Apache Server Hardening steps and commands - Ubuntu

These are the steps, procedures and commands and configuration settings are used to *harden* a default Apache Server installation on Ubuntu Linux 19.04 (Debian).

Installing apache

1) `sudo apt install apache`

Installing OpenSSL server

1) `sudo install openssh-server`

Keep system up to date

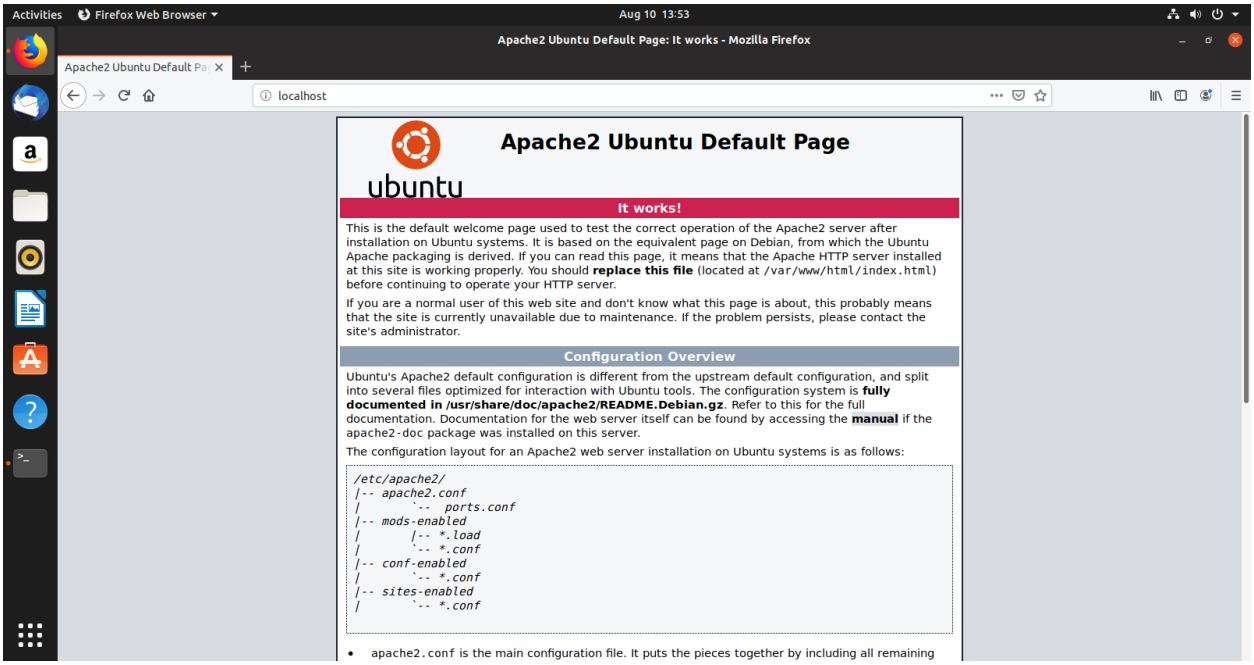
1) `sudo apachectl start`

Using Host Firewall – using ufw (uncomplicated firewall)

1) `sudo ufw status verbose`// check if the Ubuntu firewall is enabled
2) `sudo ufw enable/disable` // enable/disable host firewall
3) `sudo ufw allow from 192.168.x.x port 22` // add firewall rule open port 22 and allow only 192.168.x.x access
4) `sudo ufw delete 1` // disable the first rule in the firewall
5) `sudo ufw delete allow ssh` // remove ssh rule in the firewall

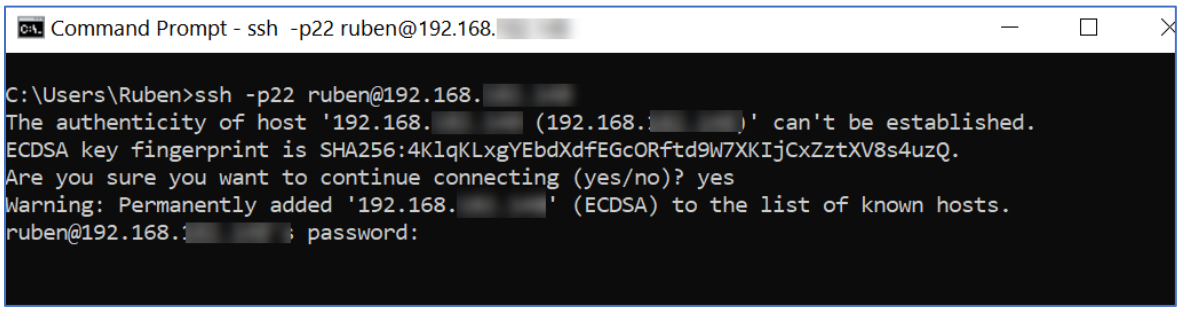
Starting Apache

1) `sudo apachectl -k start`
2) In web browser, type in localhost. The *default Apache2 Ubuntu Default Page* appears (see screen shot).



