER 2022
DOMAIN CONTROLLER
HOSTNAME : NTSER22VM1
IP ADDRESS : 192.168.1.10/24
WINDOWS 11 ENTERPRISE
DOMAIN MEMBER
HOSTNAME: NTWIN11VM1
IP ADDRESS: 192.168.1.100/24
KALI 2022.1 STANDALONE SERVER
HOSTNAME: NTKALIVM1
IP ADDRESS: 192.168.1.101/24

DOMAIN = networktute.com

NTSER22VM1 = Windows Server 2022 – Domain Controller

NTWIN11VM1 = Windows 11 – Domain Member

NTKALIVM1 = Kali 2022.1 - Standalone Server

Prerequisite

- *VMware Workstation 16 Pro*
  - When making this tutorial, we used the “Windows Server 2019” VM Template and “Windows 10 & later” VM Template. Since VMware didn’t have the updated templates.
- *Microsoft Windows Server 2022*
- *Microsoft Windows 11*
- *Kali Linux 2022.1*

PKI Concepts

Before you begin working with PKI and certificates, you should be familiar with a few terms that will aid in your comprehension of the PKI capabilities. Let's go through some of the major terms now.

Key Management

Encryption is a typical method for encrypting data. Users, however, fail to protect the keys used to encrypt the decrypted data while attempting to protect the data. The important component, the private key, must be appropriately managed, safeguarded, and protected from coming into the hands of a threat actor. A person or a company must follow correct key management procedures, which include key storage, key handling, and key usage.

Certificate Authority (CA)

A certificate authority (CA) is in charge of the full certificate management lifecycle and has the authority to issue and revoke certificates. A CA can issue the right sort of certificate to users and devices depending on the type of certificate requested.

A Root CA is at the top of the hierarchy, followed by one or more levels of intermediate CAs. An intermediate CA is trusted by the authority granted to it by a Root CA in a simple model. In a real PKI architecture, however, there could be numerous levels of intermediary CAs.

A web server operator, for example, might send a CSR (signing request) to a "intermediate" CA in order to obtain a certificate for SSL or TLS encryption. A certificate for users and devices is not generated by a Root CA. It is in charge of generating certificates for There may be one or more intermediary CAs.

In some scenarios, you can even create a self-signed certificate, although outside elements will not trust it, and validation errors may be raised. These are mainly used for testing.

Intermediate CA

A major company can set up its own Root CA and delegate signing authority to intermediate CA servers. The domain name of a website is examined by a web browser. It then starts the process of chaining back from one signed certificate to the next until it reaches a root certificate signed by one of its root store's certificates. The site's certificate is accepted if it discovers a root certificate in this way.

Registration Authority (RA)

When a client seeks a certificate, the request is validated by the Registration Authority (RA). If the validation is successful, the RA informs the CA that a certificate based on the client's request can be granted. It's worth noting that the RA never gives the client a certificate. Its sole responsibility is to validate the request and submit it to the CA.

Certificate Revocation List (CRL)

A CRL is a list that contains the names of two different types of certificates:

- Invalid certificates, which can include the expired certificates
- Revoked certificates, which have been revoked by a CA

The reason for certificate revocation, as well as the dates when the revocation occurred, are listed in a CRL. The CA issues a certificate that keeps track of the CRL. Digital signatures are employed to prevent tampering with the CRL.

Certificate Attributes

A certificate has a few areas with data. These areas are known as the attributes that characterize the certificate. For illustration, it features a Subject field, which gives the DName of the client who possesses the certificate. The DName field relates to the Catalog Title of the question. An property can be made of attribute-value sets, known as Relative Distinguished Names (RDNs). For illustration:

- CN: CommonName
- OU: OrganizationalUnit
- O: Organization
- S: State
- C: Country
- L: Locality

For example, a certificate can have the following DName:

CN=NT Cert, OU=IT, O=Network Tute, L=London 4, S=London, C=UK

Online Certificate Status Protocol (OCSP)

The OSCP convention gives the online denial administrations, which help a client approves a serial number and give the certificate's status to the client. When a client sends a request to check the certificate's status, OSCP's responder benefit checks the status and shares it with the client. It might report the certificate status as substantial, denied, or unknown.

Certificate Signing Request (CSR)

When a client demands a certificate from a CA, the ask goes within the shape of a CSR, which contains the RSA-based open key. The CSR also contains the required information that's required for the certificate. To begin with, you wish to create private and public keys. When a CSR is sent, the open key is sent in it. The CA takes the open key and inserts it into the certificate that's issued to the user.

