Apache SSL

[**Introducción**](#_dmum6bc62477) **1**

[**Referencias**](#_xff91m7d4ye5) **1**

[**Requerimientos**](#_64p1zffu858f) **1**

[**Configuración**](#_80hfcq1to57d) **1**

[Crear certificado con OpenSSL](#_s542xnhmu169) 1

[Apache](#_njxruib7430k) 2

[**Resultado**](#_m4wk7age3xl4) **4**

# Introducción

Vamos a configurar acceso por HTTPS en Apache.

# Referencias

* Digital Ocean: <https://www.digitalocean.com/community/tutorials/how-to-create-a-ssl-certificate-on-apache-for-ubuntu-14-04>
* OpenCms <http://documentation.opencms.org/opencms-documentation/server-installation/apache-webserver-configuration/>

# Requerimientos

Entorno con:

* Apache2
* Tomcat
* OpenCms
* OpenSSL

# Configuración

## Crear certificado con OpenSSL

1. Creamos una carpeta para almacenar certificados

|  |
| --- |
| $ sudo mkdir /etc/apache2/ssl/ |

1. Creamos un certificado X.509 (CSR) válido por 356 días en /etc/apache2/ssl/

|  |
| --- |
| $ sudo openssl req -new -x509 -days 365 -out /etc/apache2/ssl/server-cert.pem -keyout /etc/apache2/ssl/server-rsa-key.pem  -----  Country Name (2 letter code) [AU]:ES  State or Province Name (full name) [Some-State]:Andalucía  Locality Name (eg, city) []:Sevilla  Organization Name (eg, company) [Internet Widgits Pty Ltd]:Saga  Organizational Unit Name (eg, section) []:Desarrollo Web  Common Name (e.g. server FQDN or YOUR name) []:demilitest.ocms71.sagasoluciones.com  Email Address []:jesus.gregorio@sagasoluciones.com |

1. Creamos fichero con la contraseña del certificado para cargarla automáticamente al arrancar. Debemos crearlo accesible exclusivamente para el usuario root o apache.

|  |
| --- |
| $ sudo nano /etc/apache2/ssl/ppd.conf  $ sudo chmod 600 ppd.conf |

## Apache

1. Habilitamos módulo SSL

|  |
| --- |
| $ sudo a2enmod ssl  $ sudo systemctl restart apache2.service |

1. Aseguramos que Apache escuche el puerto 443 en el fichero /etc/apache2/ports.conf

|  |
| --- |
| $ sudo nano /etc/apache2/ports.conf  [...]  Listen 443  [...] |

1. Configuramos acceso a la contraseña del certificado

|  |
| --- |
| $ sudo nano /etc/apache2/mods-enabled/ssl.conf  # Contraseña de certificado específica  # SSLPassPhraseDialog exec:/usr/share/apache2/ask-for-passphrase  SSLPassPhraseDialog exec:/etc/apache2/ssl/ppd.conf |

1. Creamos VH para acceso SSL

|  |
| --- |
| $ sudo cp /etc/apache2/sites-enabled/opencms\_demilitest.ocms71.sagasoluciones.com /etc/apache2/sites-available/opencms\_demilitest.ocms71.sagasoluciones.com\_ssl.conf  $ sudo nano /etc/apache2/sites-available/opencms\_demilitest.ocms71.sagasoluciones.com\_ssl.conf |

1. Añadimos la configuración para acceder por el puerto 443

|  |
| --- |
| <VirtualHost \*:443>  […]  #Enable the SSLEngine  SSLEngine On  SSLProxyEngine On  SSLCertificateFile /etc/apache2/ssl/server-cert.pem  SSLCertificateKeyFile /etc/apache2/ssl/server-rsa-key.pem  […] |

1. Habilitamos nuevo VH

|  |
| --- |
| $ cd /etc/apache2/sites-enabled/  $ sudo ln -s ../sites-available/opencms\_demilitest.ocms71.sagasoluciones.com\_ssl.conf  $ sudo systemctl restart apache2.service |

### Para habilitar acceso por HTTPS mediante apache externo y redireccionar internamente con HTTP

|  |
| --- |
| <VirtualHost \*:443>  # The ServerName directive sets the request scheme, hostname and port that  # the server uses to identify itself. This is used when creating  # redirection URLs. In the context of virtual hosts, the ServerName  # specifies what hostname must appear in the request's Host: header to  # match this virtual host. For the default virtual host (this file) this  # value is not decisive as it is used as a last resort host regardless.  # However, you must set it for any further virtual host explicitly.  ServerName ocms71.sagasoluciones.com  ServerAlias \*.ocms71.sagasoluciones.com  ServerAdmin webmaster@localhost  DocumentRoot /var/www/html  # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,  # error, crit, alert, emerg.  # It is also possible to configure the loglevel for particular  # modules, e.g.  #LogLevel info ssl:warn  # Logging  LogFormat "%h %l %u %t \"%r\" %>s %b" common  CustomLog /var/log/apache2/access\_ocms71.sagasoluciones.com.log common  ErrorLog /var/log/apache2/error\_ocms71.sagasoluciones.com.log  # Enable the SSLEngine  SSLEngine On  SSLProxyEngine On  SSLCertificateFile /etc/apache2/ssl/server-cert.pem  SSLCertificateKeyFile /etc/apache2/ssl/server-rsa-key.pem  # Proxy  ProxyPreserveHost On  ProxyPass / http://192.168.1.71/  ProxyPassReverse / http://192.168.1.71/  ProxyPassReverseCookiePath / /  # For most configuration files from conf-available/, which are  # enabled or disabled at a global level, it is possible to  # include a line for only one particular virtual host. For example the  # following line enables the CGI configuration for this host only  # after it has been globally disabled with "a2disconf".  #Include conf-available/serve-cgi-bin.conf  </VirtualHost>  # vim: syntax=apache ts=4 sw=4 sts=4 sr noet |