Connecting to Ubuntu server

1) `ssh -p22 ruben@192.168.x.x` // connect to ubuntu server.

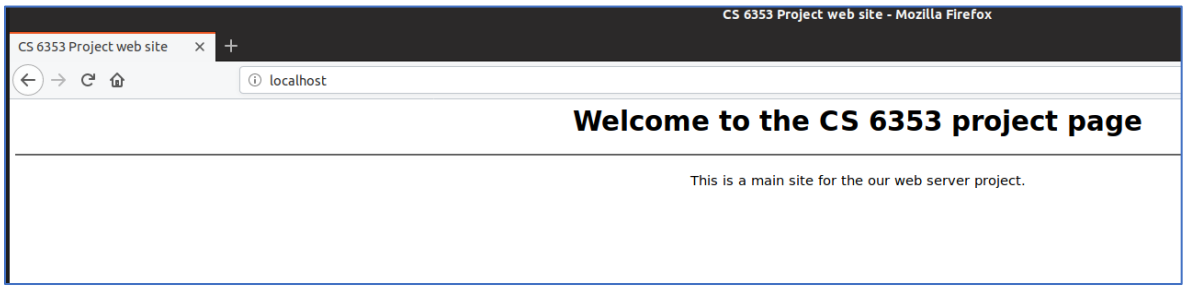


Removing default Apache page index.html

1) `sudo rm index.html`

Edit default Apache page index.html

1) `sudo vim index.html`

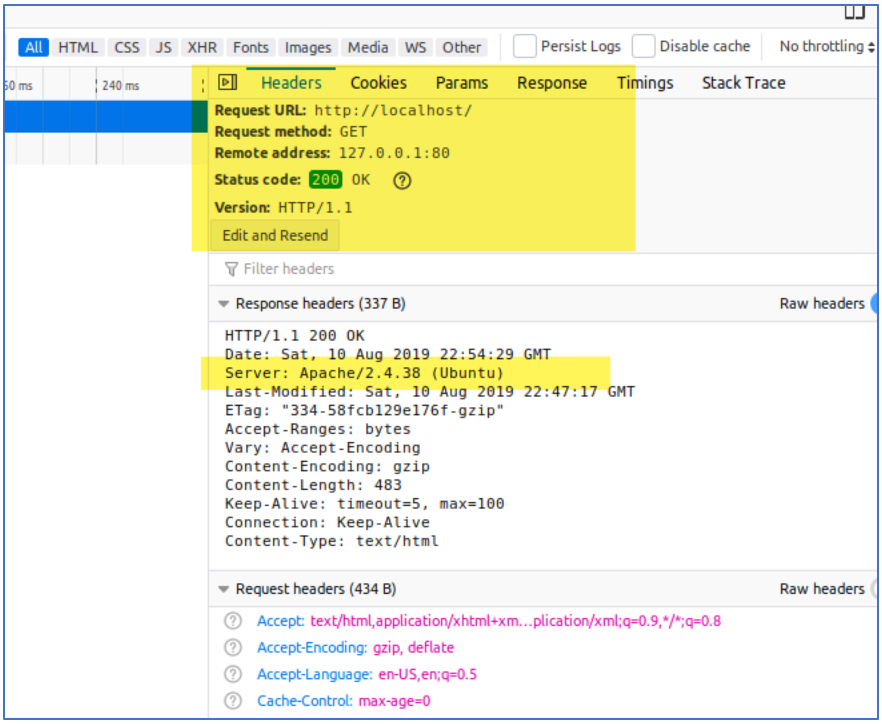


Backup Apache2.conf file before editing

1) `sudo cp apache2.conf apache2-backup.conf`

Removing HTTP Response Header

1) In FireFox web browser, in accessing the *Web Developer* tool (F12), then clicking on *Network* tab for the localhost index.html page, we can analyze the http headers for this page request. The raw headers can be displayed (see screenshot below).



