Configuring SSL Certificate for Secure Web Server Communication

Introduction

This guide outlines the steps to secure an Apache web server with an SSL certificate, ensuring encrypted communication over HTTPS.

Step 1: Obtain an SSL Certificate

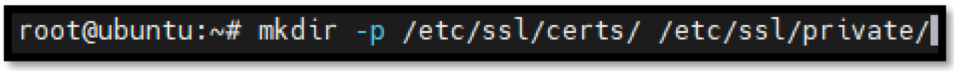
Generate a Self-Signed Certificate (For Testing)

1. Create directories to store certificate and key.

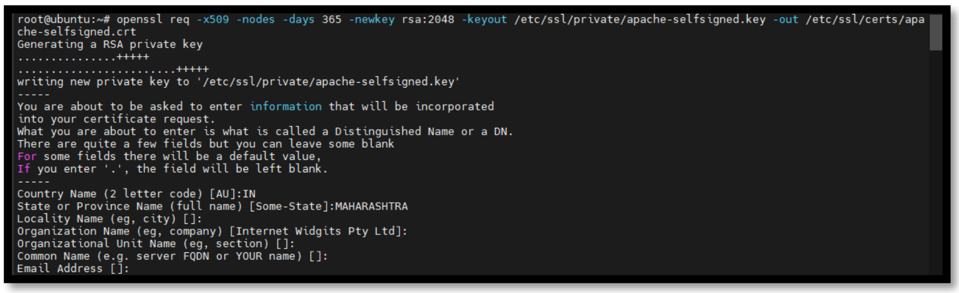
   Cmd: mkdir -p /etc/ssl/certs/ /etc/ssl/private/

2. Run the following command to generate a self-signed certificate:

   Cmd :openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt

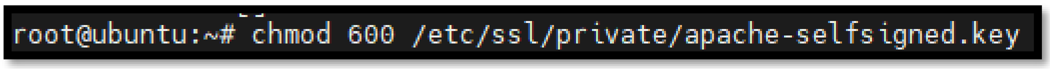


```
root@ubuntu:~# mkdir -p /etc/ssl/certs/ /etc/ssl/private/
```



```
root@ubuntu:~# openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apa
che-selfsigned.crt
Generating a RSA private key
...............+++++
.........................+++++
writing new private key to '/etc/ssl/private/apache-selfsigned.key'
-----
You are about to be asked to enter information that will be incorporated
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) [AU]:IN
State or Province Name (full name) [Some-State]:MAHARASHTRA
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:
```

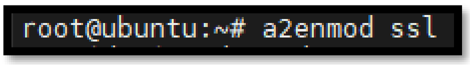
3. Store the certificate and key in **/etc/ssl/certs/ and /etc/ssl/private/** respectively.
4. Give permissions to key file .



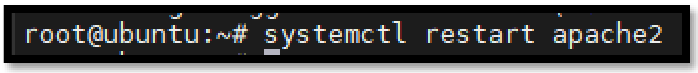
```
root@ubuntu:~# chmod 600 /etc/ssl/private/apache-selfsigned.key
```

Step 2: Configure Apache for SSL

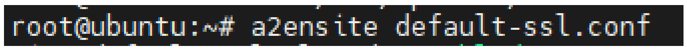
1. **Install SSL modules (if not already installed):**
2. **a2enmod ssl**
3. **systemctl restart apache2**
4. **enable the ssl virtual host and reload the service.**



```
root@ubuntu:~# a2enmod ssl
```



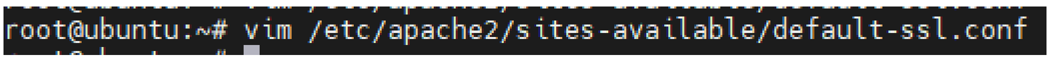
```
root@ubuntu:~# systemctl restart apache2
```



```
root@ubuntu:~# a2ensite default-ssl.conf
```

