PKI

By – Manu sir 5-10-19

Public Key Infrastructure:

- ISO authentication framework that uses public key cryptography and the X.509 standard.
- Enable authentication to happen across different networks and the Internet.
- PKI is a hybrid system of symmetric and asymmetric key algorithms and methods.

Entities and Function:

- Certification Authorities (CA).
- Registration Authorities (RA).
- Certificates.
- Certificates repository.
- Certificate revocation system.
- Keys.
- Key backup and recovery system.
- Timestamping.
- Client-side software.
- Users.

Types of CA:

- Internal.
- Public (External).

Cross certification:

- Process undertaken by CA's to establish a trust relationship.

Online Certification Status Protocol (OCSP):

- Carries out real-time validation of a certificate and back to the user whether the certificate is valid, invalid or unknown.

X.509:

- Standard that defines the CA creates the certificates.
- Currently at version 4- denotes the X.509v4.

Trusted timestamping;

- Process of surely keeping track of the creation and modification time of a document.
- No one (include the owner of the document) should be change it once it has been recorded.

Research Organization:

- Internet Research Task Force (IRTF)
- Crypto Forum Research Group (CFRG)
- The Internet Engineering Task Force (IETF)

PKI

By – Manu sir 5-10-19

PKCS:

- Public key Cryptography Standards
- Group of public-key cryptography standards devised and published by RSA security LLC.

FIPS 140-2:

- Federal Information Processing standard (FIPS) Publication 140-2, (FIPS PUB 140-2).
- Title-Security Requirements of Cryptographic Modules.

X.500:

- Series of computer networking standards covering electronic directory services.
- Protocols:
    - DAP (Directory Access Protocol)
    - DSP (Directory System Protocol)
    - DISP (Directory Information Shadowing Protocol)
    - DOP (Directory Operational Bindings Management Protocol)

S/MIME (Secure/ Multipurpose Internet Mail Extensions):

- Standard for public key encryption and signing of MIME data.

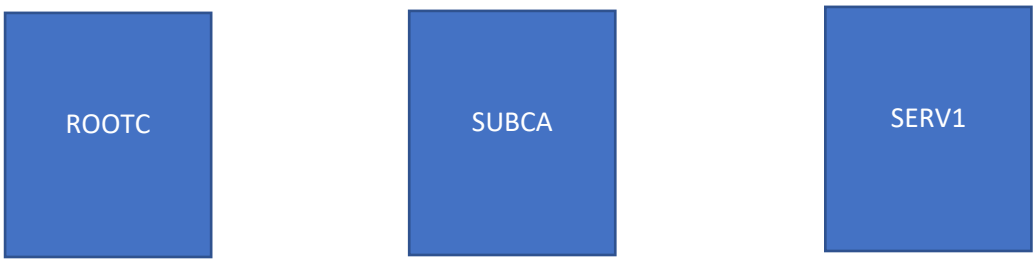
Without using password how we can authenticate the service using PKI:

1. Generate the Private and Public key in the particular der via Putty gen
2. # mkdir .ssh
3. # Chmod 700 .ssh/
4. # Cd .ssh/
5. # Vim authorized key
6. Copy public key to remote machine - .ssh/ authorized_keys2
7. # Echo publickey > authorized_key2
8. Connect to remote machine using Putty

PKI

By – Manu sir 5-10-19

Task :



.101	.102	.103
Dnsutils	Dnsutils	Dnsutils
Hostname -f -> rootca.shuhari.local		NS1
Hostname -s -> rootca		www
Hostname -d -> shuhari.local		Apache and Bind

1. rootca.shuhari.local
2. Configure hostname
3. Static ip
4. Apt
5. Install softwares
6. Configure DNS on all three machines.
7. Setup

Rootca:

1. Create a directory structure.
2. Make sure shuhari has sudo permission
3. # mkdir /home/shuhari/ca
4. # Cd /ca
5. # Mkdir -p certs crl newcerts private subca/csr subca/certs
6. # tree
7. Objective is to create a private/public key and a certificate.
8. We need to secure private folder.
9. # Chmod 700 private/
10. Create the CA Database (Files)
11. # touch index.txt
12. # touch index.txt.attr
13. # echo 1000 > serial ……(1000 it's an hexa number, cert will start with 1000)
14. # echo 1000 > crlnumber
15. # tree .
16. Download the root CA configuration File