Before enabled ServerTokens directive

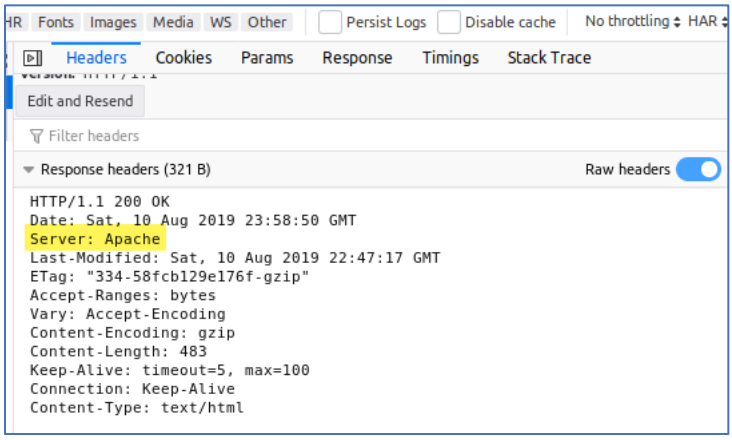
Removing the Apache 2.4 Response header and server signature to stop information leaking.

1) Edit apache 2 config file

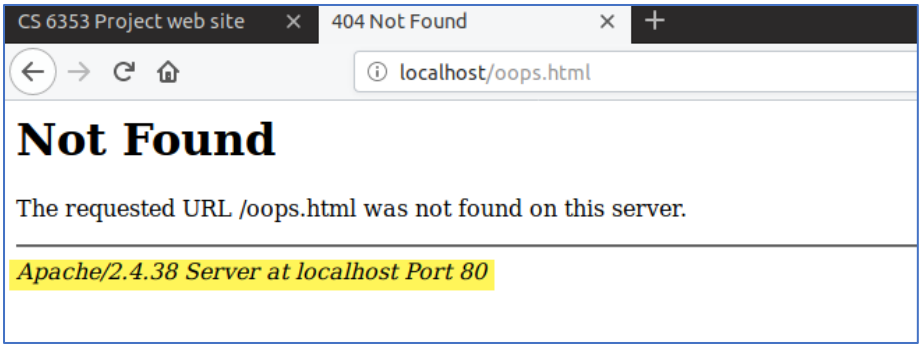
   `sudo /etc/apache2/apache2.conf`

   add these directives:

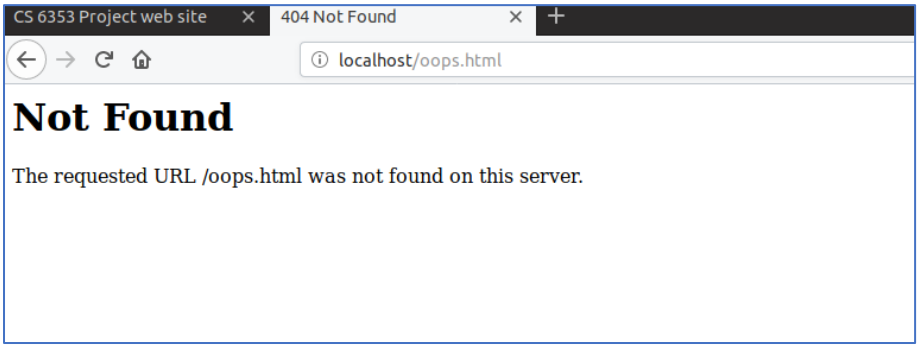
   a) `ServerTokens Min`
   b) `ServerSignature off`
   c) See http://httpd.apache.org/docs/2.4/mod/core.html#servertokens



