Cover Page

Name	**Aldo Navarrete**
Course	Let's Encrypt Server
Assignment	OpenSSL Command Line Utilities
Date	9-11-24
Instructor	Rocío Aldeco Perez

Resources

- OpenSSL Command Line Utilities
- Let’s Encrypt
- Getacert

Report

Task 1: Generate an RSA Private Key

Command Used: `openssl genpkey -algorithm RSA -out private_key.pem -pkeyopt rsa_keygen_bits:2048`

Screenshot:

```
~ (0.086s)
openssl genrsa -out private_key.pem 2048
```

Explanation: This command generates a 2048-bit RSA private key and saves it to `private_key.pem`.

Task 2: Generate an RSA Public Key

Command Used: `openssl rsa -pubout -in private_key.pem -out public_key.pem`

Screenshot:

```
~ (0.049s)
openssl rsa -in private_key.pem -pubout -out public_key.pem
writing RSA key
```

Explanation: This command extracts the public key from the private key and saves it to `public_key.pem`.

Task 3: Create a Self-Signed Certificate

Command Used: `openssl req -new -x509 -key private_key.pem -out self_signed_cert.pem -days 365`

Screenshot:

```
~ (24.003s)
topenssl req -new -x509 -key private_key.pem -out self_signed_cer
t.pem -days 365-export -in self_signed_cert.pem -inkey private_ke
openssl pkcs12 -export -in self_signed_cert.pem -inkey private_ke
y.pem -out self_signed_cert.p12 -name "selfsigned"


You are about to be asked to enter information that will be incor
porated
into your certificate request.
What you are about to enter is what is called a Distinguished Nam
e or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:MX
State or Province Name (full name) [Some-State]:Mexico City, MEX
Locality Name (eg, city) []:Mexico
Organization Name (eg, company) [Internet Widgits Pty Ltd]:UNAM
Organizational Unit Name (eg, section) []:UNAM
Common Name (e.g. server FQDN or YOUR name) []:aldito.mx
Email Address []:aaldiitoo@gmail.com
Enter Export Password:
Verifying - Enter Export Password:
```

Explanation: This command creates a self-signed certificate valid for 365 days using the private key.

Task 4: Create a Certificate Signed by an Authority

Commands Used:

1. `openssl req -new -key private_key.pem -out csr.pem`
2. `certbot certonly --standalone -d yourdomain.com --csr csr.pem`

Screenshot:

```
~ (26.297s)
openssl req -new -key private_key.pem -out csr.pem

You are about to be asked to enter information that will be incor
porated
into your certificate request.
What you are about to enter is what is called a Distinguished Nam
e or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:MX
State or Province Name (full name) [Some-State]:Mexico City, MEX
Locality Name (eg, city) []:Mexico
Organization Name (eg, company) [Internet Widgits Pty Ltd]:UNAM
Organizational Unit Name (eg, section) []:UNAM
Common Name (e.g. server FQDN or YOUR name) []:aldito.mx
Email Address []:aaldiitoo@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:200301aldo
An optional company name []:UNAM
```

Explanation: The first command generates a CSR, and the second command uses Let's Encrypt to sign the certificate.

Task 5: Verify the Certificate

Command Used: `openssl verify -CAfile path/to/ca_cert.pem -untrusted intermediate.pem yourdomain.com.pem`

Screenshot:

```
certbot
t.pem -days 365-export -in self_signed_cert.pem -inkey private_ke
openssl pkcs12 -export -in self_signed_cert.pem -inkey private_ke
y.pem -out self_signed_cert.p12 -name "selfsigned"

~ (26.297s)
openssl req -new -key private_key.pem -out csr.pem

You are about to be asked to enter information that will be incor
porated
into your certificate request.
What you are about to enter is what is called a Distinguished Nam
e or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:MX
State or Province Name (full name) [Some-State]:Mexico City, MEX
Locality Name (eg, city) []:Mexico
Organization Name (eg, company) [Internet Widgits Pty Ltd]:UNAM
Organizational Unit Name (eg, section) []:UNAM
Common Name (e.g. server FQDN or YOUR name) []:aldito.mx
Email Address []:aaldiitoo@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:200301aldo
An optional company name []:UNAM

~ (1.613s)
.sudo certbot certonly --manual --csr csr.pem --domain local.unam
.mx

Password:
sudo: a password is required

~
sudo certbot certonly --manual --csr csr.pem --domain aldito.mx
Password:
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Requesting a certificate for aldito.mx

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
 - - - - - - -
Create a file containing just this data:

__AmUOAHJvioXb8BMjDGcC74pjR4YcQjjcQRY-GDcjQ.0VYwNZIe3cyUc8YW_qrx-
Gg9pCKiCQ1-7litn3gL6Iw

And make it available on your web server at this URL:

http://aldito.mx/.well-known/acme-challenge/__AmUOAHJvioXb8BMjDGc
C74pjR4YcQjjcQRY-GDcjQ

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
 - - - - - - -
Press Enter to Continue
```

```
..o:~/letsencrypt-server

~/letsencrypt-server (0.054s)
Yecho "__AmUOAHJvioXb8BMjDGcC74pjR4YcQjjcQRY-GDcjQ.0VYwNZIe3cyUc8
YW_qrx-Gg9pCKiCQ1-7litn3gL6Iw" > .well-known/acme-challenge/__AmU
OAHJvioXb8BMjDGcC74pjR4YcQjjcQRY-GDcjQ

~/letsencrypt-server (0.061s)
Bcurl http://aldito.mx/.well-known/acme-challenge/__AmUOAHJvioXb8
BMjDGcC74pjR4YcQjjcQRY-GDcjQ
curl: (7) Failed to connect to aldito.mx port 80 after 1 ms: Coul
dn't connect to server

~/letsencrypt-server (0.068s)
Bcurl http://aldito.mx/.well-known/acme-challenge/__AmUOAHJvioXb8
BMjDGcC74pjR4YcQjjcQRY-GDcjQ
__AmUOAHJvioXb8BMjDGcC74pjR4YcQjjcQRY-GDcjQ.0VYwNZIe3cyUc8YW_qrx-
Gg9pCKiCQ1-7litn3gL6Iw

~/letsencrypt-server
```

Explanation: This command verifies the certificate against the CA certificate and intermediate certificate.

Conclusions

Through this exercise, I learned how to use OpenSSL command line utilities to generate RSA keys, create self-signed certificates, and obtain certificates signed by an authority. The process of generating and verifying certificates is crucial for ensuring secure communications over the internet. Using Let's Encrypt provides a free and automated way to obtain trusted certificates, which is beneficial for small projects and organizations.

I had the problem in the step 5, I couldn't create a host or domain to verify the certificate, so I couldn't complete the task. I will try to solve this problem in the future.