**Apache with Mod security and SSL configured (RHEL 8)**

Pre requisites (old versions – Download latest Versions from below links)

* 1. apr-1.6.3.tar.gz
  2. apr-util-1.6.1.tar.gz
  3. pcre-8.42.tar.gz
  4. modsecurity-2.9.2.tar.gz
  5. httpd-2.4.33.tar.gz

<http://apache.mirrors.ionfish.org//apr/apr-1.7.0.tar.gz>

<http://apache.osuosl.org//apr/apr-util-1.6.1.tar.gz>

<https://ftp.pcre.org/pub/pcre/pcre-8.43.tar.gz>

<https://github.com/SpiderLabs/ModSecurity/releases/download/v2.9.3/modsecurity-2.9.3.tar.gz>

<http://mirror.cogentco.com/pub/apache//httpd/httpd-2.4.39.tar.gz>

Use wget with urls to download the packages, above links also is source of latest package , download latest package as and when necessary

Download “owasp-modsecurity-crs” from <https://github.com/SpiderLabs/owasp-modsecurity-crs.git>

Unzip all the downloaded packages to any path you want (/mnt/data/downloads)

Install below RHEL packages and libraries

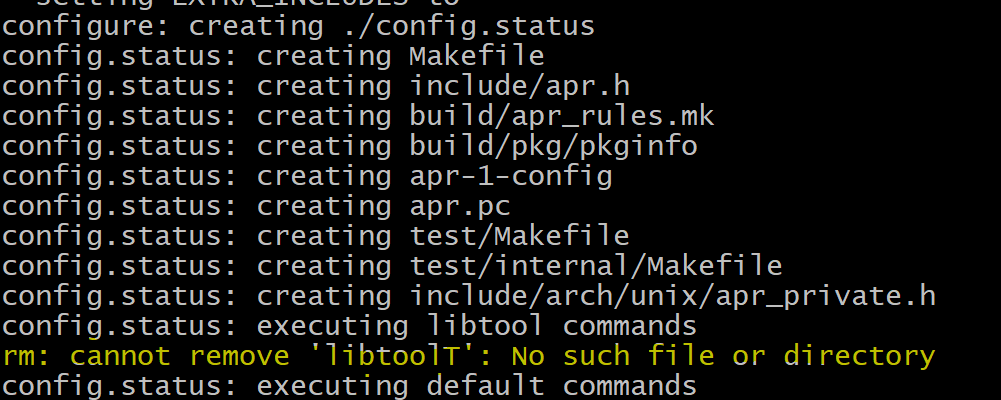
yum install openssl-devel automake libtool libxml2-devel libxml2 libxml2-devel httpd-devel pcre-devel curl-devel gcc make gcc-c++ autoconf automake expat-de\* expat-devel

yum install lua\*

## Installation of apr-1.6.3 (whichever is latest)

* cd apr-1.6.3
* ./configure
* make
* make install

While executing ./configure we get below error :



We need to refer the link <https://stackoverflow.com/questions/18091991/error-while-compiling-apache-apr-make-file-not-found> . And make changes in .configure file

Change the line

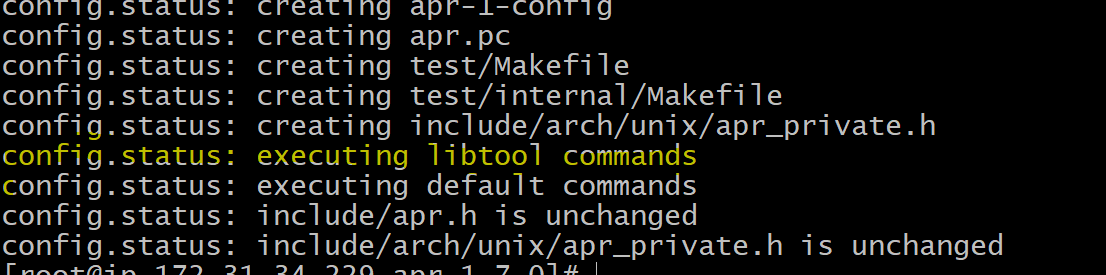
$RM "$cfgfile"

to

$RM -f "$cfgfile"

Then run  **./configure** again and see if LibTool error is eliminated

Below you can see ./configure ran successfully and no LIBTOOL error as before:



After executing the above steps, verify this path exist ‘/usr/local/apr/lib’ to confirm the installation is successful.