Before enabled ServerTokens directive



Before enabling *ServerSignature* directive

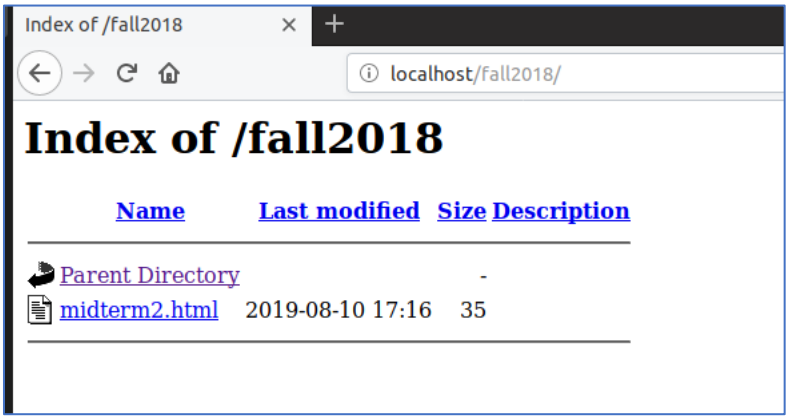


After enabling *ServerSignature* directive

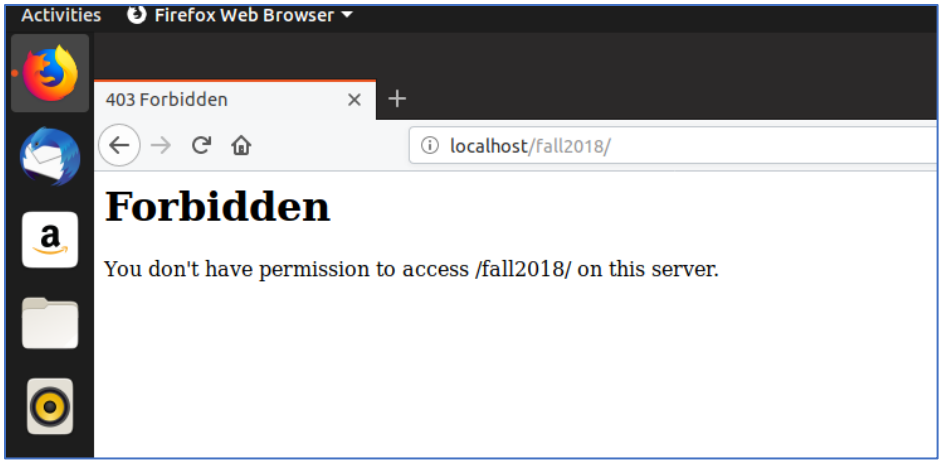
Disabling Directory Listing

By default, apache display the directory listing of a web site.

1) In apache2.conf, add directive:



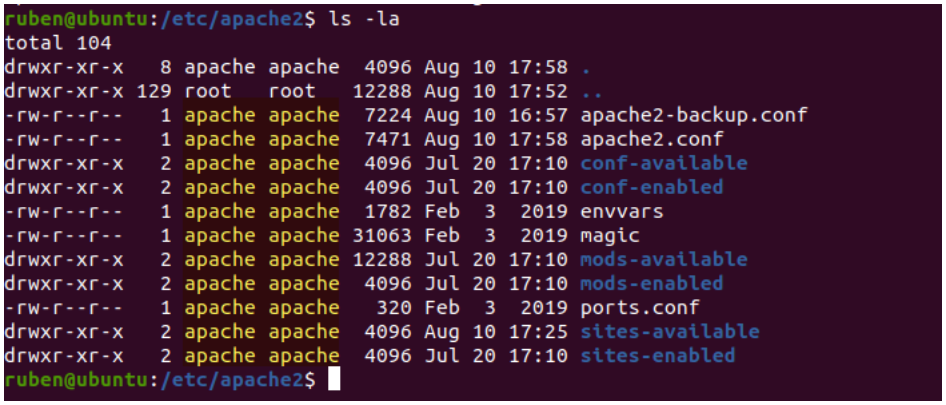
Before enabling *Options -Indexes* directive



After enabling *Options -Indexes* directive

Running Apache as a non-root different user

1) Add a group *apache2*
   `sudo groupadd apache2`
2) Add a user apache2
   `useradd -G apache2 apache2`
   `chown -R apache:apache /opt/apache`



```
ruben@ubuntu:/etc/apache2$ ls -la
total 104
drwxr-xr-x   8 apache apache  4096 Aug 10 17:58 .
drwxr-xr-x 129 root   root   12288 Aug 10 17:52 ..
-rw-r--r--   1 apache apache  7224 Aug 10 16:57 apache2-backup.conf
-rw-r--r--   1 apache apache  7471 Aug 10 17:58 apache2.conf
drwxr-xr-x   2 apache apache  4096 Jul 20 17:10 conf-available
drwxr-xr-x   2 apache apache  4096 Jul 20 17:10 conf-enabled
-rw-r--r--   1 apache apache  1782 Feb  3  2019 envvars
-rw-r--r--   1 apache apache 31063 Feb  3  2019 magic
drwxr-xr-x   2 apache apache 12288 Jul 20 17:10 mods-available
drwxr-xr-x   2 apache apache  4096 Jul 20 17:10 mods-enabled
-rw-r--r--   1 apache apache   320 Feb  3  2019 ports.conf
drwxr-xr-x   2 apache apache  4096 Aug 10 17:25 sites-available
drwxr-xr-x   2 apache apache  4096 Jul 20 17:10 sites-enabled
ruben@ubuntu:/etc/apache2$
```