PKI

By – Manu sir 5-10-19

17. # wget http:// 192.164.74.74/files/pki/rootca.cnf
18. Afte downloading the rootca.cnf, make sure to edit the file where in dir = /home/<username>/ca

19. Generate and secure rootca private key
20. # openssl genrsa -aes256 -out private/ca.key.pem 4096 ……..(4096 is size and **-aes256** is an symmetric encryption method which we are using to encrypt the RSA generated private key)
21. # tree .
22. # ls -l private/ …… (we will only give the read permission)
23. # Chmod 400 private/ca.key.pem
24. # ls -l /etc/ssl/openssl.cnf …… (path of openssl configuration file)
25. Generate and secure root CA Certificate
26. # openssl req -config rootca.cnf -key private/ca.key.pem -new -x509 -days 7300 -sha256 -extensions v3_ca -out certs/ca.cert.pem
    a. Enter your passphrase
    b. Country → IN
    c. State → MH
    d. Locality → Pune
    e. Organization Name → ShuHaRi Labs
    f. Organizational unit Name → IS
    g. Comman Name → rootca.shuhari.local
    h. Email Address → info@shuhari.local
27. # tree .
28. # chmod 444 certs/ca.cert.pem

Subca:

1. Introduction
    a. Configure Name Resolution (DNS client)
    b. Create the directory structure
    c. Secure the private folder
    d. Create the CA database (Files)
    e. Download the sub CA Configuration File
    f. Generate and secure sub CA Private Key
    g. Create the certificate signing request (CSR)
    h. Copy CSR file from sub CA to Root CA
    i. Generate / sign and secure sub CA certificate
    j. Verify the chain of trust between sub CA and root CA
    k. Copy the certificate chain file
    l. Copy the certificate chain file to sub CA (and secure the file)
2. # mkdir /home/shuhari/subca/
3. # cd subca/
4. # mkdir certs crl csr newcerts private
5. # Chmod 700 private/
6. # touch index.txt index.txt index.txt.attr
7. # tree .

PKI

By – Manu sir 5-10-19

8. # wget -q http://192.168.74.74/files/pki/subca.cnf
9. # cat subca.cnf | grep shuhari
10. # pwd
11. # openssl genrsa -aes256 -out private/subca.key.pem 4096
12. # chmod 400 private/subca.key.pem
13. # tree .
14. # openssl req -config subca.cnf -new -sha256 -key private/subca.key.pem -out csr/subca.csr.pem
    a. Name : IN
    b. Stare : MH
    c. Locality : Pune
    d. Organization Name : ShuHaRi Labs
    e. Organizational Unit : IS
    f. Common Name : subca.shuhari.local
    g. Email address : info@shuhari.local
15. Copy the subca.csr.pem to rootca's subca/csr.
16. # openssl ca -config rootca.cnf -extensions v3_intermediate_ca -days 3650 -notext -md sha256 -in subca/csr/subca.csr.pem -out subca/certs/subca.cert.pem ……(run in rootca)
17. Check both the files are same with md5.
18. # md5sum newcerts/1000.pem subca/certs/subca.cert.pem ……. (run in rootca)
19. # openssl x509 -noout -text -in subca/certs/subca.cert.pem
20. # openssl verify -CAfile certs/ca.cert.pem subca/certs/subca.cert.pem …(in rootca)
21. Copy the file subca.cert.pem from rootca to subca's certs
22. # cat subca/certs/subca.cert.pem certs/ca.cert.pem > subca/certs/ca-chain.cert.pem ……(in rootca)
23. # scp subca/certs/ca-chain.cert.pem shuhari@subca:/home/shuhari/subca/certs/ …..(in rootca)
24. # chmod 444 ca-chain.cert.pem ……(in subca)

Generate (TLS) certificate for the web server

- Create the directory structure
- Download / create the configuration file
- Generate and secure the private key
- Generate CSR
- Copy CSR to subca
- Generate and secure the certificate
- Copy web server certificate to the web server
- Copy CA certificate chain file to the web server
- Verify chain of trust between web server, rootca and subca.