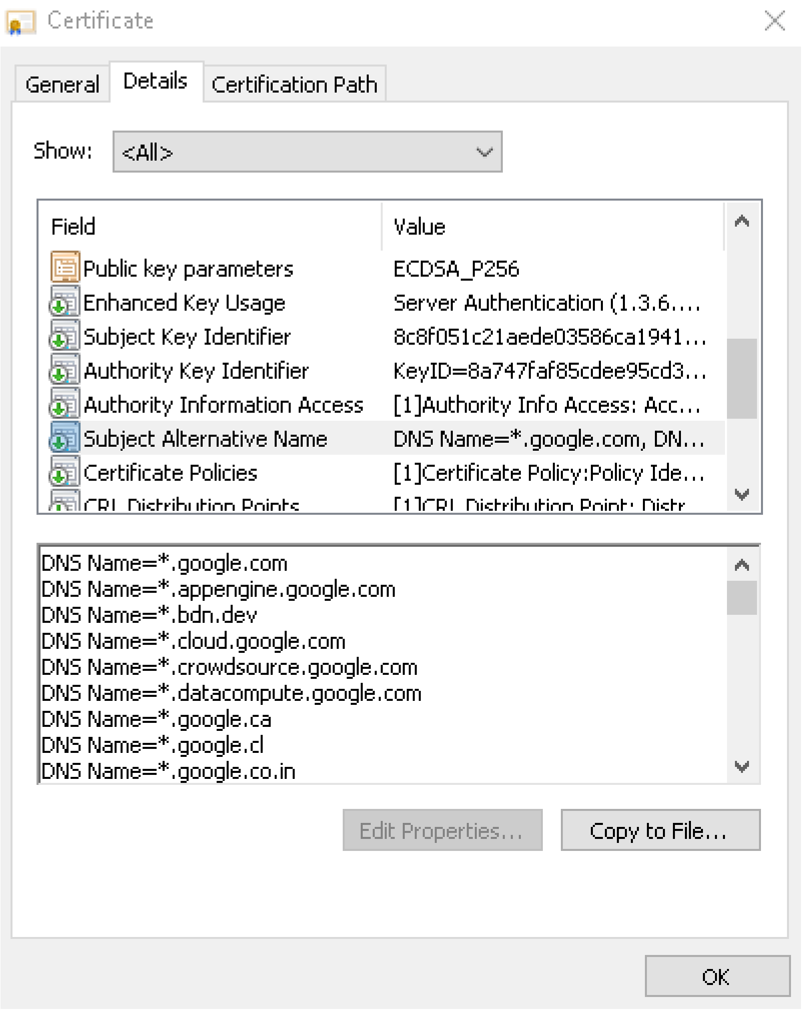
Common Name (CN)

A CN speaks to the Completely Qualified Space Title (FQDN). Once you create a CSR, you need to supply the CN, which ought to be within the shape of www.networktute.com or networktute.com.

Subject Alternative Name (SAN)

A SAN may be a multi-purpose field in a certificate. For diverse substances, it can serve different purposes. For case, when utilizing an SSL certificate, it can speak to different domain names to which the certificate applies. Within the given show, take note that SAN is applicable for the SSL certificate utilized for *.google.com.

For entities, such as a user, a SAN can represent the User Principal Name (UPN). For a computer or machine, it represents the FQDN.