After changing ownership to apache user

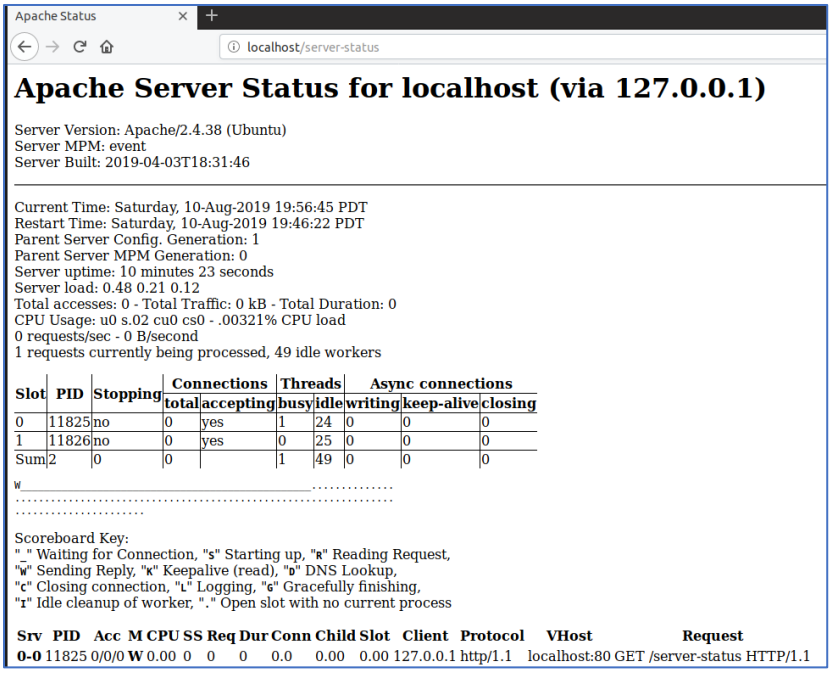
Disabling .htaccess

1)bDisabling .htaccess to allow vhosts to override the main apache2 configuration file

Inside <Directory /> directory directive tag, change `AllowOverride to None`. Note with Apache 2.4, this setting is already to None but default.

Disable unneeded modules

1) The status module display server status information. This page is display with the *status* module.



Apache Status

localhost/server-status

Apache Server Status for localhost (via 127.0.0.1)

Server Version: Apache/2.4.38 (Ubuntu)
Server MPM: event
Server Built: 2019-04-03T18:31:46

Current Time: Saturday, 10-Aug-2019 19:56:45 PDT
Restart Time: Saturday, 10-Aug-2019 19:46:22 PDT
Parent Server Config. Generation: 1
Parent Server MPM Generation: 0
Server uptime: 10 minutes 23 seconds
Server load: 0.48 0.21 0.12
Total accesses: 0 - Total Traffic: 0 kB - Total Duration: 0
CPU Usage: u0 s.02 cu0 cs0 - .00321% CPU load
0 requests/sec - 0 B/second
1 requests currently being processed, 49 idle workers

Slot	PID	Stopping	Connections		Threads		Async connections		
			total	accepting	busy	idle	writing	keep-alive	closing
0	11825	no	0	yes	1	24	0	0	0
1	11826	no	0	yes	0	25	0	0	0
Sum	2		0		1	49	0	0	0

W_____.....
.....
.....