## Installation of apr-util-1.6.1 (whichever is latest)

Execute the following steps one by one.

* cd apr-util-1.6.1
* ./configure --with-apr=/usr/local/apr
* make
* make install

After executing the above steps, verify libaprutil-\*.so file(s) exist ‘/usr/local/apr/lib’ to confirm the installation is successful

## Installation of pcre-8.42 (whichever is latest)

Execute the following steps one by one.

* cd pcre-8.42
* ./configure
* make
* make install

After executing the above steps, verify libpcre\*.so file(s) exist ‘/usr/local/lib’ to confirm the installation is successful.

## Installation of httpd-2.4.33 (whichever is latest)

Execute the following steps one by one.

* cd httpd-2.4.33
* ./configure --prefix=/data/opt/apache-2.4.33 --enable-rewrite=shared --with-mpm=prefork --enable-proxy=shared --enable-proxy\_balancer=shared --enable-proxy\_http=shared --enable-mods-shared=all --enable-ssl --enable-so --enable-ssl --with-ssl=/bin/openssl --enable-cgid --enable-cgi --enable-modules=all --with-apr=/usr/local/apr/ --with-apr-util=/usr/local/apr/
* make
* make install

After executing the above steps, verify this path exists ‘/data/opt/apache-2.4.33’ to confirm the installation is successful.

Note Modify the apache version according to the version you downloaded.

## Installation of modsecurity-2.9.2 (whichever is latest)

Execute the following steps one by one.

* cd modsecurity-2.9.2
* ./configure --with-apxs=/data/opt/apache-2.4.33/bin/apxs
* make
* make install

After executing the above steps, verify this path exists ‘/usr/local/modsecurity/lib’ to confirm the installation is successful.

* Check whether “mod\_security2.so” is available under “/data/opt/apache-2.4.33/modules/mod\_security2.so”.
* Copy “/modsecurity-2.9.2/unicode.mapping” to “/data/opt/apache-2.4.33/conf/”.
* Rename “owasp-modsecurity-crs” folder as “crs” and move it to /data/opt/ apache-2.4.33/conf.
* Append the below lines in the existing ‘httpd.conf’ which is available under /data/opt/apache-2.4.33/conf/.

Go to the apache folder ->conf -> make a backup of httpd.conf, then open httpd.conf and **append the below lines specific to AFLS In the httpd.conf**



* Create a new file ‘modsecurity.conf’ under /data/opt/apache-2.4.33/conf/ and add lines from modsecurity.txt . After that open modsecurity.conf and rectify the apache path in **SecAuditLog** to the one you have configured

**SecAuditLog /data1/opt/apache-2.4.39/logs/modsec\_audit.log**





* Unzip the attached ‘activated\_rules.zip’ and ‘optional\_rules.zip’ files under “/data/opt/apache-2.4.33/conf/crs”.
* Start the apache and check the logs to ensure apache is started with mod security configuration. Now our httpd.conf includes https configuration as well so we have to generate self signed certificates and place in apache conf folder.

NOTE- In latest Apache httpd-ssl.conf will be present in /data/opt/apache-2.4.39/conf/extra folder

Copy the httpd-ssl.conf from extra folder and put in conf folder. Also we need to make changes in httpd.conf AFLS part (MOD Security configuration area) , replace the line LoadFile /usr/lib64/liblua-5.1.so with LoadFile /usr/lib64/liblua-5.3.so as in new version /usr/lib64 has 5.3 version and not 5.1.



**Steps for generating self signed certificates**

**Refer the link** <https://www.rosehosting.com/blog/how-to-generate-a-self-signed-ssl-certificate-on-linux/>

Sometimes while running the first command i.e. openssl genrsa -des3 -passout pass:x -out server.pass.key 2048 we will get below error

Generating RSA private key, 2048 bit long modulus

...............................................................+++

...................................................................+++

e is 65537 (0x010001)

140107418351360:error:28069065:UI routines:UI\_set\_result:result too small:crypto/ui/ui\_lib.c:778:You must type in 4 to 1023 characters

140107418351360:error:28069065:UI routines:UI\_set\_result:result too small:crypto/ui/ui\_lib.c:778:You must type in 4 to 1023 characters

