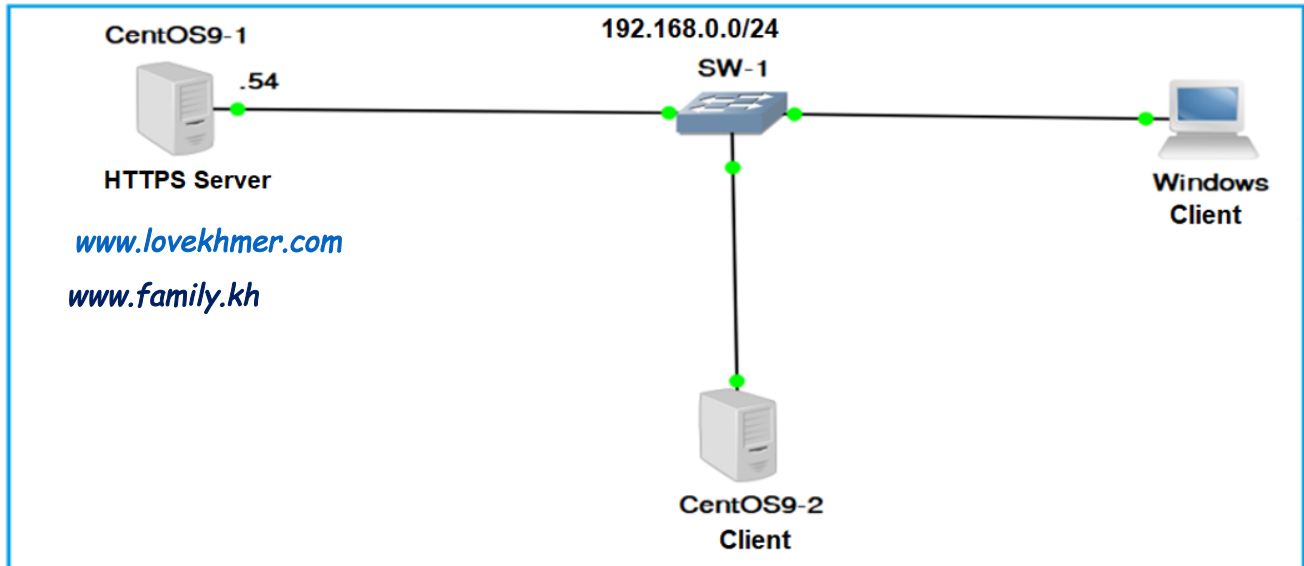


Lab (HTTPS)-Configure Apache HTTPS Server with Self-signed Certificate on Linux CentOS9

○ Network Diagram



INTRODUCTION

In this lab we will host two websites using Self-Signed Digital Certificate.

Digital certificates play a crucial role in securing online communications by verifying the identity of entities and encrypting the traffic between a web server and a web browser, enhances Data Security and increases trust of users. These certificates are issued by trusted Certificate Authorities (CAs) and are used in various security protocols, such as SSL/TLS.

What is a Digital Certificate?

A digital certificate is an electronic document used to prove the ownership of a public key. It includes information about the key, its owner's identity, and the digital signature of an entity that has verified the certificate's contents. Certificates are part of the public key infrastructure (PKI) and

are used to establish secure connections over the internet, ensuring that data transmitted between parties is encrypted and secure.

I. What is a Self-Signed Certificate?

A self-signed certificate is a digital certificate that is not signed by a trusted third-party Certificate Authority (CA) but instead is signed by the entity that will be using the certificate.

These self-signed certificates are easy to make and do not cost money. However, they do not provide any trust value. For instance, if a website owner uses a self-signed certificate to provide HTTPS services, people who visit that website cannot be certain that they are connected to their intended destination. For all they know, a malicious third-party could be redirecting the connection using another self-signed certificate bearing the same holder name. The connection is still encrypted, but does not necessarily lead to its intended target. In comparison, a certificate signed by a trusted CA prevents this attack because the user's web browser separately validates the certificate against the issuing CA. The attacker's certificate fails this validation.

Unlike certificates issued by a CA, self-signed certificates are not inherently trusted by other systems and require manual configuration to be trusted. They are often used in development and testing environments where setting up a CA is not necessary. "Self-signed" means that the public key embedded in the certificate validates the signature on the certificate.

>> How it works:

The entity creates a private key and uses it to sign the certificate.

>> Common Use Cases:

-Internal Networks: They are commonly used to secure communication within a private network or for internal applications.