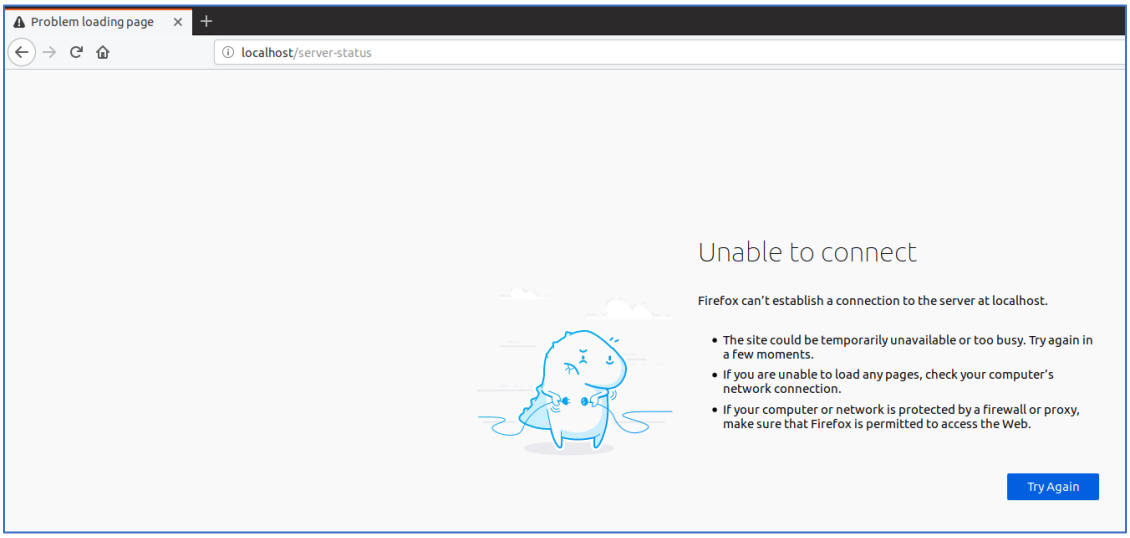
Scoreboard Key:
"_" Waiting for Connection, "s" Starting up, "R" Reading Request,
"W" Sending Reply, "K" Keepalive (read), "D" DNS Lookup,
"C" Closing connection, "L" Logging, "G" Gracefully finishing,
"I" Idle cleanup of worker, "." Open slot with no current process

Srv	PID	Acc	M	CPU	SS	Req	Dur	Conn	Child	Slot	Client	Protocol	VHost	Request
0-0	11825	0/0/0	W	0.00	0	0	0	0.0	0.00	0.00	127.0.0.1	http/1.1	localhost:80	GET /server-status HTTP/1.1