### Para forzar el acceso por HTTPS

Añadir en el VH:

|  |
| --- |
| RewriteEngine On RewriteCond %{HTTPS} off RewriteRule (.\*) https://%{HTTP\_HOST}%{REQUEST\_URI} |

* Plantilla de VH con acceso SSL

|  |
| --- |
| <VirtualHost \*:443>  DocumentRoot "$DOCUMENT\_ROOT$"  ServerName $SERVER\_NAME\_WITH\_PORT$  $ALIAS\_DIRECTIVE$ $SERVER\_ALIASES$  ServerAdmin webmaster@$SERVER\_NAME\_WITH\_PORT$  # Allow accessing the document root directory  <Directory $DOCUMENT\_ROOT$/resources>  Options FollowSymlinks  AllowOverride All  Order allow,deny  Allow from all  Require all granted  </Directory>  <Directory $DOCUMENT\_ROOT$/export>  Options FollowSymlinks  AllowOverride All  Order allow,deny  Allow from all  Require all granted  </Directory>  # Logging  LogFormat "%h %l %u %t \"%r\" %>s %b" common  CustomLog $LOGGING\_DIRECTORY$access\_$CONFIG\_FILENAME$.log common  ErrorLog $LOGGING\_DIRECTORY$error\_$CONFIG\_FILENAME$.log  LogLevel alert rewrite:trace1  # Do not forward the request if the requested URI is located in the resources folder or is static content  SetEnvIfNoCase Request\_URI ^$CONTEXT\_PATH$/resources/.\*\$ no-jk  SetEnvIfNoCase Request\_URI ^/export/.\*\$ no-jk  # Turn rewriting on and define the rules  RewriteEngine On  # Enable the SSLEngine  SSLEngine On  SSLProxyEngine On  SSLCertificateFile /etc/apache2/ssl/server-cert.pem  SSLCertificateKeyFile /etc/apache2/ssl/server-rsa-key.pem  # Deny access to php files in order to prevent reading resources  RewriteCond %{REQUEST\_FILENAME} (.+)\.php(.\*)  RewriteRule (.\*) / [F]  # If the requested URI is NOT located in the resources folder and is not static:  # Prepend an $CONTEXT\_PATH$$SERVLET\_PATH$ to everything that does not already starts with it  # and force the result to be handled by the next URI-handler ([PT]) (JkMount in this case)  RewriteCond %{REQUEST\_URI} !^$CONTEXT\_PATH$/resources/.\*\$  RewriteCond %{REQUEST\_URI} !^/export/.\*  RewriteRule !^$CONTEXT\_PATH$$SERVLET\_PATH$/(.\*)\$ $CONTEXT\_PATH$$SERVLET\_PATH$%{REQUEST\_URI} [PT]    # These are the settings for static export. If the requested resource is not already  # statically exported create a new request to the opencms404 handler. This has to be  # a new request, because the current would not get through mod\_jk because of the "no-jk".  RewriteCond %{REQUEST\_URI} ^/export/.\*\$  RewriteCond "%{DOCUMENT\_ROOT}%{REQUEST\_FILENAME}" !-f  RewriteCond "%{DOCUMENT\_ROOT}%{REQUEST\_FILENAME}/index\_export.html" !-f  RewriteRule .\* http://localhost:8080$CONTEXT\_PATH$$SERVLET\_PATH$/handle404?exporturi=%{REQUEST\_URI}&%{QUERY\_STRING} [P]  # If the request starts with $CONTEXT\_PATH$/resources, substitute the $CONTEXT\_PATH$ prefix  RewriteCond %{REQUEST\_URI} ^$CONTEXT\_PATH$/resources/.\*\$  RewriteRule ^$CONTEXT\_PATH$/(.\*)\$ /\$1  JkMount /\* ocms  </VirtualHost> |

# Resultado

Al acceder por HTTPS: <https://demilitest.ocms71.sagasoluciones.com/>

El navegador debe mostrar una advertencia ya que el certificado no está verificado por ninguna organización autorizada sino que lo hemos firmado nosotros mismos:

|  |
| --- |
|  |

Aceptamos y podremos acceder al site.