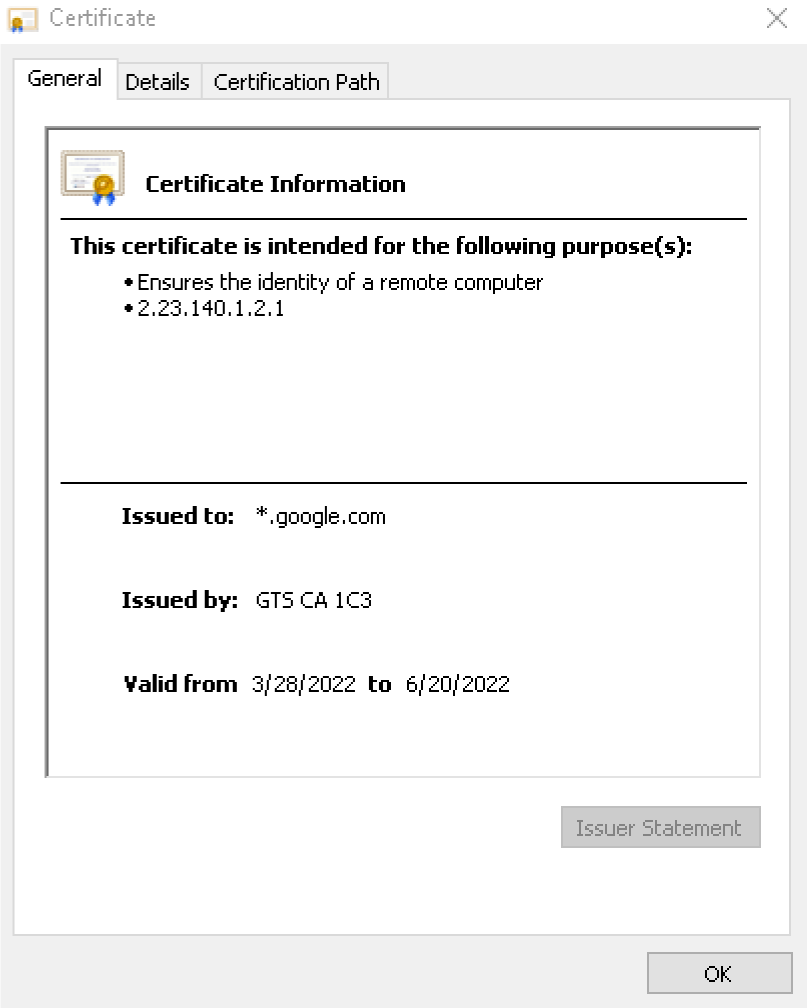


Field	Value
Public key parameters	ECDSA_P256
Enhanced Key Usage	Server Authentication (1.3.6....
Subject Key Identifier	8c8f051c21aede03586ca1941...
Authority Key Identifier	KeyID=8a747faf85cdee95cd3...
Authority Information Access	[1]Authority Info Access: Acc...
Subject Alternative Name	DNS Name=*.google.com, DN...
Certificate Policies	[1]Certificate Policy:Policy Ide...
CRL Distribution Points	[1]CRL Distribution Point: Distr

DNS Name=*.google.com
DNS Name=*.appengine.google.com
DNS Name=*.bdn.dev
DNS Name=*.cloud.google.com
DNS Name=*.crowdsource.google.com
DNS Name=*.datacompute.google.com
DNS Name=*.google.ca
DNS Name=*.google.cl
DNS Name=*.google.co.in

Expiration

Each certificate has an expiration date. After you have a certificate and a parcel of other information, you'd too see the creation and the expiration date. When a certificate expires, it cannot secure the exchanges taking put on the Site to which the certificate was connected. Once you stack the Site with a lapsed certificate, the Web browser is likely to show a certificate expiration warning.



Certificate Information

This certificate is intended for the following purpose(s):

- Ensures the identity of a remote computer
- 2.23.140.1.2.1

Issued to: *.google.com

Issued by: GTS CA 1C3

Valid from 3/28/2022 **to** 6/20/2022

Online vs. Offline CA

A CA can be kept as online or offline depending on its part and how as often as possible it needs to sign a certificate. A root CA ought to continuously be kept offline. You'd not require a root CA to sign certificates frequently. The term offline implies that the CA isn't effectively generating certificates and is part of the arrange either within the shutdown state or fair not associated.

A root CA would sign the middle CA certificates. Other than this, they have little role to play and must be secured, which can be done by keeping them offline. An online intermediate CA can handle CRLs, certificate approval demands, and overseeing and distributing the certificates.

Stapling

When a user makes a certificate request, the information about the Responder is also transmitted. Consider a situation in which a CA must respond to thousands of requests at the same time. It has the potential to put more strain on the CA. As a result, to lighten the weight on the certificate request is submitted along with the Responder information to the server. As a result, it is a Rather than two distinct requests, make a single request. Stapling is the term for this procedure.

Pinning

Pinning is the process of hard-coding a digital certificate into an application. Consider the case of a mobile phone that connects to several networks as it goes about the city. A digital certificate is used to guarantee that the apps are not hacked. When a server has to be verified, its public key is compared to the hashed public key of the certificate.

Trust Model

Several components, such as protocols, services, and devices, work together in a trust model to ensure a given level of security. When it comes to a single domain, trust is simple to develop. If the trust needs to be established across two or more domains, however, a trust anchor is required.

If you and another user need to communicate via encrypted Emails, for example, you may not be able to trust each other. When you view another user's certificate, which was generated by the same CA as yours, you will establish trust immediately. This is the case when the certificate is issued for a single domain by the same CA.

However, there are chances that your CA may have to trust another CA. The trust between two CA is developed using a trust model, which can be of the following types:

- **Hierarchical:** Usually the foremost widely used trust model. There's a root CA, at that point the intermediate, and after that you'll moreover have the leaf CA, which is at the foot of the hierarchy. The whole progression offers the root CA certificate and the open keys.
- **Peer-to-peer**: In this show, there are CA, but they are not subordinate to each other. Each CA issues certificate to its substances. In the event that a cross-certification between two CA is required, at that point they got to certify the open key for each other. When they certify the public key for each other, a bi-directional believe is made.
- **Hybrid**: An organization may ought to communicate with accomplices, merchants, or even client organizations. Each of them can have their own CA, which is utilizing the hierarchical trust show. However, to set up a believe between these CA, you'll actualize the peer-to-peer believe demonstrate, where each CA certifies the open key for each other.