140107418351360:error:0906906F:PEM routines:PEM\_ASN1\_write\_bio:read key:crypto/pem/pem\_lib.c:336:

To resolve this use below command use below command instead , more info in this link <https://bugzilla.redhat.com/show_bug.cgi?id=1467669>

# **openssl genpkey -algorithm RSA -pkeyopt rsa\_keygen\_bits:2048 -out server.key**

**Example Steps:**

openssl genpkey -algorithm RSA -pkeyopt rsa\_keygen\_bits:2048 -out server.key

openssl rsa -passin pass:x -in server.key -out 172.31.34.229.key

rm server.key

openssl req -new -key 172.31.34.229.key -out 172.31.34.229.csr

openssl x509 -req -days 365 -in 172.31.34.229.csr -signkey 172.31.34.229.key -out 172.31.34.229.crt

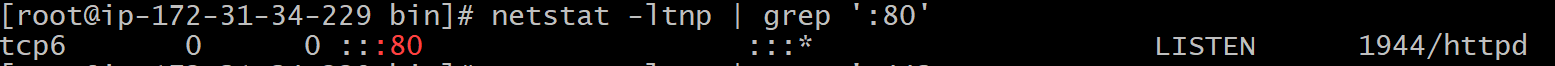
Now go to apache conf folder and modify the path for certificates (Its preferable to keep all certs in apache conf folder) and comment the line that starts with **SSLSessionCache**

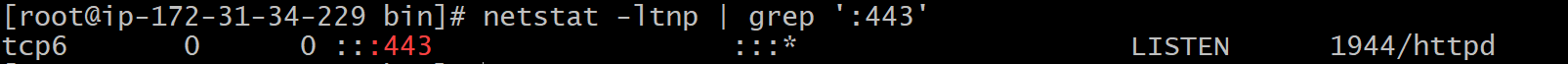
Now go to apache bin folder and start apache **./apachectl start**

**Verification steps**:

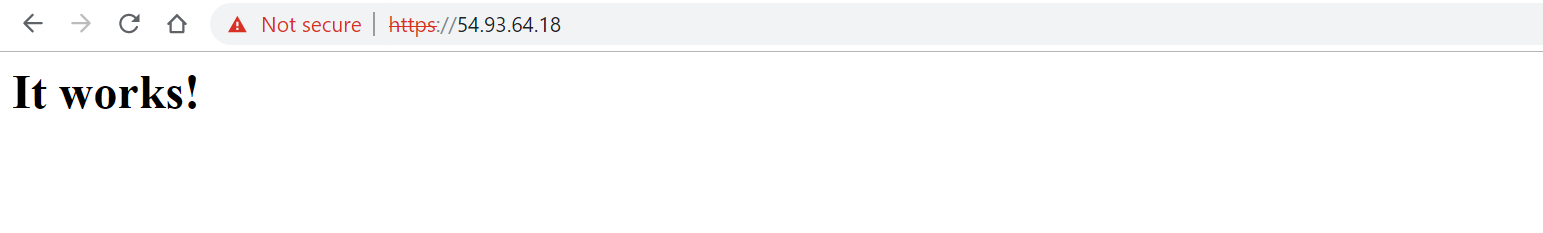
netstat -ltnp | grep ':80'

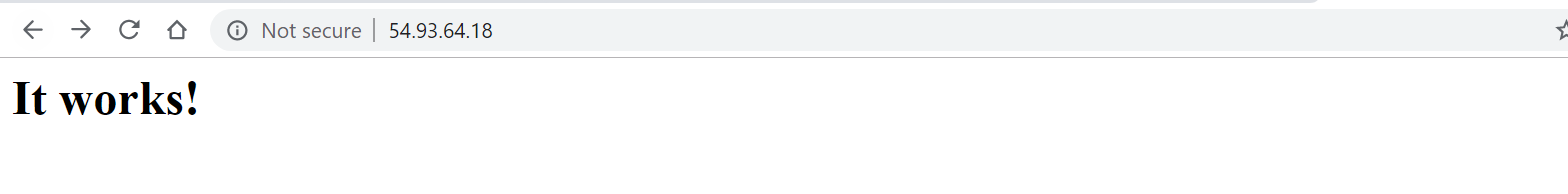
netstat -ltnp | grep ':443'





**Now please use the browser to check the http and https connection**





Further information can be retrieved by going to apache logs folder and taking a look at all log files