-Development and Testing: They are used for testing SSL/TLS connections in a development environment.

-Local Servers: They can be used to secure local web servers for personal use or testing purposes.

>>Limitations:

-No inherent trust: Browsers and other systems will typically not trust self-signed certificates by default, leading to warnings or errors.

-Manual configuration: You need to manually configure the client systems to trust the self-signed certificate.

-Security concerns: Self-signed certificates can be a security risk if not properly managed, as they can be compromised and used to impersonate a website or server.

>>Alternatives:

For public-facing websites or applications, it's recommended to use certificates issued by a trusted CA.

II. What is a CA-Signed Certificate?

A CA-signed certificate is a digital certificate issued by a Certificate Authority (CA) that verifies the identity of the certificate holder. It ensures that the certificate holder is trusted and authentic.

ជំហាន (Steps):

1. Install Some Packages that require to configure apache web server with SSL encryption: **httpd**, openssl & mod_ssl

>> Install

```
yum install httpd openssl mod_ssl -y
```

>> Verify

```
[root@linuxserver1 ~]# rpm -q httpd mod_ssl openssl  
httpd-2.4.62-4.el9.x86_64
```

`mod_ssl-2.4.62-4.el9.x86_64`

`openssl-3.2.2-7.el9.x86_64`

`[root@linuxserver1 ~]# rpm -qi mod_ssl`

Name : mod_ssl

Epoch : 1

Version : 2.4.62

Release : 4.el9

Architecture: x86_64

Install Date: Thu 20 Mar 2025 04:39:51 PM +07

Group : Unspecified

Size : 278926

License : ASL 2.0

Signature : RSA/SHA256, Fri 31 Jan 2025 10:15:36 PM +07, Key ID
05b555b38483c65d

Source RPM : httpd-2.4.62-4.el9.src.rpm

Build Date : Thu 30 Jan 2025 12:48:34 AM +07

Build Host : x86-01.stream.rdu2.redhat.com

Packager : builder@centos.org

Vendor : CentOS

URL : <https://httpd.apache.org/>

Summary : SSL/TLS module for the Apache HTTP Server

Description :

The mod_ssl module provides strong cryptography for the Apache HTTP server via the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols.

`[root@linuxserver1 ~]#`

`[root@linuxserver1 ~]# rpm -qi openssl`

Name : openssl

Epoch : 1

Version : 3.2.2

Release : 7.el9

Architecture: x86_64

into your certificate request.

What you are about to enter is what is called a Distinguished Name 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) [XX]:CD

State or Province Name (full name) [:]TAKEO

Locality Name (eg, city) [Default City]:PHNOM PENH

Organization Name (eg, company) [Default Company Ltd]:RUPP

Organizational Unit Name (eg, section) [:]ITd

Common Name (eg, your name or your server's hostname) [:]linuxserver1

Email Address [:]kim.no120282kh@gmail.com

[root@linuxserver1 ~]# ls

anaconda-ks.cfg Desktop Downloads mycerti.cert Pictures Templates

cert-dir Documents Music mypri-key.key Public Videos

យើងទទួលបាន SSL certificate (mycerti.cert) & Private key (mypri-key.key) ដូច
តម្រូវការ។

mypri-key.key is the private key associated with your certificate. It must be kept
secure and private.

mycerti.cert is the public certificate that will be shared with clients connecting to
your server. It includes the public key and information about your server.

3. Copy certificate & private key to their default directory

[root@linuxserver1 ~]# cp mycerti.cert /etc/pki/tls/certs/

[root@linuxserver1 ~]# cp mypri-key.key /etc/pki/tls/private/

[root@linuxserver1 ~]#

[root@linuxserver1 ~]# ls /etc/pki/tls/certs/

ca-bundle.crt ca-bundle.trust.crt cer.crt localhost.crt mycerti.cert
postfix.pem

[root@linuxserver1 ~]# ls /etc/pki/tls/private/

cer.csr cer.key localhost.key mypri-key.key postfix.key

[root@linuxserver1 ~]#

4. Edit Configuration file for the SSL (ssl.conf)

```
[root@linuxserver1 ~]# vim /etc/httpd/conf.d/ssl.conf
#Line 83: Change certificate name from the localhost.crt to mycerti.cert
SSLCertificateFile /etc/pki/tls/certs/mycerti.cert
#Line 91: Change private key file name from the localhost.key to mypri-key.key
SSLCertificateKeyFile /etc/pki/tls/private/mypri-key.key
```