Key escrow

Private key maintenance may be a challenging undertaking for a user. The key escrow technique, which employs a third-party to handle the user's private key, can relieve the user of managing and safeguarding the private key. The user's private key is divided into two halves and encrypted throughout this operation. After then, each element of the private key is kept in its own place. When the user wants to utilize the private key, the user retrieves both portions and combines them to make a copy of the private key.

In the key escrow process, a copy of the private key can always be retrieved. This process is also helpful when you have lost the private key and need to retrieve it.

Certificate Chaining

The certificates in a certificate chain trust each other. The root certificate, which is issued by the root CA, is the first certificate issued. The intermediate certificate, which signs the certificate to the user or any device, is then signed with the root certificate. The intermediate certificate issued by the user or device certificate is trusted by the user or device certificate. CA in the middle. The root certificate is trusted by the intermediate certificate. This results in the certificates are linked in a chain of trust.

Types of Certificates

Certificates can be broadly categorized into various categories. Some of the key ones are as follows:

Wildcard

In most cases, a certificate is only issued to one entity. Assume a certificate for networktute.com is issued. The certificate is only valid for networktute.com in this case. In a scenario where there are multiple entities, such as www.networktute.com, web.networktute.com, cdn.networktute.com, and so on. For each of these entities, you'll need a certificate.

You have two options in this situation. The first option is to obtain a separate certificate for each of these entities. Obtaining a wildcard certificate is another alternative that is better and more manageable. The certificate would be granted to *.networktute.com and would be valid for entities like www.networktute.com, web.networktute.com, and cdn.networktute.com, among others.

Subject Alternative Name (SAN)

A Subject Alternative Title (SAN) computerized certificate is more commonly known as a UCC certificate. UCC stands for Bound together Communications Certificate. A UCC certificate is used with bound together communication applications, such as Microsoft Trade Server.

Code Signing

When designers compose the code for an application, they have to be ensured the code. To be able to do this, the designers can utilize computerized certificates and sign the code. Once the code is marked, it avoids the code from being adjusted in an unauthorized way.

Self-Signed

Consider the case where you have an internal Webserver that you utilize for testing. You want to include a certificate in this Webserver but don't want to spend any money on it. You can construct a self-signed certificate to meet these requirements. As the certificate's creator, you must certify it before the Webserver can utilize it. A self-signed certificate is a one-of-a-kind certificate that does not require the use of a certificate authority. Self-signed certificates are used by a variety of network devices and applications.

Machine/Computer

You can create a machine or computer certificate that can be used to validate its identification using a certificate authority. A network device can use a certificate to verify that it is a genuine device and offer the services for which it was commissioned. To authenticate it via the network, a machine or computer certificate is used.

Email

You can also build an Email digital certificate that a user can use to sign and encrypt incoming emails. The incoming encrypted messages can be decrypted using the same certificate.

User

A user certificate is a document that identifies a person. When a user uses the Encrypting File System (EFS) to encrypt a hard disk and then wants to decrypt it, the certificate is used to authenticate the user.

Root

A root certificate is one that is issued by the root certificate authority. It's the first certificate in the chain of trust. When a root CA issues a certificate, it signs it with its own signature, and that certificate becomes the root certificate.

Domain Validation

Domain validation (DV) certificates verify a company's authorization to use a given domain name. This form of certificate verifies the legitimacy of the company that owns the domain name. A padlock appears before the domain name in the domain validation certificate. The certificate information can be viewed by clicking on the padlock.

Extended Validation

The upgraded version of the domain validation certificate is an Extended Validation (EV) certificate. The CA must verify the legitimacy of the owner, the organization, the physical address, and the operations of the company. The owner of the website must also establish his or her identity and domain name ownership. EV certificates, like domain validation certificates, display a padlock and the company name in the address bar before the Website name.

Certificate Formats

When creating a certificate, you can encode them in different formats. Each encoded format can have a different extension. Here are the certificate formats:

Distinguished encoding rules (DER): A certificate file with the CRT extension is encoded as a DER binary file or as an ASCII file. If a certificate's extension is.der, it means it's in binary format. A KEY file can alternatively be encoded as a DER binary or ASCII file. The binary data is a crucial piece of information in the DER format. It's usually just used as a single certificate.

Privacy enhanced mail (PEM): This format protects the security and integrity of email communications by encrypting them. A CRT file encoded in PEM is equivalent to a CER file encoded in PEM.

Personal information exchange (PFX): It's a file type that's used to verify the authenticity of websites and programs. Both the private and public keys are included in this form of certificate. It is also secured by a password.

.cer: A CER extension indicates a Microsoft version of a CRT certificate file.

P12: The P12 format is based on the RSA Corporation's standards. It also includes both public and private keys.

P7B: The file extension for this format is.p2b or.p7c. Only the public key is included in this format; the private key is not. It may also include certificates, a certificate chain, and a certificate revocation list (CRL).