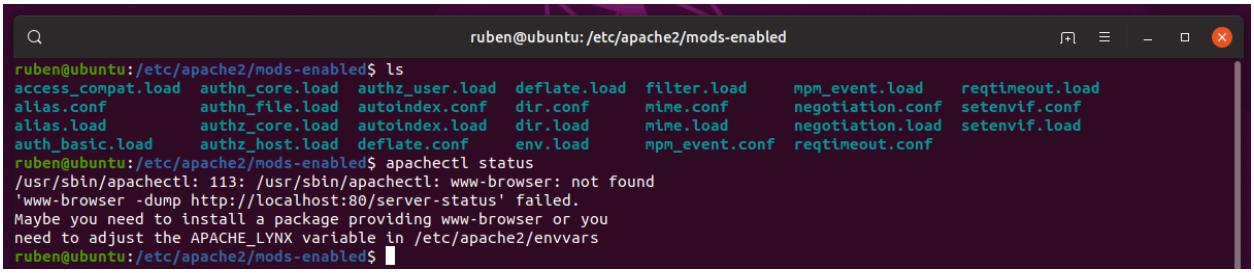
Before disabling *status* module

Disable the status (*mod_status.so*) module

```
sudo a2dismod status
```



After disabling *status* module

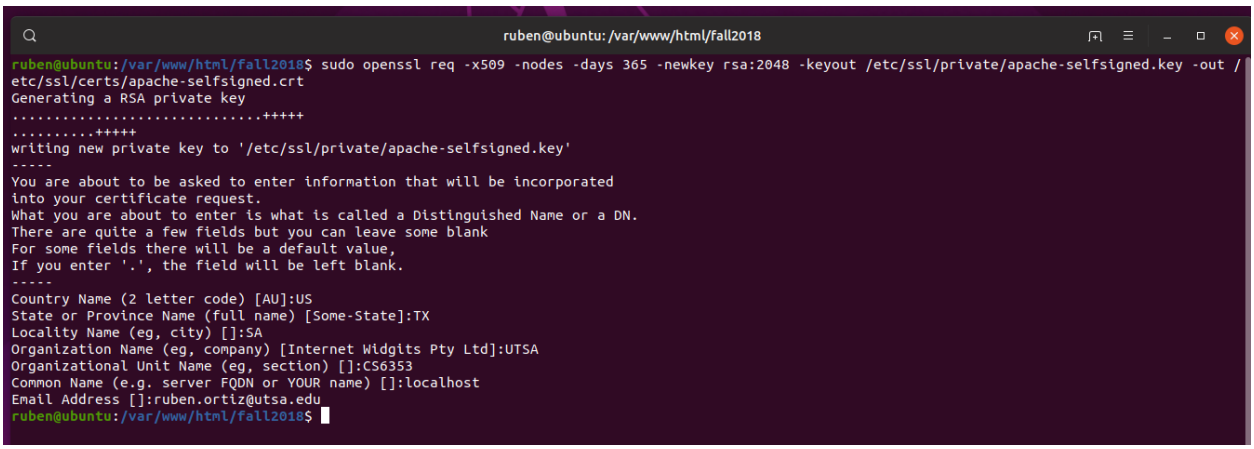


After disabling *status* module, part 2

Creating a Self-Signed Certificate

1) openssl:

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



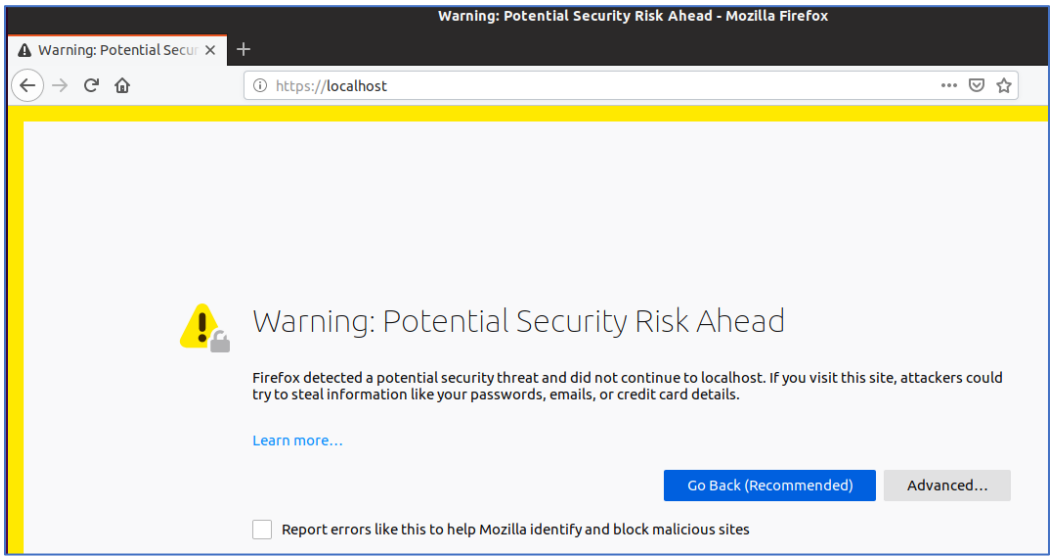
Enabling SSL on Apache

1) switch directory to sites-available directory
   `cd etc/apache2/sites-available`
2) modify the default ssl conf file
   `sudo vim default-ssl.conf`
3) Inside default-ssl.conf *comment out* these parameters:
   ```
   #SSLCertificateFile      /etc/ssl/certs/ssl-cert-snakeoil.pem
   #SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key
   ```
4) Inside default-ssl.conf *add* these parameters:
   ```
   ServerName localhost
   Redirect "/" "https:localhost"
   SSLCertificateFile  /etc/ssl/certs/apache-selfsigned.crt
   SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key
   ```
5) Enable SSL module
   `sudo a2ensite default-ssl`