>> Save & Exit from ssl.conf file

esc:wq

5. Create apache https configuration file (httpd.conf) to host 2 websites

```
[root@linuxserver1 ~]# vim /etc/httpd/conf.d/httpd.conf
<VirtualHost *:443>
SSLEngine on
SSLCertificateFile /etc/pki/tls/certs/mycerti.cert
SSLCertificateKeyFile /etc/pki/tls/private/mypri-key.key
ServerName www.lovekhmer.com
DocumentRoot /var/www/html/lovekhmer.com
</VirtualHost>
<VirtualHost *:443>
SSLEngine on
SSLCertificateFile /etc/pki/tls/certs/mycerti.cert
SSLCertificateKeyFile /etc/pki/tls/private/mypri-key.key
ServerName www.family.kh
DocumentRoot /var/www/html/family.kh
</VirtualHost>
```

>> Save & Exit from httpd.conf file

esc:wq

6. Create lovekhmer.com & linuxcen.net for storing website file

```
[root@linuxserver1 ~]# cd /var/www/html/
[root@linuxserver1 html]# mkdir lovekhmer.com family.kh
```


7. Copy website file into each directory

8. Check Selinux

```
[root@linuxserver1 html]# ls -ldZ lovekhmer.com linuxcen.net/  
drwxr-xr-x. 2 root root unconfined_u:object_r:httpd_sys_content_t:s0 24  
Mar 26 09:46 linuxcen.net/  
drwxr-xr-x. 3 root root unconfined_u:object_r:httpd_sys_content_t:s0 37  
Mar 26 09:46 lovekhmer.com  
[root@linuxserver1 html]#  
[root@linuxserver1 html]#
```

9. Check Syntax Error

```
[root@linuxserver1 html]# httpd -t  
Syntax OK  
[root@linuxserver1 html]#
```

10. Start & Enable httpd service

```
[root@linuxserver1 html]# systemctl start httpd  
[root@linuxserver1 html]# systemctl enable httpd  
[root@linuxserver1 html]#
```

11. Firewall

```
[root@linuxserver1 html]# firewall-cmd --add-port=443/tcp --permanent  
[root@linuxserver1 html]# firewall-cmd --reload  
[root@linuxserver1 html]#  
[root@linuxserver1 html]#
```

12. DNS Server

>> Create 2 Forward Lookup Zones (lovekhmer.com & family.kh)

>> Create 2 forward Lookup Zone Files

សូមមើល Lab Primary DNS Server

13. Test access from web clients with https

>> Linux Client

<https://www.lovekhmer.com>

<https://www.family.kh>

Activities Firefox Mar 28 16:38 en

លោកគ្រូ គឹម នៅ <https://www.lovekhmer.com>

CentOS Blog Documentation Forums



លោកគ្រូ គឹម នៅ

សាស្ត្រាចារ្យជំនួយ

សាកលវិទ្យាល័យភូមិន្ទភ្នំពេញ

~~ទស្សនវិស័យ ~~


រៀនច្រើនកើនចំណេះ ធ្វើច្រើនកើនជំនាញ ទាញបានទាំងចំណូល ចូលរួមកំណើនសេដ្ឋកិច្ចក្រសួង និងសង្គម !!!



Activities Firefox Mar 28 16:40 en

លោកគ្រូ គឹម នៅ <https://www.family.kh>

CentOS Blog Documentation Forums



~~គ្រួសារខ្ញុំសុភមង្គលខ្ញុំ~~

សុភមង្គលក្នុងគ្រួសារគឺជាសេចក្តីប្រាថ្នារបស់សមាជិកគ្រប់រូប គួរស្វាមីភរិយាចង់បានក្តីស្រឡាញ់ការយល់ចិត្ត និងភាពស្មោះត្រង់ចំពោះគ្នា ទៅវិញទៅមក កូនចង់បានក្តីស្រឡាញ់ថ្នាក់ថ្នម ការលើកទឹកចិត្តពីឪពុកម្តាយ ចំណែកឯឪពុកម្តាយចង់អោយកូនស្រឡាញ់ និងចេះគោរព

ស្តាប់បង្គាប់...។

យោងលើលទ្ធផលនៃការតេស្តខាងលើសបញ្ជាក់ឱ្យឃើញថា *Apache HTTPS Server with Self-signed Certificate* បានរៀបចំត្រឹមត្រូវ ដូចការត្រាងទុក។