5. **Edit the Apache SSL configuration file:**

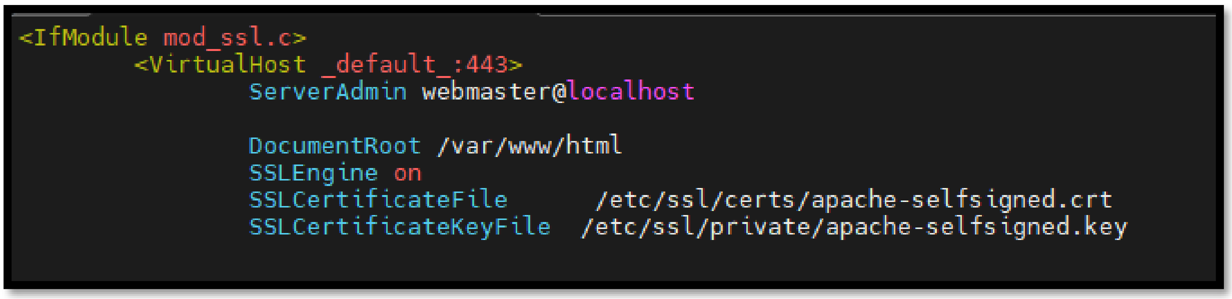
   Cmd: vim /etc/apache2/sites-available/default-ssl.conf



```
root@ubuntu:~# vim /etc/apache2/sites-available/default-ssl.conf
```

6. **Update the file with the correct SSL certificate paths:**

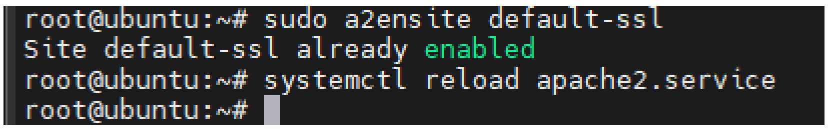
<VirtualHost *:443>
ServerName yourdomain.com
DocumentRoot /var/www/html
SSLEngine on
SSLCertificateFile   /etc/ssl/certs/apache-selfsigned.crt
SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key
</VirtualHost>



```
<IfModule mod_ssl.c>
        <VirtualHost _default_:443>
                ServerAdmin webmaster@localhost

                DocumentRoot /var/www/html
                SSLEngine on
                SSLCertificateFile      /etc/ssl/certs/apache-selfsigned.crt
                SSLCertificateKeyFile  /etc/ssl/private/apache-selfsigned.key
```

7. **Enable the SSL site and restart Apache:**
8. **sudo a2ensite default-ssl**
9. **systemctl restart apache2**

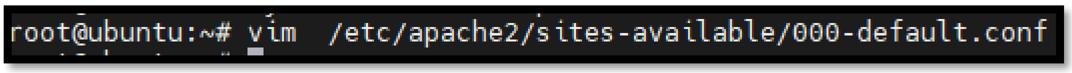


```
root@ubuntu:~# sudo a2ensite default-ssl
Site default-ssl already enabled
root@ubuntu:~# systemctl reload apache2.service
root@ubuntu:~#
```

Step 3: Redirect HTTP to HTTPS

1. Open the default Apache configuration file:

Cmd :vim /etc/apache2/sites-available/000-default.conf



```
root@ubuntu:~# vim  /etc/apache2/sites-available/000-default.conf
```

Add the following redirect rule:

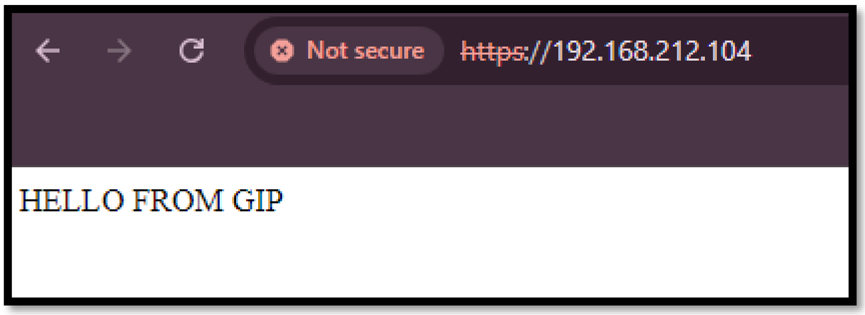
```
<VirtualHost *:80>
 ServerName yourdomain.com
 Redirect permanent / https://yourdomain.com/
</VirtualHost>
```

2. **Restart Apache:**

Cmd:sudo systemctl restart apache2

Step 4: Test and Verify HTTPS Access

1. Open a web browser and navigate to **https://yourdomain.com**.
2. Verify that there are no security warnings.



HELLO FROM GIP

Acceptance Criteria

✓ Apache is correctly configured with the SSL certificate.
✓ The website is accessible via HTTPS without security warnings.
✓ HTTP requests are redirected to HTTPS.