# Apache Debian HTTPS Setup

The Debian approach uses Apache console command line utilities that controls which modules to load and selecting site configurations. An overview for the Debian Appache 2 directory structure is described in “/etc/apache2/apache2.conf” file. With their approach to enable modules the a2enmod and a2dismod command line utilities are available . For the web sites use a2ensite and a2dissite utilities. A list of the available module files are located in “/etc/apache2/mods-available” directory.

```
/etc/apache2/
    |-- apache2.conf
    |       `--  ports.conf
    |-- mods-enabled
    |       |-- *.load
    |       `-- *.conf
    |-- conf.d
    |       `-- *
    `-- sites-enabled
            `-- *
```

To determine if Open SSL is installed, enter the following command:

```
openssl version
OpenSSL 1.0.1e 11 Feb 2013
```

If there's a need to create a certificate then execute the following:

```
openssl req -new -x509 -days 365 -nodes -out /etc/ssl/localcerts/apache.pem -keyout /etc/ssl/localcerts/apache.key
```

Use the key to create a certificate.

```
openssl req -x509 -new -set_serial 1 -key apache.key -out apache.crt
```

Change the certificate files access permissions with the following command:

```
chmod 600 /etc/ssl/localcerts/apache*
```

Install the module is requird by using package installer. To view current modules installed enter: apachectl -M

```
rewrite_module (shared)
ssl_module (shared)
```

To add a module enter the following command: *a2enmod ssl*

As a note in sites-enabled folder you will find links to the current configuration file in use:

```
lrwxrwxrwx 1 root root 26 Oct 15  2013 000-default -> ../sites-available/default
```

The current LSS Database virtual machine has default with the following virtual host definitions:

```
NameVirtualHost *:80
NameVirtualHost *:443

<VirtualHost *:80>
        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/
</VirtualHost>

<VirtualHost *:443>
        SSLEngine on
        SSLCertificateFile /etc/ssl/localcerts/apache.crt
        ssLCertificateKeyFile /etc/ssl/localcerts/apache.key
        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/
</VirtualHost>

<Directory /var/www/>
        Options Indexes FollowSymLinks MultiViews
        Order allow,deny
        allow from all
</Directory>
```

To setup the site name use the following command with the desired name:

```
a2ensite sitename
```

In the files “index.php” uncomment the following to use only https:

```
// Redirect to https
if(window.location.protocol !== 'https:') {
            location.href = location.href.replace("http://", "https://");
}
```

When apache has been configured enter the following command:

```
service apache2 restart
```

In the following are some helpful debugging tips for getting the web site up and running.

To ensure that the modules and configuration files are loaded execute:

```
/etc/init.d/apache2 reload
```

To verify the configuration enter: *apachectl configtest*
The command utility is in the /usr/sbin directory and require root user (su) to access it.

For error check review the log at:

```
tail -f /var/log/apache2/error.log
```

To expand the amount of data being logged enable debug mode by editing the /etc/apache2/apache2.conf file as root user rights (su command recommended). Within the file look for the following section and edit as shown.

```
# LogLevel: Control the number of messages logged to the error_log.
# Possible values include: debug, info, notice, warn, error, crit,
# alert, emerg.
#
LogLevel debug
```

To verify the Virtual Hosts enter the following command:

```
apache2ctl -S
```

- Remember that *included files* are read at the point of their inclusion, before the rest of the original file is read.
- `<Directory>` settings are read whenever the server starts or is reloaded. `.htaccess` files, on the other hand, are read before resources are served. As a result, `.htaccess` files can override directory configurations. To test whether this is occurring, temporarily disable `.htaccess` files.
- `<Location>` directives are read after `<Directory>` and `<Files>` sections, so settings here might override other earlier settings.
- Configuration files are read serially. For example, an option set in the beginning of the `apache2.conf` or `httpd.conf` file could be overridden by a setting in the `conf.d/` file or a virtual host file.
- When an entire directory is included, the files from that directory are included sequentially (alphabetically) based on name.
- Debian and Ubuntu systems have a file called `/etc/apache2/ports.conf`, where the `NameVirtualHost` and `Listen` directives are set. These values determine the IP address or addresses to which Apache binds, and on which port(s) the web server listens for HTTP requests. This can sometimes conflict with settings in other files.

When SE Linux is installed and running there can be a case that’s policies are blocking the web servers operation by checking is status as root and using the following commands:

```
selinux or check-selinux-installation
```

default Apache webserver listen on port 80 (http) and port 443 (https i.e. secure http). Apache webserver uses the TCP protocol to transfer information/data between server and browser. The default Iptables configuration does not allow inbound access to the HTTP (80) and HTTPS (443) ports used by the web server.

Another item that could block communications is the iptables configuration and iproute2. These utilities can block inbound and outbound communications. Use the ifconfig command to review interface status and then review the iptables and iprout2 configurations. The configuration files for iproute can be found in the /etc/iproute2 directory. The current startup routes are located in /etc/iptables.up.rules file.

The use of the iptables can effectively serve as a method to firewall an interface.

```
root@lamp /etc# cat iptables.up.rules
*nat
:PREROUTING ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
COMMIT
*mangle
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
COMMIT
*filter
:FORWARD ACCEPT [0:0]
:INPUT DROP [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -i lo -j ACCEPT
-A INPUT -p icmp -m icmp --icmp-type echo-request -j ACCEPT
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 443 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 12320 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 12321 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 12322 -j ACCEPT
COMMIT
root@lamp /etc#
```