Ser1:

PKI

By – Manu sir 5-10-19

1. # mkdir certs
2. # cd certs
3. Download the configuration file
4. # wget -q http://192.168.74.74/files/pki/subca.cnf
5. Generate the private key and secure
6. # openssl genrsa -out www.shuhari.local.key.pem 2048
7. # chmod 400 www.shuhari.local.key.pem
8. Generate CSR
9. Openssl req -config subca.cnf -key ww.shuhari.local.key.pem -new -sha256 -out www.shuhari.local.csr.pem
    a. Country Name : IN
    b. State : MH
    c. Locality Name : Pune
    d. Organization Name : ShuHaRi Labs
    e. Organizational Unit : IS
    f. Common Name : www.shuhari.local
    g. Email address : info@shuhari.local
10. Copy CSR to subca
11. # scp www.shuhari.local.csr.pem shuhari@subca:/home/shuhari/subca/csr/
12. Generate and secure the Cert
13. # openssl ca -config subca.cnf -extensions server_cert -days 376 -notext -md sha256 -in csr/www.shuhari.local.csr.pem -out certs/www.shuhari.local.cert.pem …..(in subca)
14. # chmod 444 certs/www.shuhari.local.cert.pem ……(in subca)
15. # scp certs/www.shuhari.local.cert.pem shuhari@www:/home/shuhari/certs/ ……(in subca)
16. Copy ca chain file
17. # scp certs/ca-chain.cert.pem shuhari@www:/home/shuhari/certs/ ……(in subca)
18. Verify chain of trust
19. # openssl verify -CAfile ca-chain.cert.pem www.shuhari.local.cert.pem
20. Configure Apache to use SSL
    a. Copy files to the apache folder and secure the files
    b. Configure apache to use SSL
    c. Test apache using OpenSSL
    d. Test Apache using Browser (FireFox)
    e. Add the Root CA certificate to the Trusted Root Server list
21. # sudo mkdir -p /etc/apache2/ssl
22. # sudo cp ca-chain.cert.pem /etc/apache2/ssl/
23. # sudo cp www.shuhari.local.cert.pem /etc/apache2/ssl/
24. # sudo cp www.shuhari.local.key.pem /etc/apache2/ssl/
25. # sudo chmod 600 /etc/acpache2/ssl/*
26. Configure Apache to use ssl
27. # sudo a2enmod ssl
28. # sudo a2ensite default-ssl
29. # ss -ant | grep 443

30. # sudo systemctl restart apache2
31. # sudo systemctl reload apache2

PKI

By – Manu sir 5-10-19

32. # sudo nano /etc/apache2/sites-enabled/default-ssl.conf .....(make sure you have backup)
    a. Add lines in the file below <virtualHost_default_:443>
        i. ServerAdmin webmaster@localhost
        ii. ServerName www.shuhari.local:443
        iii. below SSLCertificateFile directive change the path
        iv. /etc/apache2/ssl/www.shuhari.local.cert.pem
        v. /etc/apache2/ssl/www.shuhari.local.key.pem
        vi. Below Certificate authority (CA):
        vii. Uncoment the line which contains bundle and edit
        viii. SSLCACertificateFile /etc/apache2/ssl/ca-chain.cert.pem
33. # openssl s_client -connect www.shuhari.local:443

How to digitally sign a PDF document:

1. # apt-get install default-jdk openssl
2. # openssl req -x509 -nodes -days 365 -newkey rsa:1024 -keyout MyKey.key -out Mycert.pem -subj “Generating a 1024 bit RSA private key”
3. We have to convert pem file to pfx file
4. # openssl pkcs12 -export -out MyCert.pfx -in MyCert.pem -inkey MyKey.Key
5. We have to download the portable signer installer under root
6. # mkdir ps1
7. # java -jar PortableSigner-Install-2.0.38c0573.jar
8. # wget -q http://192.168.74.74/sw/utilities/one.pdf
9. # cd ps1
10. # java -jar PortableSigner.jar -n -b en -t /root/one.pdf -o /root/two.pdf -s /root/MyCert.pfx -p toor -c “DITISS PDF Signing Demo” -r “Demo Class” -l “pune”
11. # apt-get install libimage-exiftool-perl
12. # exiftool -a <two.pdf>

IIS self signed certificate.
Apache self signed certificate.
Passwordless authentication into Debian to linux and Debian to windows.
Gpg symmetric ASCII and symmetric Binary.
Gpg Asymmetric ASCII and Symmetric Binary.
Windows cipher.
PDF signing.
Rootca, subca