Public key cryptography in OpenSSL
HW6 - CNS Sapienza

Lombardo Andrea 1893440

12/12/2019

1 Introduction

The main goal of this abstract is to get familiar with :

- generating keys
- generating certificates
- converting certificates
- digital signing

Using command line from bash I manage to generating a pairs of keys(public and private) RSA and DSA through genpkeys options of the openssl tool. Then I need to sign and verify a file and also in that case I use the OpenSSL command line tool.

2 Design

Generate an RSA keypair with a 2048 bit private key:
openssl genpkey -algorithm RSA -out private_rsa_key.pem -pkeyopt rsa_keygen_bits:2048

instead for the DSA keypair with a 2048 bit first I should create a parameter for building the private key:
openssl genpkey -genparam -algorithm DSA -out parameter_dsa_key.pem -pkeyopt dsa_paramgen_bits:2048

Extracting the public key from an RSA keypair and put it in a new file, public_rsa_key.pem
openssl rsa -pubout -in private_rsa_key.pem -out public_rsa_key.pem
instead for DSA first I should define the private_key:
openssl genpkey -paramfile parameter_dsa_key.pem -out private_dsa_key.pem
after this
make sure to prevent other users from reading your keys allowing the rights after defining both private_keys:
chmod go-r private_rsa_key.pem && chmod go-r paramater_dsa_key.pem
I evaluate the follwing command:
openssl dsa -pubout -in private_dsa_key.pem -out public_dsa_key.pem

Now I should create a self-certification: **openssl req -x509 -new -key private_rsa_key.pem -out certification_rsa.pem** and then according to the following script I can check if it's ok: **openssl x509 -in certification_rsa.pem -keyform PEM -text -noout**

Now we can see that modifying the word from rsa to dsa we can build and check DSA certification.

Now for the digital signature I take a file called "data.txt" and insert inside a content choice randomic and according the keys built before I should use the following scripts:
openssl dgst -sign private_rsa_key.pem -keyform PEM -sha256 -out data.txt.sign -binary data.txt

instead for verify the signature I should use the following scripts:
openssl dgst -verify public_rsa_key.pem -keyform PEM -sha256 -signature data.txt.sign -binary data.txt .
Now first I sign the file into a new file called data.txt.sign and then I check if the bash return "Verified OK" that measn that everything is OK. How is it describe in the article in the following link:
"https://www.go4expert.com/articles/digital-certificate-formats-filename-t24831/" is possible to modify the format of the X.509 certificates: where the most common formats are

- **em - (Privacy Enhanced Mail) - PEM**
- **cer, .crt, .der,**
- **p7b, .p7c - PKCS#7 - PKCS #7,p12 - PKCS#12 ,**
- **pfx - PFX (Personal Information Exchange)**

1) **convert an x509 certificate from cer to PEM format:** openssl x509 –in cert.cer –out cert.pem

2) **Convert PEM Format Certificate to PFX Format Certificate:** openssl x509 pkcs12 -export -out certificate.pfx -inkey rsa_key.key -in certificate.pem

3 7-Homework

In this seven homework I develop the job through a Netkit lab. First I use a simple netkit lab made during the Network Infrastracture course where the net is composde by two Virtual Machines and a router. They are able to exchange messages or files through netcat tool. The right ip where one should be connected could be available through the script in bash "ip a " in pc1 bash or pc2 bash after the "lstart -p0" which let me to identify what is the ip . Before of this I create a copy of the openssl.cnf from /usr/lib/ssl to the directory used for the homework and I call it root.cnf. I identify pc1 as the server and pc2 as the client. First I follow the scripts applied in the link :**https://roll.urown.net/ca/ca_root_setup.html**. Then I modify the root.cnf modifying the directory and the different files in a way that is conform to the link selected above. Then first I create the certification of pc1 and then with netcat I pass the root.cnf to my client file so in pc2's directory. Now pc2 is able to generate private key, public key and request that should be signed by pc1. So after creating the request I send it to pc1, through netcat, it should be responsible to sign it with the certification. After this we can see the revokations and the list of certifications. After the descriptions we could see the different scripts:

3.1 pc2

```
mkdir -p example.net.ca/root-ca/certreqs,certs,crl,newcerts,private
cd example.net.ca/root-ca
chmod 700 private
touch root-ca.txt
echo 00 > root-ca.crlnum
openssl rand -hex 16 > root-ca.serial
openssl genrsa -aes256 -out private_key.pem 2048
openssl rsa -pbout -in private_key.pem -out public_key.pem
nc -l -p 9999 -vv >root.cnf
openssl req -new -sha256 -out request.pem
nc 10.0.0.100 9999 -vv<request.pem
```

3.2 pc1

```
nc 10.0.1.100 9999 -vv <root.cnf
nc -l -p 9999 -vv >request.pem
openssl ca -config root.cnf -extensions usr_ert -days 90 -notext -md sha256
-in request.pem -out request_signature.pem
nc 10.0.1.100 9999 -vv <request_signature.pem
openssl ca-config root.cnf -revoke request_signature.pem
echo 1234 >root-cacrlnum
openssl ca -config root.cnf -gencrl -out crl.pem
openssl crl -in rl.pem -noout -text
cat root-ca.txt-> list of certificates
```

References

[1] https://en.wikibooks.org/wiki/Cryptography/Generate_a_keypair_using_OpenSSL

[2] https://gist.github.com/tsaarni/14f31312315b46f06e0f1ecc37146bf3

[3] https://roll.urown.net/ca/ca_root_setup.html

[4] https://www.pillolhacking.net/2008/11/06/trasferire-file-con-netcat/

[5] https://stackoverflow.com/questions/10782826/digital-signature-for-a-file-using-o

[6] https://rietta.com/blog/openssl-generating-rsa-key-from-command/

[7] https://developers.yubico.com/PIV/Guides/Generating_keys_using_OpenSSL.html

[8] https://security.stackexchange.com/questions/5096/rsa-vs-dsa-for-ssh-authenticati

[9] https://www.ssh.com/ssh/keygen/

[10] https://security.stackexchange.com/questions/161526/why-does-generating-a-self-s

[11] https://www.zimuel.it/blog/sign-and-verify-a-file-using-openssl

[12] https://www.openssl.org/docs/manmaster/man1/genpkey.html

[13] https://stackoverflow.com/questions/21297139/how-do-you-sign-a-certificate-signi

[14] https://ingegneria.online/questions/146306/openssl-genera-diversi-tipi-di-certif