Enabling Redirect module

1) Enable the redirect module with a2enmod command
   `sudo a2enmod rewrite`

2) Enable the default ssl web site:
```
sudo a2ensite default-ssl
```



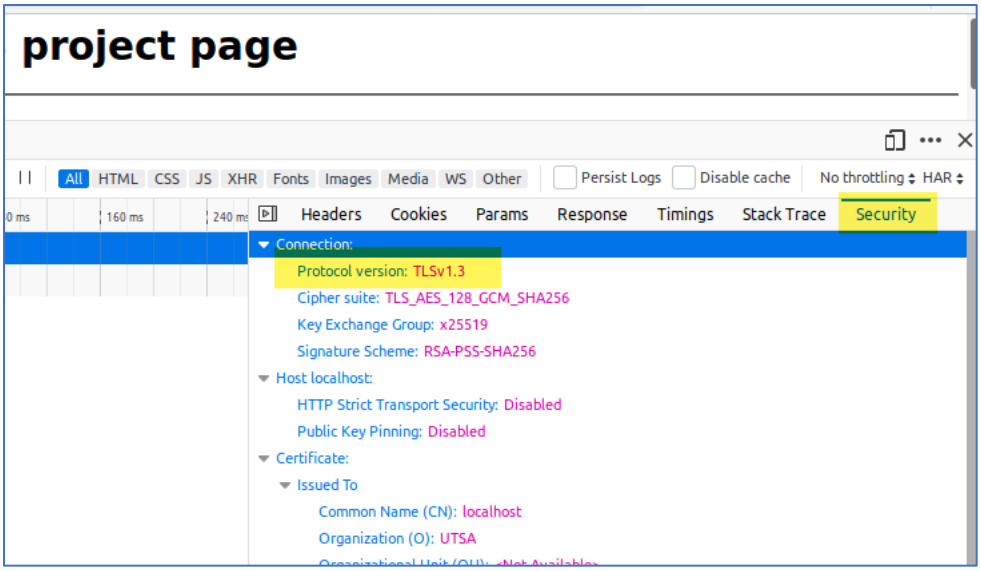
Firefox warning about self-signed certificate

3) In Firefox web browser, a security exception can be added to allow web browser to not display "Security Risk Ahead" message.

Site with SSL encryption and FireFox exception added.

Enabling TLS encryption and Disable older encryptions

1) In the ssl.conf module
```
sudo vim ssl.conf
```
2) Comment out this line (#):
```
SSLProtocol all -SSLv2
```
3) Add this line:
```
SSLProtocol -all +TLSv1.3
```



The web page connection using TLSv1.3 after adjusting the *ssl.conf* file.

Disabling weak ciphers

1) sudo vim ssl.conf
2) comment out this line:
   `SSLCipherSuite HIGH:!aNULL`
3) Add this line:
   `SSLCipherSuite HIGH:!MEDIUM:!aNULL:!MD5:!RC4`

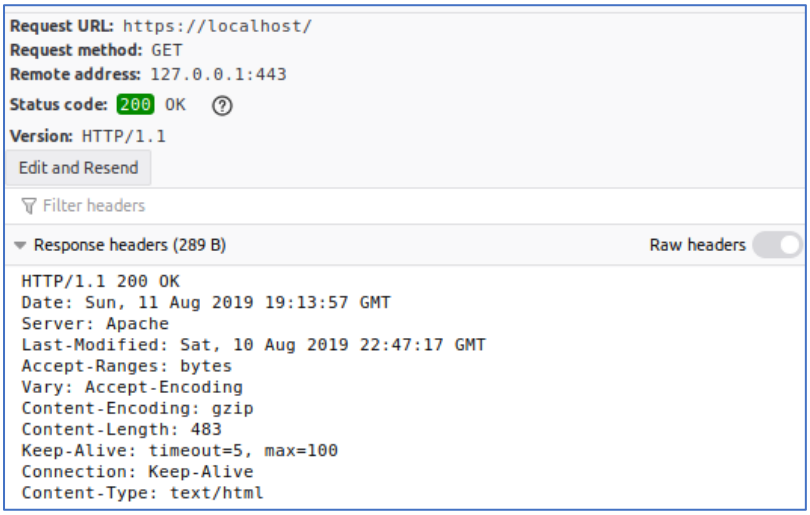
Enable mod_ALLOWMETHODS module

1) Run a2enmod to enable allow methods mod:
   `sudo a2enmod allowmethods`
2) Stop and Start apache2
   ```
   sudo apachectl -k stop
   sudo apachectl -k start
   ```
3) Edit apache2.conf
   Add the LimitExcept directive code block in yellow.
   ```
   <Directory /var/www/>
           #Options Indexes FollowSymLinks
           # CS6353 - Disable Directory listing
           Options -Indexes
           AllowOverride None
           Require all granted
           <LimitExcept GET POST>
              Deny from all
           </LimitExcept>
   </Directory>
   ```

Remove ETag header and set timeout setting

1) Sudo vim apache2.conf
2) Change timeout setting from 300 to 60:
   `Timeout 60`

3) Add Directive:
   Add directive: `FileETag None`



Raw headers without ETag response header.